SIP (2021), vol. 10, e3, page 1 of 12
doi:10.1017/ATSIP.2020.28

ORIGINAL PAPER

Toward community answer selection by jointly static and dynamic user expertise modeling

YUCHAO LIU,[1] MENG LIU[2] AND JIANHUA YIN[1]

Answer selection, ranking high-quality answers first, is a significant problem for the community question answering sites. Existing approaches usually consider it as a text matching task, and then calculate the quality of answers via their semantic relevance to the given question. However, they thoroughly ignore the influence of other multiple factors in the community, such as the user expertise. In this paper, we propose an answer selection model based on the user expertise modeling, which simultaneously considers the social influence and the personal interest that affect the user expertise from different views. Specifically, we propose an inductive strategy to aggregate the social influence of neighbors. Besides, we introduce the explicit topic interest of users and capture the context-based personal interest by weighing the activation of each topic. Moreover, we construct two real-world datasets containing rich user information. Extensive experiments on two datasets demonstrate that our model outperforms several state-of-the-art models.

Keywords: User expertise, community question answering, answer selection, graph neural network

Received 25 September 2020; Revised 25 December 2020

I. INTRODUCTION

Community-based question answering (cQA) sites such as Zhihu, Quora, and Stack Overflow are forums for information exchanging and knowledge sharing which have become more and more popular. These sites enable users to find suitable answers by posting questions. It provides users a simple and effective way of solving personal questions. The benefits of cQA sites which are proven in [1] are well-recognized. Nowadays, with a large number of users joining cQA sites, there are many answers for each question including some unreliable answers.

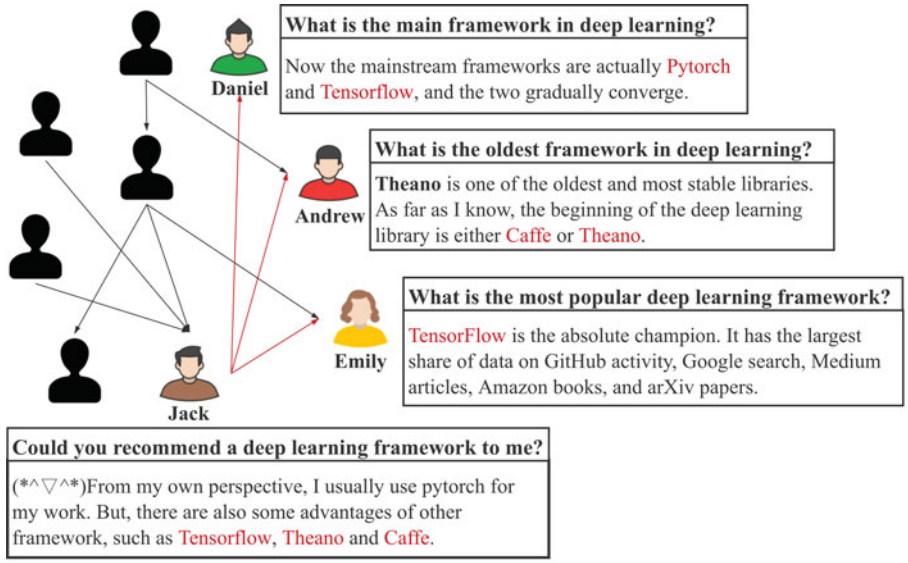
Answer selection aims at selecting a high-quality answer that is relevant to the given question from a list of candidate answers. It is a significant task in question answering. Answer selection in cQA sites has drawn much attention, which can save time for filtering out unreliable answers and give users a more satisfying experience.

Currently, deep learning models are widely used in natural language processing. Most existing approaches in answer selection also leverage deep learning architecture, e.g. convolutional neural network [2] and recurrent neural network [3]. These approaches regarded answer selection as a text matching task and designed deep matching neural networks to learn the semantic relevance score of questions and answers. The representation-based methods encode questions and answers from a high-dimensional representation to a low-dimensional distributed representation separately based on their context content, and then a prediction layer is utilized to calculate the final matching score [2, 4, 5]. The interaction-based methods exploit attention mechanism to learn the word-level interaction of question–answer pairs to get more semantic information [6–8]. Although these traditional approaches show promising performance in modeling the semantic similarity of questions and answers, they mainly focus on text content while ignoring the influence of other factors existing in the community such as user expertise.

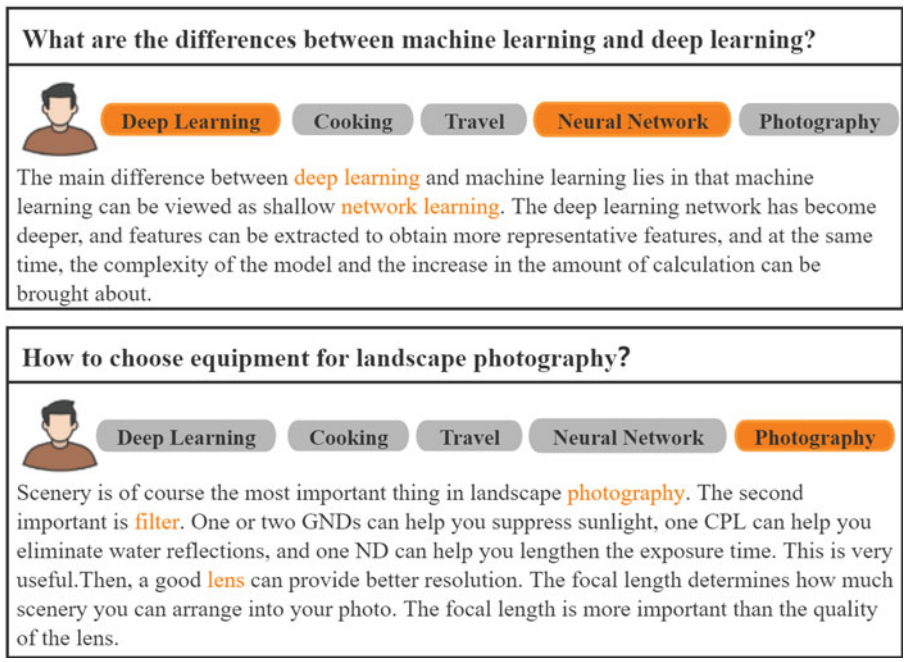
Actually, user expertise plays an important role in evaluating the answer quality, in that users are more likely to provide high-quality answers to questions in the field they are expert in. Recently, many researchers have focused on user authority modeling. The authors utilized user-generated answers and the given question to model the expertise of users in [9]. And an adversarial training module is applied to handle the noise issue caused by introducing the user's historical answers in [10]. Moreover, latent factors are introduced to capture the implicit interested topics of users in [11]. In addition, Fang *et al.* [12], Hu *et al.* [13], and Zhao *et al.* [14] proposed heterogeneous social network learning architecture to model questions, answers, and users jointly. They used the random walk-based method [12] or the meta-path-based method [13] to gain additional social context information from the heterogeneous network. Most existing user modeling approaches only focus on a single factor

[1]Shandong University, Jinan, China
[2]Shandong Jianzhu University, Jinan, China

Corresponding author:
J. Yin
Email: jhyin@sdu.edu.cn



(a)



(b)

Fig. 1. Illustration of (a) static social influence and (b) dynamic personal interest.

and model user expertise as a static representation. In cQA sites, user expertise is influenced by various factors and it can be obtained from multiple aspects.

To model user expertise from different views and integrate it into the answer selection model is non-trivial, nevertheless, due to users in the cQA sites confronted with multi-aspect influences: (1) static social influence. In cQA sites, users can follow other users with similar interest, and their opinions are apt to be affected by them. As shown in Fig. 1(a), Jack follows Emily, Andrew, and Daniel. Therefore, he frequently browses the answers posted by them, leading to his point is easily influenced by these users. In light of this, the social network causes an intrinsic influence on modeling the user expertise. And (2) dynamic personal interest. Users follow some topics they are interested in or related to their fields, which reflect the personal interest of users. However, user interest varies dynamically for specific contexts. In other words, the interest of users can cover many aspects at the same time, while only part of the interest information can be activated given the specific context. As Fig. 1(b) shows, the user is interested in deep learning, photography, and so on. When faced with a question relevant to deep learning, his interest in "deep learning" is activated. And his interest in "photography" is activated by the other question. Activating irrelevant interest information may lead to negative user modeling and further influences

https://doi.org/10.1017/ATSIP.2020.28

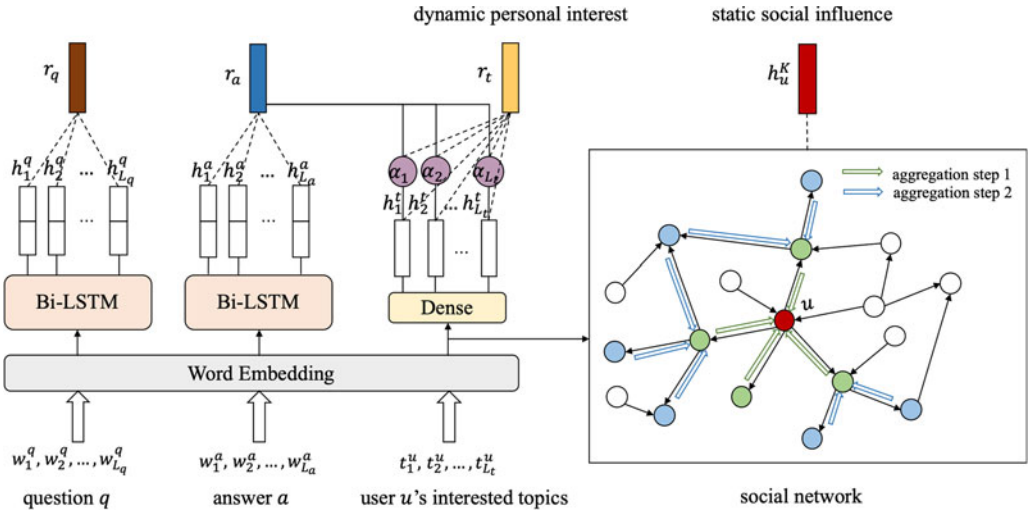


Fig. 2. Overview of our proposed SCAD model. It encodes the user expertise by jointly considering the dynamic personal interest and the static social influence. The aggregation process of the social influence takes $K = 2$ as an example.

the answer selection. Thus, it is expected precise modeling of the user interest.

As both of the above aspects would influence the user expertise modeling, we should jointly consider these factors to model user expertise and then integrate it into the answer selection model. Inspired by this, in this paper, we propose a model toward community answer selection by jointly StatiC And Dynamic user expertise modeling, dubbed as SCAD. As illustrated in Fig. 2, it is a novel architecture converging the user information obtained from different perspectives. In other words, we model users from both long-term inherent social influence and dynamic personal interest. Specifically, on the one hand, to capture the social influence from adjacent users, we introduce an inductive framework to aggregate the neighbors' interest. On the other hand, instead of learning users' interest implicitly, we extract the personal interest of each user from the explicit interested topics of users. The activation of each topic is weighed dynamically to capture the precise interest description for the specific context.

In summary, the main contributions of this paper are as follows:

- We jointly consider two factors, social influence, and personal interest, that impact the user expertise statically and dynamically to improve the capacity of our model.
- We are the first to introduce the explicit user interested topics and obtain users' dynamic personal preference by weighing the activation of each topic. Moreover, we extend the graph neural network (GNNs) to model the social influence existing in the community more convincingly.
- We construct real-world datasets and conduct extensive experiments on them. The results demonstrate that our model outperforms several state-of-the-art approaches. We also conduct ablation experiments to verify the effect of each module. As a byproduct, we have released the data, codes, and involved parameter settings to facilitate other researchers.[1]

The rest of this paper is organized as follows. We present previous related work in Section II. We elaborate the detail of our proposed model in Section III. Experiment results and analysis are shown in Section IV. Finally, we conclude this paper in Section V.

II. RELATED WORK

A) Community-based answer selection

Deep learning models have shown state-of-the-art performance in many natural language processing tasks, including answer selection. The neural network-based methods applied to answer selection can be divided into two main categories: representation-based methods and interaction-based methods. The representation-based methods generate a low-dimensional representation for questions and answers independently, and then calculate the matching score based on their vector distance in the same feature space which can reflect their semantic similarity. Severyn and Moschitti [2] employed convolutional neural networks to mapping questions and answers to their distributional vectors and used k-max pooling to aggregate the information. Qiu and Huang [4] proposed a convolutional neural tensor network to obtain rich representation information for question answer matching. As an alternative to the tensor layer, Tay *et al.* [5] adopted holographic composition with associative memory and circular correlation to model the relationship between question and answer embeddings. The counterpart sentence isn't taken into account until the final matching layer that results in limited performance. Instead of encoding questions and answers separately, interaction-based methods

[1] https://lycresearch.wixsite.com/scad

https://doi.org/10.1017/ATSIP.2020.28

are proposed to capture more fine-grained word-level or sentence-level interaction of questions and answers. Zhang *et al.* [8] proposed a novel tensor interactive attention mechanism architecture to depict interactions. Khanh Tran and Niedereée [7] located answer segments that are relevant to questions via multiple attention steps. Wen *et al.* [9] proposed a hybrid attention mechanism considering the local and mutual importance of the words in QA pairs. And an adversarial training module is applied in [10] to handle the noise issue caused by introducing the user's historical answers. Moreover, latent user vectors are introduced in [11] to capture the implicit topic interests of users.

Although the above approaches have gained great performance in cQA answer selection, they consider answer selection as a text matching task and only exploit text information of questions and answers. That may be sufficient for factoid question answering such as WikiQA, but for cQA, the judgment of answer quality is based on many other factors such as the authority of users. There are some works exploiting users' authority.

B) Graph neural networks

Recently, GNNs proposed in [15] have been employed for generating representation for graph-structured data due to its convincing performance and high interpretability. In [16], they proposed convolutional GCN for semi-supervised graph classification which extends convolutional operation from the regular Euclidean domains to non-Euclidean graph domains. In [17] the authors proposed a general inductive framework to generate node embedding for previously unseen data.

GNNs have been widely used in recommendation or natural language processing. In [18], they exploited GCN to simulate how users were influenced by the recursive social diffusion process of social recommendation. In [19], a graph-attention neural network is proposed to model the context-dependent social influence of users. A single text graph is built for a corpus based on word co-occurrence and document word relations [20]. And the researchers in [21] proposed the graph pooling layer and the hybrid convolutional layer for text modeling. To the best of our knowledge, there hasn't been anyone applied GNN to model the social influence of user expertise in cQA.

III. OUR PROPOSED MODEL

A) Problem formulation

We formulate answer selection as a ranking task. Its inputs include a given question q, a list of answer candidates represented by $\mathcal{A}_q$, and users who provided these candidates, i.e. $\mathcal{U}_q$. In addition, a list of the user's interested topics is also inputted to the model, such as "deep learning," and "photography." In this study, the quality of answer is based on the interaction modeling among the question, the answer, and the answerer. Specifically, a score function $f(q, a, u)$ is required to calculate the matching degree of each tuple (q, a, u).

B) Question and answer representations

For a question q with word sequence $(w_1^q, w_2^q, ..., w_{L_q}^q)$, we first represent each word by the pre-trained word embedding [22]. Then question q can be represented as $(\boldsymbol{x}_1^q, \boldsymbol{x}_2^q, ..., \boldsymbol{x}_{L_q}^q)$, where $\boldsymbol{x}_i^q$ is the word embedding for the i-th word in question q. As the long short-term memory (LSTM) [23] is widely used to learn long-term dependencies across the sentence, we hence use it to obtain context representations of questions and answers. Taking the sequence of word embedding $(\boldsymbol{x}_1^q, \boldsymbol{x}_2^q, ..., \boldsymbol{x}_{L_q}^q)$ as the input of the LSTM, the hidden state h_t at each time step t is calculated by the following equations:

$$\begin{aligned}
\boldsymbol{i}_t &= \sigma_g(\boldsymbol{W}_i \boldsymbol{x}_t + \boldsymbol{U}_i \boldsymbol{h}_{t-1} + \boldsymbol{b}_i) \\
\boldsymbol{f}_t &= \sigma_g(\boldsymbol{W}_f \boldsymbol{x}_t + \boldsymbol{U}_f \boldsymbol{h}_{t-1} + \boldsymbol{b}_f) \\
\boldsymbol{o}_t &= \sigma_g(\boldsymbol{W}_o \boldsymbol{x}_t + \boldsymbol{U}_o \boldsymbol{h}_{t-1} + \boldsymbol{b}_o) \\
\boldsymbol{c}_t &= \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tanh(\boldsymbol{W}_c \boldsymbol{x}_t + \boldsymbol{U}_c \boldsymbol{h}_{t-1} + \boldsymbol{b}_c) \\
\boldsymbol{h}_t &= \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t)
\end{aligned} \tag{1}$$

where $\boldsymbol{W}_i$, $\boldsymbol{W}_f$, and $\boldsymbol{W}_o$ are parameters, $\boldsymbol{i}_t$, $\boldsymbol{f}_t$ and $\boldsymbol{o}_t$ represent the input gate, the forget gate and the output gate, respectively. $\boldsymbol{c}_t$ denotes the cell state, σ_g is the sigmoid function, and $\odot$ is an element-wise multiplication.

To obtain the context information provided by future words, we employ the bi-directional LSTM (Bi-LSTM). The hidden state at time step k is the concatenation of the forward direction and the backward direction, i.e. $\boldsymbol{h}_t = [\overrightarrow{\boldsymbol{h}_t}, \overleftarrow{\boldsymbol{h}_t}]$. Afterward, we obtain the hidden state sequence $\boldsymbol{H}_q = (\boldsymbol{h}_1^q, \boldsymbol{h}_2^q, ..., \boldsymbol{h}_{L_q}^q)$ for question q. Besides, we take the mean pooling approach over $\boldsymbol{H}_q$ to obtain the final question representation $\boldsymbol{r}_q$. The final representation $\boldsymbol{r}_a$ for answer a is obtained in a similar way.

C) Social influence modeling

We capture the social influence from the following relation in the community. As illustrated in [14], the following relation in cQA sites is asymmetric, where the intrinsic social influence only comes from the neighbor users that the user is following. Hence, we formulate the following relation in cQA sites as a directed graph $G = (V, E)$, where V is the set of users and E is the set of edges. There is a directed edge (w, v) when user w follows user v.

1) Embedding layer

The initial embedding of each user u is the concatenation of the user's interest feature $\boldsymbol{x}_u$ and the latent vector $\boldsymbol{p}_u$. The former is obtained by applying the mean-pooling method over the word embedding of each explicit user interested topic. And the latter is random initialized to capture the implicit influence. The final user embedding is formulated

https://doi.org/10.1017/ATSIP.2020.28

as:

$$\boldsymbol{h}_u = \tanh(\boldsymbol{W}_p[\boldsymbol{x}_u, \boldsymbol{p}_u] + \boldsymbol{b}_p) \tag{2}$$

where $\boldsymbol{W}_p$, $\boldsymbol{b}_p$ are the trainable transformation parameters, and $[\boldsymbol{x}_u, \boldsymbol{p}_u]$ refers to the concatenation of the two vectors.

2) Aggregation layer

We extend the GNN to model the social influence aggregation, where each user node embedding with social influence is obtained with a hierarchical multi-step structure. For user u, let $\boldsymbol{h}_u^{k-1}$ denote the user's representation at $k-1$-th step. $\boldsymbol{h}_u^{k-1}$ contains social influence to user u from his/her $k-1$-hop neighbors. At the next k-th step, each user node aggregates the influence from his/her immediate neighbors. The propagation of social influence is calculated as:

$$\begin{aligned} \boldsymbol{h}_{\mathcal{N}(u)}^k &= Aggregate_k(\{\boldsymbol{h}_v^{k-1}\}, v \in \mathcal{N}(u)) \\ \boldsymbol{h}_u^k &= relu(\boldsymbol{W}_k[\boldsymbol{h}_u^{k-1}, \boldsymbol{h}_{\mathcal{N}(u)}^k]) \end{aligned} \tag{3}$$

where $\mathcal{N}(u)$ is the immediate neighbors of user u, $\boldsymbol{W}_k$ is the projection matrix and $Aggregate_k$ is the aggregate architecture at step k. In this study, we adopt the mean aggregator at each step. As the right part in Fig. 2 shows, the social influence is propagated to users through the following relation step by step. The user representation $\boldsymbol{h}_u^k$ got at k-th step contains the social influence from his/her neighbors within k distance. In addition, the user representation at step $k = 0$ is initialized with the user embedding obtained from the embedding layer. Totally, we model the social influence with K aggregate steps. The user representation at the final step is taken as the user representation modeled from the social influence, denoted as $\boldsymbol{h}_u^K$.

D) Personal interest modeling

As discussed before, the personal interest is also a significant factor influencing the user expertise. And the activation of each interest changes with the specific question context. Therefore, we model the user interest as a context-based dynamic representation. To obtain a precise depiction of the personal interest, the explicit interested topics of each user are incorporated. First, each word in topics is converted into the corresponding word vector by the pre-trained word embedding. The embedding of topics is obtained by applying the mean-pooling over the word embedding of all words in it. The embedding matrix of user's interested topics is represented by $(\boldsymbol{x}_1^t, \boldsymbol{x}_2^t, ..., \boldsymbol{x}_{L_t}^t)$, where L_t is the number of topics. And then we learn the hidden representation of each topic with a dense layer, in which the topic embedding is transformed as:

$$\boldsymbol{h}_i^t = relu(\boldsymbol{W}_t \boldsymbol{x}_i^t + \boldsymbol{b}_t) \tag{4}$$

where $\boldsymbol{W}_t$ and $\boldsymbol{b}_t$ are trainable parameters, $\boldsymbol{x}_i^t$ is the i-th interested topic embedding of user u. And then, let $\boldsymbol{H}_t = (\boldsymbol{h}_1^t, \boldsymbol{h}_2^t, ..., \boldsymbol{h}_{L_t}^t)$ denote the representation of topics after transformation.

Intuitively, the topic which is irrelevant to the current question should have a lower weight of activation. As the user-generated answer to the given question is considered as a personal expression of the user, containing some information of dynamic personal interest. The weight of each topic is calculated based on the specific answer by a multi-layer perceptron (MLP) attention mechanism [24] as:

$$\begin{aligned} \boldsymbol{s}_i &= \tanh(\boldsymbol{W}_a \boldsymbol{r}_a + \boldsymbol{W}_t \boldsymbol{h}_i^t) \\ \boldsymbol{\alpha}_i &= softmax(\boldsymbol{w}_s^T \boldsymbol{s}_i) \end{aligned} \tag{5}$$

where $\boldsymbol{W}_a$ and $\boldsymbol{W}_t$ are trainable attentive matrices and $\boldsymbol{w}_s$ is a trainable attentive vector. We calculate the dynamic personal expertise by weighted sum of the interested topics:

$$\boldsymbol{d}_t = \sum_i \boldsymbol{\alpha}_i \boldsymbol{h}_i^t \tag{6}$$

Furthermore, the overall personal interest of each user is added as a complement, which is expected to avoid the loss of some global information. The fusion is calculated as equation (7) by the gate mechanism:

$$\begin{aligned} \boldsymbol{g} &= \sigma(\boldsymbol{W}_g([\boldsymbol{d}_t, \boldsymbol{m}_t]) + \boldsymbol{b}_g) \\ \boldsymbol{r}_t &= \boldsymbol{g} \odot \boldsymbol{d}_t + (1 - \boldsymbol{g}) \odot \boldsymbol{m}_t \end{aligned} \tag{7}$$

where $\boldsymbol{W}_g$ is the transformation matrix, σ is the sigmoid function, $\boldsymbol{d}_t$ is the dynamic user expertise and $\boldsymbol{m}_t$ denotes the global user interest obtained by applying the mean-pooling over all topic representations.

Subsequently, we concatenate the two representations, which contain information from the social influence and the personal interest, as the final representation of user u:

$$\boldsymbol{r}_u = [\boldsymbol{h}_u^K, \boldsymbol{r}_t] \tag{8}$$

E) Matching layer

For a tuple (q, a, u), the final matching layer takes the representation of question q, answer a and user u as the input and outputs the matching score of it:

$$\begin{aligned} \boldsymbol{h} &= \tanh(\boldsymbol{W}_q \boldsymbol{r}_q + \boldsymbol{W}_a \boldsymbol{r}_a + \boldsymbol{W}_u \boldsymbol{r}_u) \\ f(q, a^q, u^q) &= \tanh(\boldsymbol{W}_s \boldsymbol{h} + \boldsymbol{b}_s) \end{aligned} \tag{9}$$

where $\boldsymbol{W}_q$, $\boldsymbol{W}_a$, $\boldsymbol{W}_u$, and $\boldsymbol{W}_s$ are projection matrices, h denotes the representation of tuple (q, a, u).

F) Training

We choose the max-margin loss function to train our model. Our training case is each tuple $(q, a_i^q, u_i^q, a_j^q, u_j^q)$ constructed from datasets. The answer a_i^q given by user u_i^q has a higher quality (receiving more thumbs-up) than answer a_j^q, and its matching score is expected to be larger. The objective function with hinge loss are designed as follows:

$$L = \sum_{(q, a_i^q, u_i^q, a_j^q, u_j^q) \in \mathcal{S}} \max(0, M + f^-(q, a_j^q, u_j^q) - f^+(q, a_i^q, u_i^q)) \tag{10}$$

where hyper-parameter $0 < M < 1$ is the margin, $\mathcal{S}$ is the set of training samples constructed from dataset,

https://doi.org/10.1017/ATSIP.2020.28

Table 1. Statistics of two datasets (Train/Val/Test)

Datasets	Zhihu-L	Zhihu-S
questions	16,381/2019/2019	2345/289/289
answers	158,169/19,206/19,545	34,125/4100/4103
users	14,759	3381
topics	25,582	18,011

$f^+(q, a_i^q, u_i^q)$ with superscript denotes the score of high quality answers and $f^-(q, a_j^q, u_j^q)$ denotes the score of low quality answers. During training, we minimize the objective function through stochastic gradient descent with the Adadelta [25] optimizer.

IV. EXPERIMENTS

A) Datasets

To evaluate our proposed model, we constructed two real-word datasets Zhihu-L and Zhihu-S by collecting data from Zhihu, a popular cQA site in China. These datasets contain rich user information including users' interested topics and users' following relations. The major difference between the two versions lies in the size and the ratio of questions to answers. Zhihu-L contains more samples where the ratio of questions to answers is about 1:10. Zhihu-S contains fewer samples, where the ratio of questions to answers is about 1:20. The two versions of datasets can contribute to the verification of the performance of our model under different situations. We take all corresponding answers for each question as the candidate answers and their received thumbs-up/down as the ground truth of answer ranking. The answers with more thumbs-up tend to have a higher quality. Different from other datasets, they contain rich user information, including the user's following topics and user's following relation. We remove users who have answered less than five questions. Questions with less than five answers and questions whose best answer has less than 10 thumbs-up are filtered out.

We use 80% questions for training, 10% questions for validation, and the rest 10% questions for test. So, there is no overlap between training set and validation set or test set. Considering the length of all questions and answers in the dataset, we choose 14 as the truncated length of questions and 154 as the truncated length of answers. The statistics of our dataset are shown in Table 1.

B) Parameter settings

In our experiment, questions and answers are tokenized by the Chinese text segmentation tool, i.e. Jieba.[2] We utilized the pre-trained Word2Vec model [22] to generate the word representation for questions, answers and topics. The dimension of the word embedding is 64. The Bi-LSTM hidden size is tuned in 32,64,128,512 and is set to 512 for a better performance. There are two hidden layers in the Bi-LSTM network. What's more, the dimension of social influence representation and personal interest representation are the same as the Bi-LSTM hidden size. The batch size is fixed to 128. The hyperparameter margin m is set to 0.2. The initial learning rate for the Adadelta optimizer is set to 0.001.

As we consider the community-based answer selection as a ranking task, we choose three main ranking metrics for evaluation, that is, precision@1 (*P@1*), mean reciprocal rank (*MRR*) and normalized discounted cumulative gain (*nDCG*). Actually, P@1 and MRR measure the ranking quality of the best answers from different aspects. nDCG is the metric for all candidate answers.

[2] https://github.com/fxsjy/jieba/

C) Evaluation metrics

As we considered the community-based answer selection as a ranking task, we choose three main ranking metrics for evaluation, that is, P@1, MRR, and nDCG. Actually, P@1 and MRR measure the ranking quality of the best answers from different aspects. nDCG is the metric for all candidate answers.

- P@1: This criterion takes the rank position of the best answer into consideration, calculated by

$$P@1 = \frac{|\{q \in Q | rank'_{best} = 1\}|}{|Q|}, \tag{11}$$

  where $rank'_{best}$ is the rank position of the best answer predicted by the algorithm and $|Q|$ is the number of questions. This criterion computes the average number of times that the best answer is ranked on top.
- MRR: Given a set of questions, MRR is the average of the reciprocal ranks of the best answer, given by

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rank'^{i}_{best}}, \tag{12}$$

  where $rank'^{i}_{best}$ refers to the rank position of the best answer for the i-th question predicted by a certain algorithm.
- nDCG: Take question q as an example, nDCG is calculated by

$$nDCG = \frac{DCG}{IDCG},$$

$$where\ DCG = rel_1 + \sum_{i=2}^{|A_q|} \frac{rel_i}{\log_2 i},$$

  where IDCG is the discounted cumulative gain of ideal ordering, $|A_q|$ is the number of candidate answers for question q and rel_i is the relevance between question q and answer at the position i which is indicated by thumbs-up/down value.

https://doi.org/10.1017/ATSIP.2020.28

D) Baselines

We compare our model with other state-of-the-art approaches which are described as follows:

- *BOW* represents question and answer by bag-of-words (BOW) vector. The matching score is calculated based on the BOW representation.
- *CNN* [5] obtains question and answer representations by CNN networks with *k*-max pooling layer on it. The representations are concatenated with additional features for final matching calculation.
- *AI-CNN* [8] calculates the interaction between each pair of representations. The interaction is depicted by 3D tensor and summarized by attentive max-pooling.
- *Multihop-Sequential-LSTM* [7] uses the question vector to deduce the answer vector via multiple steps of sequential attention. Different attention distribution and matching score are summed up for the final matching score.
- *UIA-LSTM-CNN* [9] employs a hybrid attention mechanism for semantic matching of questions and answers. And it models users from user-generated answers.
- *AMRNL* [14] models user by the social network and designs the ranking function for matching based on the deep semantic relevance of question–answer pairs and the users' authority.
- *LatentUVec* [26]: LatentUVec learns user expertise by latent factors. It explicitly models the relevance between the question–user pair and introduces latent user vectors into the representation learning of the answer.
- *AUANN* [10]: This model interactively enhances user engagement and it alleviates the noise issue caused by introducing the user's historical answers by applying an adversarial training module.

E) Overall comparison

The comparison results on two datasets are presented in Tables 2 and 3. From them, we have the following observations. First, all the deep learning methods with distributed representation obtain better performance than traditional BOW models, proving the semantic information is captured by word embedding. Second, AI-CNN and multihop-Sequential outperform CNN, indicating the low-level interaction between question and answer is effective. Third, the user expertise-based models, UIA-LSTM-CNN, AMRNL, LatentUVec, and AUANN perform better than those text matching methods which only take semantic relevance into consideration, showing the effectiveness of user expertise. What's more, modeling users in different ways result in quite different performance. UIA-LSTM-CNN and AUANN modeling user expertise with user-generated answers perform well with sufficient history answers while it has a poor performance on the Zhihu-S dataset where the history answers are not sufficient. AMRNL models users through the social network which is not influenced by the size of dataset. LatentUVec introduces the latent user vector to model the authority-sensitive and the topic-sensitive user expertise. Overall, our model consistently achieves the best performance, since we introducing user interested topics explicitly and consider user expertise from multiple aspects of social influence and personal interest. User information from different aspects is a complement for each other.

Table 2. Performance comparison on Zhihu-L

Models	P@1	MRR	nDCG
BOW	0.1931	0.4101	0.7864
CNN	0.2714	0.4894	0.8158
AI-CNN	0.2828	0.5052	0.8242
Multihop-Sequential	0.2877	0.5089	0.8243
UIA-LSTM-CNN	0.2991	0.5138	0.8245
AMRNL	0.3184	0.5286	0.8276
LatentUVec	0.3368	0.5438	0.8330
AUANN	0.3610	0.5634	0.8423
SCAD	**0.3902**	**0.5828**	**0.8456**

Table 3. Performance comparison on Zhihu-S

Models	P@1	MRR	nDCG
BOW	0.1903	0.3740	0.7224
CNN	0.2802	0.4832	0.7671
AI-CNN	0.3183	0.5109	0.7753
Multihop-Sequential	0.2975	0.4957	0.7724
UIA-LSTM-CNN	0.2975	0.4907	0.7715
AMRNL	0.3217	0.5098	0.7759
LatentUVec	0.3391	0.5434	0.7941
AUANN	0.3252	0.5191	0.7826
SCAD	**0.4221**	**0.6008**	**0.8111**

Table 4. Performance comparison among the variants of our proposed model on Zhihu-L

Models	P@1	MRR	nDCG
SCAD-NoInterest	0.3263	0.5395	0.8361
SCAD-NoSocial	0.3784	0.5784	0.8466
SCAD-Attention	0.3689	0.5703	0.845
SCAD-MeanPooling	0.3744	0.5728	0.8448
SCAD	**0.3902**	**0.5828**	**0.8456**

F) Ablation study

To validate the effectiveness of each component in our model, we conducted experiments with the variants of our model on Zhihu-L and Zhihu-S. The ablation results are reported in Tables 4 and 5. The SCAD-NoInterest removes the dynamic personal interest modeling, and the SCAD-NoSocial removes the social influence module. From the results, we found that both the social influence and the dynamic personal interest can improve the performance.

Furthermore, we also evaluated the necessity of the gate mechanism in the personal interest modeling layer. The SCAD-Attention obtains the dynamic interest of users, while ignores the overall interest. The SCAD-Mean exploits the personal interest in a static way by applying the mean-pooling. From the results, we can observe that the complete model achieves the best performance on both datasets.

https://doi.org/10.1017/ATSIP.2020.28

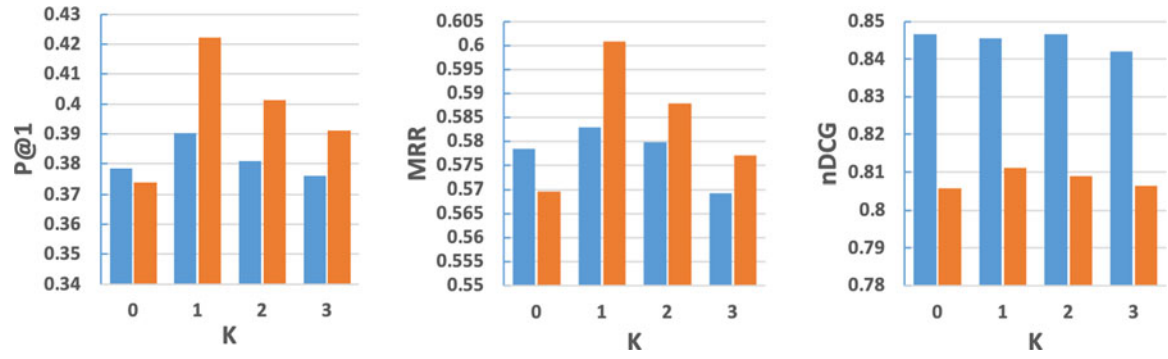


Fig. 3. (a) P@1. (b) MRR. (c) nDCG. The influence of aggregate steps K. The blue is the results of Zhihu-L and the orange is the results of Zhihu-S.

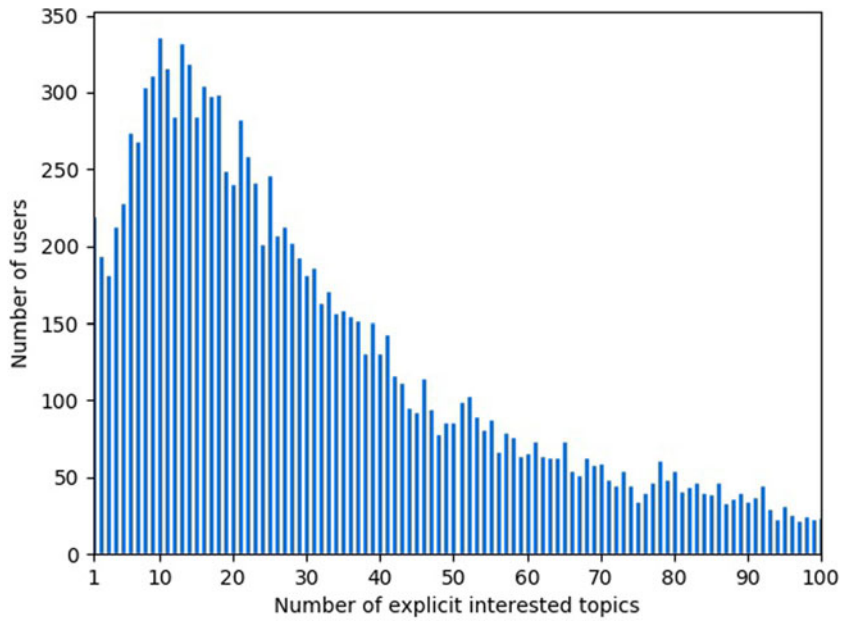


Fig. 4. Statistics of users based on different number of explicit interested topics on Zhihu-L.

Table 5. Performance comparison among the variants of our proposed model on Zhihu-S

Models	P@1	MRR	nDCG
SCAD-NoInterest	0.3391	0.5454	0.796
SCAD-NoSocial	0.3737	0.5696	0.8058
SCAD-Attention	0.3979	0.5834	0.8072
SCAD-MeanPooling	0.4013	0.5919	0.8077
SCAD	**0.4221**	**0.6008**	**0.8111**

This verifies that the gate mechanism is indeed effective for information supplement. It fuses two kinds of interest by controlling the information flow, therefore generating a comprehensive representation of the personal interest for the specific context.

Moreover, the variants with the personal interest achieve more improvements than others without it, which reflects the effectiveness of our creative introduction of the explicit interested topics of users.

G) Parameter analysis

1) The influence of aggregate steps

We carried out experiments over Zhihu-L and Zhihu-S to verify the influence of aggregate steps K. The results are shown in Fig. 3. When $K = 0$, there is no social influence. The influence of aggregate steps K on two datasets are relatively consistent. As the figure shows, rising the social influence from $K = 0$ to $K = 1$, the performance increases for three metrics on both datasets. This demonstrates the effectiveness of the social influence. Apparently, the best performance is achieved at step $K = 1$. When K continues to increase, the performance drops, except for a tiny improvement on MRR at $K = 2$ on Zhihu-S. A possible explanation for this may be that users would not browse answers posted by non-adjacent users directly. The influence delivered from these users through a deep aggregate layer can cause some negative impacts on the user modeling.

https://doi.org/10.1017/ATSIP.2020.28

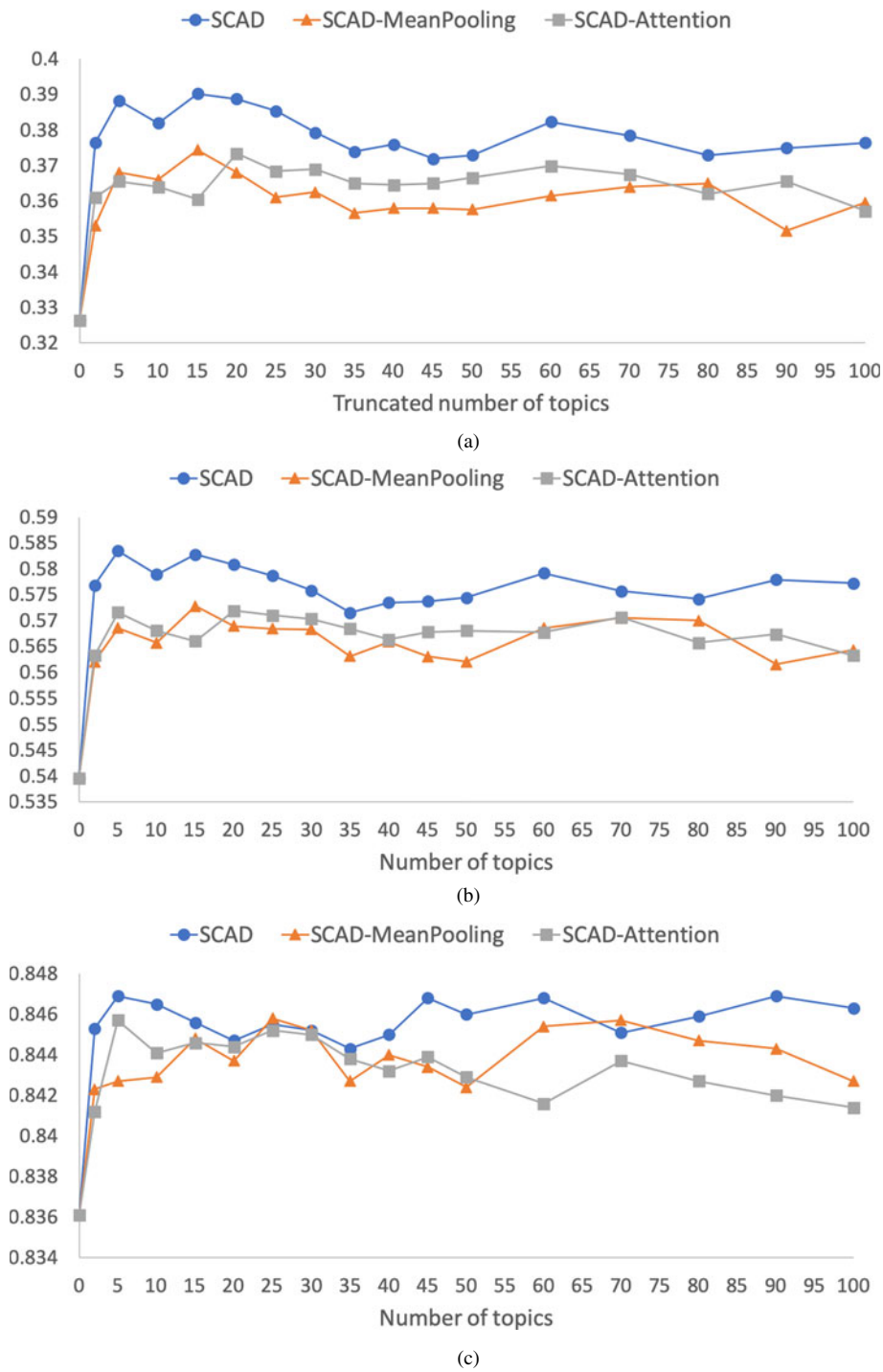


Fig. 5. (a) P@1. (b) MRR. (c) nDCG. The influence of the truncation number of explicit interested topics.

2) THE NUMBER OF EXPLICIT INTERESTED TOPICS

We introduced the explicit interested topics of each user to obtain a precise description of the personal interest. The number of explicit interested topics is different among users. As shown in Fig. 4, it is apparent that most users' explicit interested topic number is in [5,25]. Although there are also some users follow more explicit interested topics, it can be seen that with the number of topics increasing, the amount of users has a clear trend of decreasing. In addition, a small number of users' explicit interested topics are more than 100 which is not presented in the figure.

In our experiments, we selected a fixed number to truncate the number of user's explicit interested topics. The truncation number of explicit interested topics may limit the performance of our model. In other words, few interested topics seemingly insufficient to express the personal interest of a user, while too many interested topics may introduce noise. Thus, we conducted experiments

https://doi.org/10.1017/ATSIP.2020.28

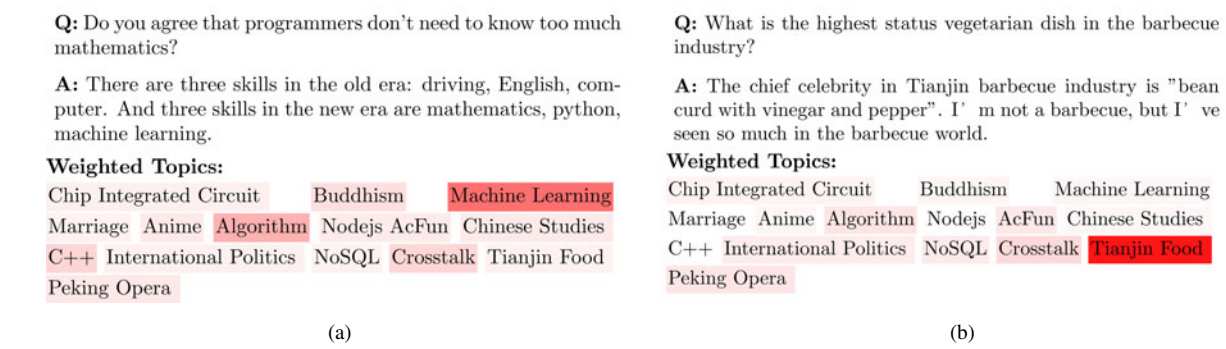


Fig. 6. (a) and (b) An example of topic attention weights in different contexts. The text content in the example is translated from Chinese.

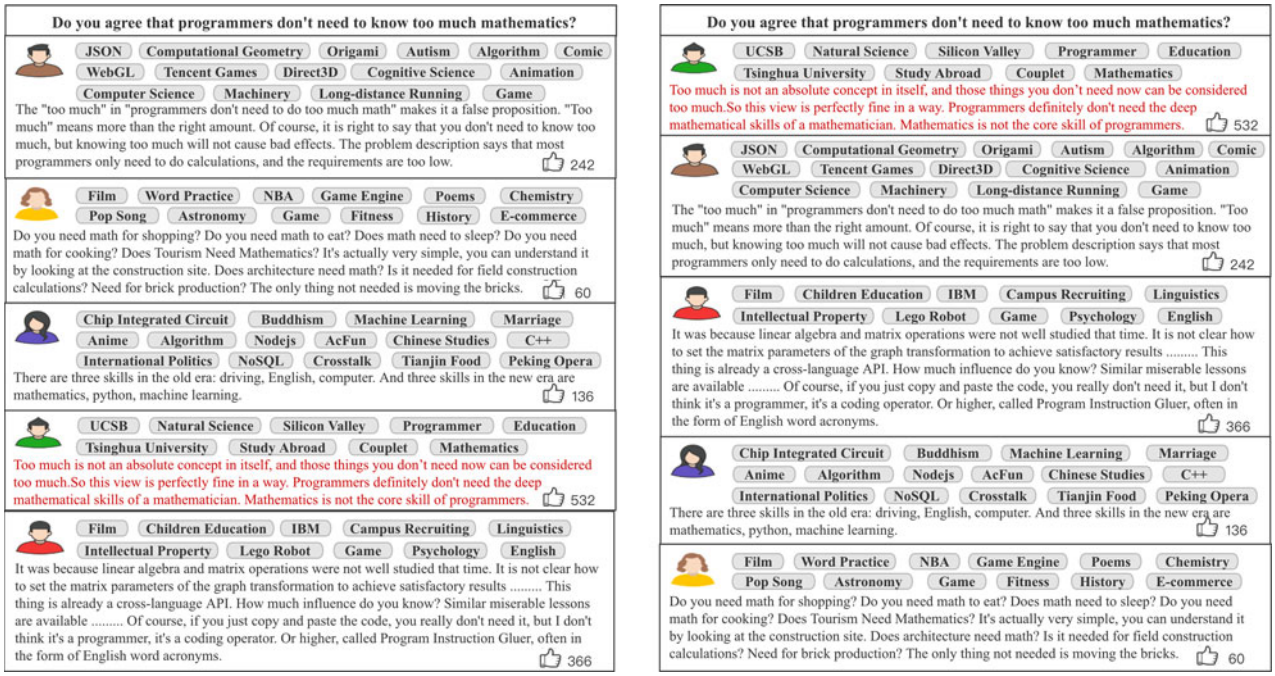


Fig. 7. (a) AMRNL. (b) SCAD. The answer ranking results calculated by the AMRNL and the proposed SCAD. The text content in the example is translated from Chinese.

over Zhihu-L with different truncation numbers of topics to investigate the influence of the truncation number of explicit interested topics. The performances of our model as well as its variants are provided in Fig. 5. An analysis of the results reveals that (1) there are slight differences with different truncation numbers of explicit interested topics for three variants. It confirms that the model performance is not sensitive to the truncation number of user's explicit interested topics. Specifically, on the one hand, it suggests that a small number of interested topics are robust enough to reveal the personal topic of a user. And on the other hand, as we adopted the dynamic and overall personal interest by MLP attention mechanism and mean-pooling, the confusion caused by a large variety of topics are alleviated effectively. (2) Our SCAD model always shows better performance than the other two variants on P@1 and MRR with the same truncation number of topics. And comparing the best results that each model has obtained with different truncation number of explicit interested topics, we found that SCAD achieved the highest score on all three evaluation metrics. This suggests that the gate mechanism for aggregating dynamic and overall personal interest is stable and effective toward different truncation numbers. Taken together, these results further suggest the stability and robustness of SCAD for modeling personal interest. And in the future investigation, we will take the parent-child relationship of topics into account such as clustering topics.

H) Visualization of topic attention weights

To analyze the significance of the topic attention in modeling context-based dynamic user interest, we demonstrated two heat maps in Fig. 6. Figures 6(a) and (b) depict the attention weights of a certain user's explicit interested topics in different contexts respectively. Each topic is marked with various background colors reflecting their

https://doi.org/10.1017/ATSIP.2020.28

attention weights. The stronger the background color is, the more active the topic is for the specific context. Figure 6(a) intuitively presents that the user's interest related to programmers are more active, such as "Machine Learning," "Algorithm," and "C++." Although, as suggested in Fig. 6(b), the user's interest in "Tianjin Food" is activated by the other question which is exactly related to food. Thus, by introducing attention mechanism we precisely modeled the dynamic activation of each interest.

I) Qualitative results

Figure 7 compares the answer ranking results to question "Do you agree that programmers don't need to know too much mathematics" given by AMRNL and SCAD respectively. The groundtruth best answer to the question is marked in red. As can be seen from Fig. 7(b), our proposed SCAD model puts the best answer first. However, AMRNL ranks the best answer after several low-quality ones (Fig. 7(a)). What's more, the ranking list of candidate answers calculated by SCAD is closer to the ground truth. SCAD ranks higher quality answers with more thumbs-up before lower quality answers. Thus, our SCAD model achieve better performance. It further suggests the effectiveness of introducing the explicit interested topics and modeling user expertise from both static and dynamic perspective.

V. CONCLUSION

Most existing studies simply consider community-based answer selection as a text matching task or model users from one aspect. The current study was undertaken to design a network to model users from multiple perspectives and evaluate the effect of user information for answer selection. It jointly consider two factors, the social influence and the personal interest from static and dynamic views. And this study is the first investigation of the explicit user interested topics. Moreover, it measure the changing activation of each topic. The constructed real-world datasets may be of assistance to the future research. The results of extensive experiments demonstrate that our model outperforms the baseline models and verify the effectiveness of different user information.

FINANCIAL SUPPORT

This study is supported by the National Natural Science Foundation of China, No. 61802231; and the Shandong Provincial Natural Science Foundation, No. ZR2019QF001.

REFERENCES

1 Jurczyk, P.; Agichtein, E.: Discovering authorities in question answer communities by using link analysis, in *Proc. of the ACM Conf. on Information and Knowledge Management*, 2007, 919–922.

2 Severyn, A.; Moschitti, A.: Learning to rank short text pairs with convolutional deep neural networks, in *The International ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2015, 373–382.

3 Wang, D.; Nyberg, E.: A long short-term memory model for answer sentence selection in question answering, in *The Annual Meeting of the Association for Computational Linguistics*, 2015, 707–712.

4 Qiu, X.; Huang, X.: Convolutional neural tensor network architecture for community-based question answering, in *Twenty-Fourth International Joint Conf. on Artificial Intelligence*, 2015.

5 Tay, Y.; Phan, M.C.; Anh Tuan, L.; Cheung Hui, S.: Learning to rank question answer pairs with holographic dual LSTM architecture, in *The International ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2017, 695–704.

6 Tan, M.; Dos Santos, C.; Xiang, B.; Zhou, B.: Improved representation learning for question answer matching, in *The Annual Meeting of the Association for Computational Linguistics*, 2016, 464–473.

7 Khanh Tran, N.; Niedereée, C.: Multihop attention networks for question answer matching, in *The International ACM SIGIR Conf. on Research & Development in Information Retrieval*, 2018, 325–334.

8 Zhang, X.; Li, S.; Sha, L.; Wang, H.: Attentive interactive neural networks for answer selection in community question answering, in *the AAAI Conf. on Artificial Intelligence*, 2017, 3525–3531.

9 Wen, J.; Ma, J.; Feng, Y.; Zhong, M.: Hybrid attentive answer selection in CQA with deep users modelling, in *The AAAI Conf. on Artificial Intelligence*, 2018, 2556–2563.

10 Xie, Y.; Shen, Y.; Li, Y.; Yang, M.; Lei, K.: Attentive user-engaged adversarial neural network for community question answering, in *The AAAI Conf. on Artificial Intelligence*, 2020, 9322–9329.

11 Lyu, S.; Ouyang, W.; Wang, Y.; Shen, H.; Cheng, X.: What we vote for? Answer selection from user expertise view in community question answering, in *The World Wide Web Conf.*, 2019, 1198–1209.

12 Fang, H.; Wu, F.; Zhao, Z.; Duan, X.; Zhuang, Y.; Ester, M.: Community-based question answering via heterogeneous social network learning, in *The AAAI Conf. on Artificial Intelligence*, 2016, 122–128.

13 Hu, J.; Qian, S.; Fang, Q.; Xu, C.: Attentive interactive convolutional matching for community question answering in social multimedia, in *The ACM Multimedia Conf. on Multimedia Conf.*, 2018, 456–464.

14 Zhao, Z.; Lu, H.; Zheng, V.W.; Cai, D.; He, X.; Zhuang, Y.: Community-based question answering via asymmetric multi-faceted ranking network learning, in *the AAAI Conf. on Artificial Intelligence*, 2017, 3532–3539.

15 Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G.: The graph neural network model. *IEEE Trans. Neural Networks*, **20** (1) (2009), 61–80.

16 Kipf, T.N.; Welling, M.: Semi-supervised classification with graph convolutional networks, in *the International Conf. on Learning Representations*, 2017.

17 Hamilton, W.L.; Ying, Z.; Leskovec, J.: Inductive representation learning on large graphs, in *The Advances in Neural Information Processing Systems*, 2017, 1024–1034.

18 Wu, L.; Sun, P.; Fu, Y.; Hong, R.; Wang, X.; Wang, M.: A neural influence diffusion model for social recommendation, in *The International ACM SIGIR Conf. on Research and Development in Information Retrieval*, 2019, 235–244.

19 Song, W.; Xiao, Z.; Wang, Y.; Charlin, L.; Zhang, M.; Tang, J.: Session-based social recommendation via dynamic graph attention networks, *The Twelfth ACM International Conf. on Web Search and Data Mining*, 2019.

https://doi.org/10.1017/ATSIP.2020.28

20 Yao, L.; Mao, C.; Luo, Y.: Graph convolutional networks for text classification, in *The AAAI Conf. on Artificial Intelligence*, 2019, 7370–7377.

21 Gao, H.; Chen, Y.; Ji, S.: Learning graph pooling and hybrid convolutional operations for text representations. *The World Wide Web Conference*, 2019.

22 Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J.: Distributed representations of words and phrases and their compositionality, in *The Advances in Neural Information Processing Systems*, 2013, 3111–3119.

23 Hochreiter, S.; Schmidhuber, J.: Long short-term memory. *Neural Comput.*, **9** (8) (1997), 1735–1780.

24 Bahdanau, D.; Cho, K.; Bengio, Y.: Neural machine translation by jointly learning to align and translate, in *The International Conf. on Learning Representations*, 2015.

25 Zeiler, M.D.: ADADELTA: an adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

26 Lyu, S.; Ouyang, W.; Wang, Y.; Shen, H.; Cheng, X.: What we vote for? answer selection from user expertise view in community question answering, in *The World Wide Web Conf.*, 2019, 1198–1209.

Yuchao Liu received the B.E. degree from School of Computer Science and Technology, Shandong University in 2019. She is currently pursuing the M.S. degree with School of Computer Science and Technology, Shandong University. Her research interests include question answering, information retrieval.

Meng Liu received the M.S. degree from Department of Computational Mathematics, Dalian University of Technology in 2016, and the Ph.D. degree from School of Computer Science and Technology, Shandong University in 2019. She is currently a Professor in School of Computer Science and Technology, Shandong Jianzhu University. She has published several papers in the top venues, such as ACM MM, SIGIR and TIP. Her research focuses on multimedia computing and information retrieval.

Jianhua Yin received the B.E. degree from Department of Computer Science and Technology, Xidian University in 2012, and the Ph.D. degree from Department of Computer Science and Technology, Tsinghua University in 2017. He is currently a tenure-track assistant professor in School of Computer Science and Technology, Shandong University. He has published several papers in the top venues, such as ACM SIGKDD, ICDE. His research interests include data mining, machine learning. In addition, he has served as a reviewer for many top conferences and journals.

https://doi.org/10.1017/ATSIP.2020.28