mathematics

MDPI

Article

Feature-Alignment-Based Cross-Platform Question Answering Expert Recommendation

Bin Tang [1,2], Qinqin Gao [3], Xin Cui [3], Qinglong Peng [3] and Xu Yu [3,4,*]

[1] Qingdao Innovation and Development Base, Harbin Engineering University, Qingdao 266000, China
[2] Ship Science and Technology Co., Ltd., Harbin Engineering University, Qingdao 266000, China
[3] College of Information Science and Technology, Qingdao University of Science and Technology, Qingdao 266061, China
[4] Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China
* Correspondence: yuxu0532@qust.edu.cn

Abstract: Community question answering (CQA), with its flexible user interaction characteristics, is gradually becoming a new knowledge-sharing platform that allows people to acquire knowledge and share experiences. The number of questions is rapidly increasing with the open registration of communities and the massive influx of users, which makes it impossible to match many questions to suitable question answering experts (noted as experts) in a timely manner. Therefore, it is of great importance to perform expert recommendation in CQA. Existing expert recommendation algorithms only use data from a single platform, which is not ideal for new CQA platforms with sparse historical interaction and a small number of questions and users. Considering that many mature CQA platforms (source platforms) have rich historical interaction data and a large amount of questions and experts, this paper will fully mine the information and transfer it to new platforms with sparse data (target platform), which can effectively alleviate the data sparsity problem. However, the feature composition of questions and experts in different platforms is inconsistent, so the data from the source platform cannot be directly transferred for training in the target platform. Therefore, this paper proposes feature-alignment-based cross-platform question answering expert recommendation (FA-CPQAER), which can align expert and question features while transferring data. First, we use the rating predictor composed by the BP network for expert recommendation within the domains, and then the feature matching of questions and experts between two domains by similarity calculation is achieved for the purpose of using the information in the source platform to assist expert recommendation in the target platform. Meanwhile, we train a stacked denoising autoencoder (SDAE) in both domains, which can map user and question features to the same dimension and align the data distributions. Extensive experiments are conducted on two real CQA datasets, Toutiao and Zhihu datasets, and the results show that compared to the other advanced expert recommendation algorithms, this paper's method achieves better results in the evaluation metrics of MAE, RMSE, Accuracy, and Recall, which fully demonstrates the effectiveness of the method in this paper to solve the data sparsity problem in expert recommendation.

Keywords: community question answering; expert recommendation; cross-domain recommendation; stacked denoising autoencoder; feature alignment

MSC: 68T07; 68T30

check for updates

Citation: Tang, B.; Gao, Q.; Cui, X.; Peng, Q.; Yu, X. Feature-Alignment-Based Cross-Platform Question Answering Expert Recommendation. *Mathematics* **2023**, *11*, 2174. https://doi.org/10.3390/math11092174

Academic Editors: Sławomir Nowaczyk, Rita P. Ribeiro and Grzegorz Nalepa

Received: 1 March 2023
Revised: 21 April 2023
Accepted: 28 April 2023
Published: 5 May 2023

1. Introduction

In recent years, with the rapid development of Web 2.0 services and the fast growth of knowledge demand, CQA has become an important platform for people to acquire and share knowledge due to its openness and interactivity. Users can share knowledge in the form of asking and answering questions in CQA. Compared with traditional question answering systems, CQA has the advantages of lower time costs, more accurate and comprehensive answers, and higher user activeness [1]. Currently, there are several popular CQA platforms, including Zhihu, Toutiao, Baidu Knows, Yahoo! Answers, and Stack Overflow. Among these platforms, Zhihu, the most popular Chinese Internet CQA platform nowadays, has accumulated more than 44 million total questions and an average of more than 2.5 million active users per month. Toutiao is the latest mobile social CQA platform. Since its inception, it has become a CQA platform with many users and hundreds of thousands of new questions generated every day.

In the CQA, the questioner first posts a text question, and the platform then adds tags or classification information based on the input content and matches suitable users to answer the question. Next, the answerers can give an answer based on their own experience or expertise, and any user who views the answer can evaluate it. Finally, the platform will filter the best answer according to certain strategies and return it to the questioner. At the same time, any user can retrieve questions that have been asked by other users in the platform and filter out the questions or answers that are closest to the user's demand. From the operation mechanism of CQA, it can be observed that CQA usually involves three basic elements: questions, answers, and users. Users play four different roles in CQA: questioner, answerer, evaluator, and searcher.

With the popularity of CQA in real life, the number of questions and users in the platform is rapidly increasing. Although users can freely access and answer questions based on their experience and expertise, the number of user answers is limited in the face of the large number of questions accumulated in the platform, which usually makes it impossible to find suitable experts to answer such questions in time. Specifically, we can analyze it from three perspectives: the questioner, the answerer, and the platform:

- From the perspective of the questioner: It is the urgent demand of the questioner to get high-quality answers quickly, but it is very time-consuming to select competent and reliable experts from a large number of users;
- From the perspective of the answerer: It often takes lots of time and effort to find the questions that they are interested in and capable of answering, which affects the user's experience and motivation;
- From the perspective of the platform: It is very difficult to balance the needs of both questioners and answerers.

Facing the information overload problem in CQA platforms, many scholars have investigated expert recommendation algorithms to implement feature matching between questions and experts. The research includes topic-based expert recommendation [2–7], where new questions asked by users in the community are considered as question texts, while the questions and answers given by users are combined into user texts. The topic model and its optimization model are used to mine the correlation between two texts. Expert recommendation based on link analysis [8–14] constructs a question–answer relationship directed graph based on the historical interaction behavior between users in the community and then performs link analysis on the directed graph and calculates the authority of each user. Neural-network-based expert recommendation [15–20] encodes higher-level questions and feature representations of expert texts with the help of word2vec and graphs and then extracts features by convolutional neural networks and recurrent neural networks.

Most of the above expert recommendation algorithms only exploit the interaction features between users and questions in the platform, which is suitable for CQA communities with long running time and rich historical interaction data but not for new CQA platforms. Facing this situation, we design the question answering expert recommendation method for the new CQA platforms. On the one hand, we can alleviate the problem of sparse historical interaction data and the small number of users and questions in the target platform by transferring the rich historical data from the source platform to the target platform; on the other hand, we solve the problem that the feature composition of experts and questions is inconsistent in different platforms by feature alignment.

Based on the information mentioned above, this paper proposes a feature-alignment-based cross-platform question answering expert recommendation algorithm (FA-CPQAER). For the convenience of description, consistent with the concept in transfer learning, the target platform and the source platform are termed the target domain and the source domain. First, a BP-network-based rating predictor for expert recommendation is trained in the source and target domains, respectively. Then, the feature vectors of experts and questions are mapped to the same dimension using an SDAE in two domains. Next, the feature matching of questions and experts between domains is calculated by similarity. Finally, suitable experts are recommended for new questions in the target domain by the recommendation models of the target and source domains. Extensive experiments conducted on the Toutiao community and Zhihu community show that the algorithm in this paper has obvious advantages and significantly improved recommendation performance. The major contributions of this paper are as follows:

- A cross-platform expert recommendation model is proposed to improve recommendation performance in new CQA by transferring rich information from the source domain to the target domain, which solves the data sparsity problem of new CQA and effectively improves recommendation performance in the target domain;
- A feature dimension alignment algorithm based on an SDAE is proposed to solve the problem that the similarity cannot be calculated due to inconsistent feature dimensions and data distributions in two domains;
- Adequate experiments are conducted on two real datasets, and the results show that FA-CPQAER significantly outperforms existing expert recommendation algorithms.

Section 2 of this paper reviews the research works related to expert recommendation. Section 3 proposes a feature-alignment-based cross-platform question answering expert recommendation algorithm. Section 4 conducts extensive experiments on the Toutiao and Zhihu datasets, giving a series of comparative results and analyses. Section 5 concludes the full paper.

2. Related Work

The cross-platform expert recommendation in CQA in this paper is an application of traditional cross-domain recommendation algorithms in the CQA community, so we first review the work related to classical cross-domain recommendation algorithms and then introduce the current research related to expert recommendation in CQA.

2.1. Cross-Domain Recommendation

In recent years, recommendation systems have been widely used in e-commerce and online social media to address the growing information overload problem. The collaborative filtering algorithm is one of the most widely used methods in recommendation systems, with the advantages of generality, good interpretability, and no content information features. Researchers have proposed a series of cross-domain recommendation algorithms based on collaborative filtering (CF). Berkovsky et al. [21] proposed a content-based cross-domain collaborative filtering method. Singh et al. [22] proposed a collective matrix factorization (CMF) model, which decomposed the interaction matrix of multiple domains with common entities and shared the feature factors of these entities in each domain. Hu et al. [23] proposed a cross-domain triadic factorization (CDTF) model based on tensor factorization to learn the triadic user–item–domain relationship. Li et al. [24] proposed the codebook-based transfer (CBT) method to learn user rating patterns of items from the source domain, which is shared among different domains, and then transfer the information in the codebook to reconstruct the rating matrix of the target domain. Li et al. [25] proposed the rating-matrix generative model (RMGM) to share knowledge among multiple domains in the form of potential cluster-level ratings. Gao et al. [26] proposed the cluster-level latent factor model (CLFM), which not only learns the cluster-level ratings shared by all domains but also learns the pattern at the cluster level specific to each domain at the same time.

2.2. Expert Recommendation

The number of historical question and answer records, the level of users' expertise, the relationship between users, and the description of question texts all affect the effectiveness of recommendations in CQA. In early studies, mainly domain-specific expert recommendations were performed using information retrieval techniques [27], which used vectors to represent users' expertise and performed poorly in expert search. Later studies have modeled experts by constructing topic models. Li et al. [2] further considered user availability, weighted it with the user's relevance and answer quality to represent the user's expertise for a given question. Du et al. [3] proposed a topic model using a two-parameter Dirichlet process to discover the hierarchy of topics. Li et al. [4] proposed a category-sensitive QLL model, where once a question is classified, the task is to find the user that is most likely to answer the question in the category. Xu et al. [5] proposed a dual-role model (DRM), modeling the user's role as a requester and responder, respectively, and deriving the corresponding probabilistic model based on PLSA, in addition to modeling the relationship between the responder and requester to obtain better expert recommendation performance. Yang et al. [6] proposed a topic expertise model (TEM), which jointly learns the topic and expertise by a textual content model and link structure analysis. Sahu et al. [7] used only tag information to infer users' topic interests, but its effectiveness depends on the accuracy and availability of tags.

The above methods are mainly based on text for expert recommendation, but there are also ranking algorithms based on link analysis that use the idea of page ranking to rank experts in the network. Zhang et al. [8] proposed an ExpertiseRank model, which argues that answerers usually have better expertise on a particular topic than requesters and then propagates users' expertise scores through the relationship of questions and answers in the user–user network. Liu et al. [9] made comprehensive use of various relationships between the requester and responder, as well as the relationship between the best responder and the other responder to find users and recommend experts. Zhu et al. [10,11] also considered the category relevance of questions and user permissions for ranking to improve the performance of expert recommendation in the extended category link graph. Liu et al. [12] performed expert recommendation, combining user–topic relevance (calculated by cosine similarity) and the user's reputation with user-specific authority obtained through link analysis. Zhao et al. [13] performed expert recommendation from the perspective of matrix factorization, based on the rating matrix of users' answered questions, to estimate the quality of users' answers to new questions and used the users' social networks to infer the users' expertise, thus improving the quality of expert recommendations. He et al. [14] considered that knowledge communities are dynamic and thus users' topic preferences and expertise levels change over time and then proposed a modeling approach that incorporates temporal information, user topic preferences, and expertise to recommend appropriate experts for users who submit questions.

In recent years, with the continuous development of deep learning techniques, some researchers have applied deep learning to CQA recommendation. Zhao et al. [15] proposed a ranking metric network learning framework that explores the quality ranking of users' relevance to questions and users' social relationships. Huang et al. [16] proposed a framework for finding experts in collaborative networks that summarizes text blocks from a semantic perspective and infers knowledge domains by clustering pre-trained word vectors, using a graph-based clustering approach to extract knowledge domains and matrix factorization techniques to identify shared latent factors. Wang et al. [17] proposed an expert recommendation model that uses a convolutional neural network, which represents the question and user's interest preferences as a word embedding vector, and then the word embedding is used as the input feature of the convolutional neural network for learning. Chen et al. [18] applied a random walk learning method based on a neural network with RNN to match similarities between new and historical questions. Sun et al. [19] modeled the interaction between different objects (questions, questioners, and answerers) by FMs, which takes advantage of users' historical question answering behavior. Jian et al. [20]

proposed an undirected heterogeneous graph that encodes users' past question answering activities and textual information of the questions. This model allows embedding nodes in the graph using higher-order graph structure and content information and simultaneously routing questions.

All of the above methods perform expert recommendation in a single platform only, with a limited number of question records and insufficiently detailed user expertise, which have a significant impact on the speed and accuracy of expert recommendation.

3. The Proposed Algorithm

In this paper, we propose a cross-platform expert recommendation algorithm for community question answering. The framework of the method is shown in Figure 1. The details of how each part is implemented will be given in the next subsections. The framework in this paper consists of four main parts: question and expert feature representation, rating prediction based on the target and source domains, feature matching based on the target and source domains, and expert list merging.

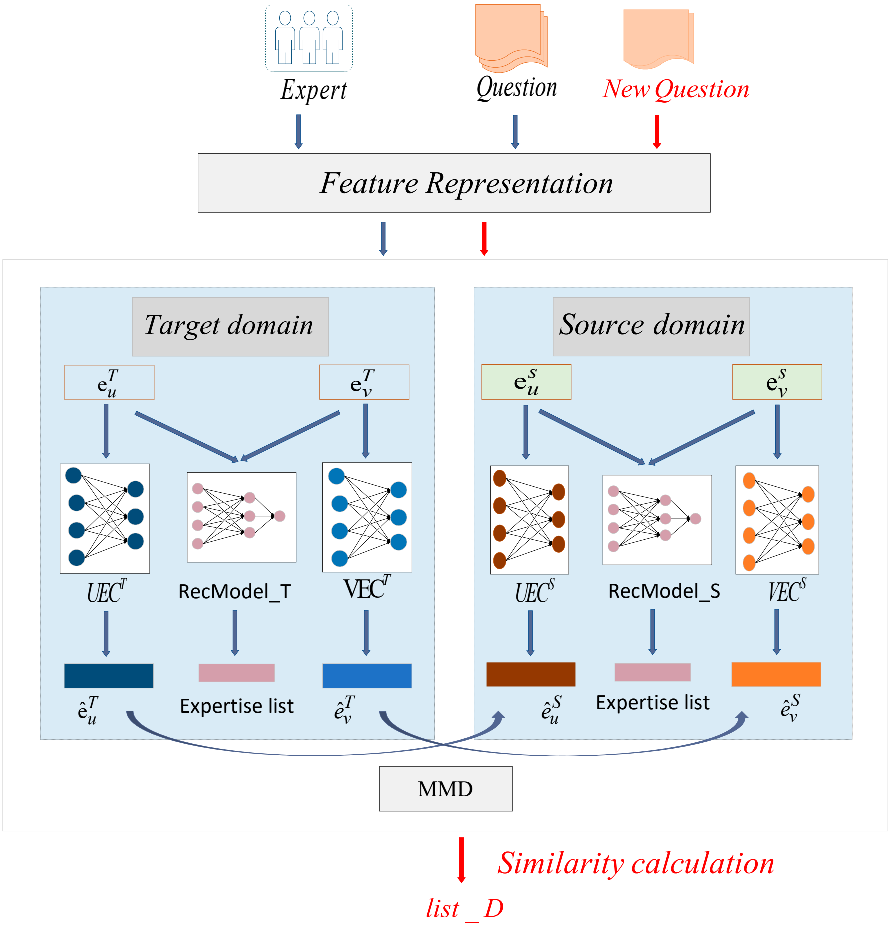

Figure 1. The framework of cross-platform expert recommendation algorithm in CQA.

3.1. Problem Definition

Let the sets of experts in the target and source domains be U^T and U^S, and the lengths of both are m^T and m^S, respectively. The datasets of existing questions are V^T and V^S, and the lengths are n^T and n^S, respectively. The dataset of cold-start questions in the target domain is V^N. The rating matrices of the target and source domains are $R^T \in \mathbb{R}^{m^T \times n^T}$ and $R^S \in \mathbb{R}^{m^S \times n^S}$, respectively. The goal of this paper is to recommend suitable experts for new questions in a newly committed CQA (target domain) with the help of an information-rich CQA (source domain).

In this section, we also introduce the used notations throughout the article. We summarize the used notations in Table 1.

Table 1. The notations used in this paper.

Notation	Explanation
U^T	expert dataset in the target domain
U^S	expert dataset in the source domain
m^T	length of expert dataset in the target domain
m^S	length of expert dataset in the source domain
V^T	question dataset in the target domain
V^S	question dataset in the source domain
n^T	length of question dataset in the target domain
n^S	length of question dataset in the source domain
V^N	new question dataset in the target domain
$R^T \in \mathbb{R}^{m^T \times n^T}$	rating matrices of the target domain
$R^S \in \mathbb{R}^{m^S \times n^S}$	rating matrices of the source domain
v^T, v^S, v^N	question
u, u^S, u^T	expert
e_v^T, e_v^S, e_v^N	vector representation of the question
e_u^T, e_u^S	vector representation of the expert
$\tilde{e}_v^T, \tilde{e}_v^S, \tilde{e}_v^N$	vector representation of the question encoded by SDAE
$\tilde{e}_u^T, \tilde{e}_u^S$	vector representation of the expert encoded by SDAE
$VEC^T(\cdot)$	question encoder for the target domain
$VEC^S(\cdot)$	question encoder for the source domain
$UEC^T(\cdot)$	expert encoder for the target domain
$UEC^S(\cdot)$	expert encoder for the source domain

3.2. Question and Expert Feature Representation

The questions and experts in CQA contain text, categorical, and numeric features. Tables 2 and 3 give detailed information about the question and expert features in the target domain. The source domain contains richer information, which makes the feature composition and feature dimensions of questions and experts inconsistent in different communities. The question and expert matching is only possible through feature alignment, which is described in Section 3.4.

Table 2. Question feature information in the target domain.

Feature	Data Format	Description
Title	Text	Title of the question
Question description	Text	Description of the question details
Topic	Categorical	The topic to which the question belongs
Number of likes	Numeric	Number of likes for the question
Number of responses	Numeric	Number of responses to the question

Table 3. Expert feature information in the target domain.

Feature	Data Format	Description
Expert description	Text	Description of the expert
Interest label	Categorical	Topics concerned by the user

The question and expert feature vectors are generated in the following steps:

- Text features: For text features such as the title and the description of questions and users, such information is a piece of textual description information. The title and question description usually represent the subject area and semantic scope to which the question belongs, and the user description contains the expert's expertise reflecting the expert's competency. In order to better represent these textual features, they are transformed into digital vector representations. Taking the problem text feature as an example, we first synthesize the title and question description into a text and then use the doc2vec [28] model to obtain a vector representation of the question text.

- Categorical features: Categorical features (topics, interest labels) are taken from the labels provided in the platform. We convert them into discrete vectors using one-hot coding. Taking interest labels as an example, each label represents a feature, e.g., if the total number of labels is 150 and the interest label owned by the expert is A, it is represented as [1, 0, . . . , 0]; if the labels owned by the expert are B and C, the vectorization is represented as [0, 1, 1, 0, . . . , 0].
- Numeric features: It is necessary when the features of datasets have different ranges of values among them. To eliminate the effect of different magnitudes, the numerical features are normalized and mapped to [0, 1] with the following formula:

$$x^* = \frac{x - \min}{\max - \min} \tag{1}$$

where min is the minimum value of the numeric feature, and max is the maximum value.

The above three processed features are stitched together to obtain the vectorized representation of the question v_j as $e_{v_j} \in \mathbb{R}^{d_v}$. Similarly, the vectorized features of the expert u_i are represented as $e_{u_i} \in \mathbb{R}^{d_u}$.

- Meanwhile, the number of collections received by the expert answering the question is used as the rating in the source domain, and the number of likes is used as the rating in the target domain. Normalization is performed using the following formula:

$$\hat{x} = \frac{x - \mu}{x_{\max} - x_{\min}} \times 5 \tag{2}$$

where $x_{\max}$ is the maximum value of the score, $x_{\min}$ is the minimum value, and μ is the mean value. The rating matrices for the target and source domains are obtained after processing as $R^T \in \mathbb{R}^{m^T \times n^T}$ and $R^S \in \mathbb{R}^{m^S \times n^S}$, respectively.

3.3. Rating Prediction Based on the Target and Source Domains

BP networks tend to have better fitting ability, so the rating prediction of expert recommendation is performed by a multi-layer BP network to obtain a list of expert recommendations of length N in this paper. The structure diagram of the BP network with L layers is shown in Figure 2.

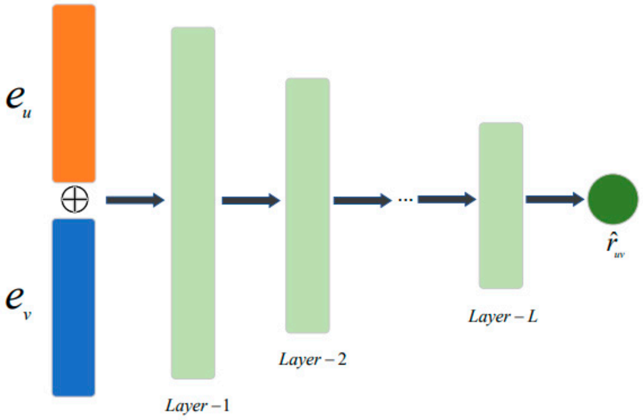

Figure 2. Structure of BP network with L-layer.

A BP network usually consists of an input layer, an output layer, and several hidden layers, whose most basic component is the neuron. The basic operating logic of a BP network is as follows: a linear operation by weights and bias terms is applied to the activation function to obtain the value on the next neuron connected to this neuron. Then, the cumulative error is minimized according to the error sum of each output neuron in the output layer, and the connection weights are adjusted in the reverse direction to achieve a stable network convergence. In this paper, a BP network with one hidden layer is used, and the propagation process includes two steps: First, forward propagation, that is, the vectorized representation of questions and experts, is processed by the hidden layer, and the output value is calculated by the *relu* activation function, which is a non-linear transformation function. Then, the predicted output value is calculated after calculating the input weighting by the output layer, which is compared with the actual output. If the actual and predicted outputs are not equal, the backward propagation is entered. Second, during the backward propagation process, i.e., after calculating the cumulative error, the new connection weights are computed by the chain derivative method. The above process is iterated to minimize the error signal. The calculation process of the BP network is as follows:

$$h^{(0)} = e_u \oplus e_v \tag{3}$$

$$h^{(1)} = relu\left(h^{(0)}\mathbf{W}^{(1)} + b^{(1)}\right) \tag{4}$$

$$h^{(2)} = relu\left(h^{(1)}\mathbf{W}^{(2)} + b^{(2)}\right) \tag{5}$$

$$\dots \tag{6}$$

$$h^{(L)} = relu\left(h^{(L-1)}\mathbf{W}^{(L)} + b^{(L)}\right) \tag{7}$$

$$\hat{r}_{uv} = h^{(L)} \tag{8}$$

where *relu* is the non-linear activation function, and **W** and b are the weight matrix and bias. The optimization objectives of the rating prediction network are as follows:

$$\min \sum_{(u,v)\in D} (r_{uv} - \hat{r}_{uv})^2 + \lambda\|\theta\|^2 \tag{9}$$

where D is the dataset, λ is the regularization coefficients, $\theta = \{\mathbf{W}, b\}$ is the set of parameters, and $\|\cdot\|$ is the F-norm of the matrix. This part uses *Adam* to optimize the model parameters. After the model training is completed, the vector representation of the new question is associated with the vector representation of existing experts, which is then fed into the network to predict the scores and recommend the N experts with the highest predicted scores for this new question. Using the above model, we can train the expert recommendation model for the target domain *RecModel_T* and the source domain *RecModel_S*. The input to the recommendation model is a vectorized representation of questions and experts, and the output is a list of expert recommendations of length N.

3.4. Feature Matching Based on the Target and Source Domains

In this paper, we use similarity calculation to achieve feature matching between the target domain and the source domain. First, we find similar questions existing in the source domain for new questions in the target domain. Then, we recommend experts for similar questions with the help of the source domain recommendation model. Finally, we determine the recommended list of experts in the target domain by calculating the similarity between the experts in the recommended list of the source domain and the experts in the target domain. However, the feature dimensions and data distributions are not consistent in the target and source domains, which makes the similarity calculation impossible. So we solve this problem by a stacked denoising autoencoder in this paper.

3.4.1. Stacked Denoising AutoEncoder

In this paper, a stacked denoising autoencoder (SDAE) can map the features of questions in both domains to the same dimension and handle the features of experts in the same way. An SDAE consists of multiple stacked denoising autoencoders, where a single denoising autoencoder is composed of an encoder and a decoder. The encoder is calculated as follows:

$$y = \mathrm{LeakyRelu}\left(W^{Encoder}\widetilde{x} + b^{Encoder}\right) \tag{10}$$

where $\widetilde{x} \sim N\left(x, \sigma^2 I\right)$ is the feature representation of encoder input x after adding Gaussian noise. $W^{Encoder}$ and $b^{Encoder}$ are the weight and bias of the decoder, respectively, and LeakyRelu is the activation function. Similarly, the decoder is calculated as follows:

$$z = \mathrm{LeakyRelu}\left(W^{Decoder}y + b^{Decoder}\right) \tag{11}$$

where $W^{Decoder}$, $b^{Decoder}$ are the weight and bias of the decoder, respectively, LeakyRelu is the activation function, and z is the reconstructed representation of encoder input x. The loss function of the denoising autoencoder is as follows:

$$L_{DAE}(x,z) = \frac{1}{2}\sum_{i=1}^{d}\left|\left|x^{(i)} - z^{(i)}\right|\right|^2 \tag{12}$$

The structure of the stacked denoising autoencoder is shown in Figure 3. The pre-training schematic is shown in Figure 4.

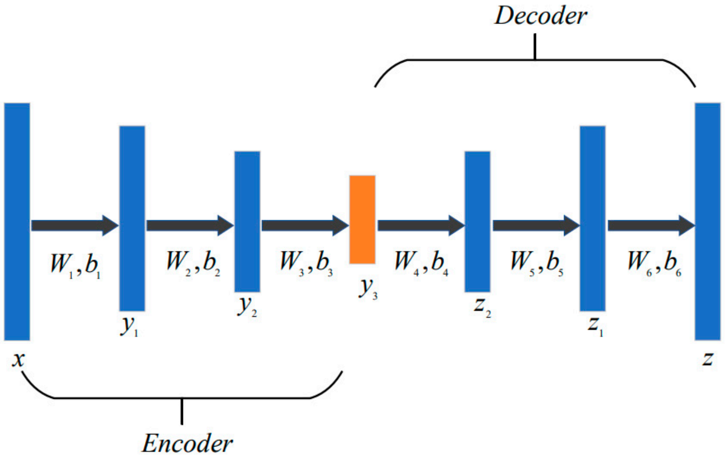

Figure 3. Stacked Denoising AutoEncoder.

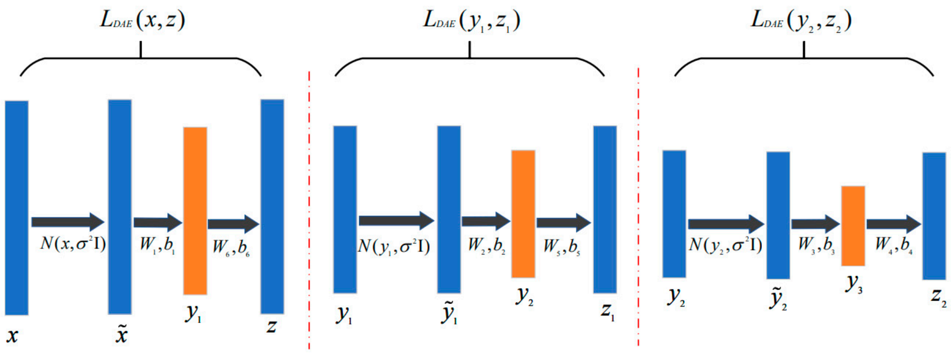

Figure 4. Schematic diagram of pre-training.

Because the question and expert features in the target domain and source domains are inconsistent, the inherent differences are significant. Although we map questions (experts) to the same feature dimension, it is not guaranteed to learn features with the same semantics and share similar distributions. Therefore, we introduce the maximum mean discrepancy (MMD) loss to constrain the distribution difference of features in the target domain and source domains, which adopts the method of calculating the mean distance between the two distributions. The training process of the SDAE uses layer-by-layer training with fine-tuning and uses Adam to optimize the model parameters. Taking the expert SDAE as an example, we pre-train the expert features in the target and source domains separately and then fine-tune them simultaneously. The data distributions between the two domains are aligned by minimizing the MMD between the hidden-layer feature distributions of the source and target domains. MMD is defined as follows:

$$\mathrm{MMD}(y_S, y_T) = ||\frac{1}{m^S}\sum_{i=1}^{m^S} y_S^{(i)} - \frac{1}{m^T}\sum_{i=1}^{m^T} y_T^{(i)}||_2^2 = tr\left(yMy^T\right) \tag{13}$$

where $y = (y_S, y_T)$, m^S, and m^T denote the number of experts in the source and target domains, respectively. M is the coefficient matrix, and M_{ij} denotes the i-th row and j-th column element in M, which is calculated as follows:

$$M_{ij} = \begin{cases} \frac{1}{\left(m^S\right)^2}, & i \le m^S, j \le m^S \\ \frac{1}{\left(m^T\right)^2}, & i > m^S, j > m^S \\ -\frac{1}{m^S \times m^T}, & otherwise \end{cases} \tag{14}$$

Thus, the final loss function for the fine-tuning phase is as follows:

$$L = (L_{DAE}^S + L_{DAE}^T + \beta \mathrm{MMD}) \tag{15}$$

where L_{DAE}^S and L_{DAE}^T denote the feature reconstruction loss in the source and target domains, respectively, and β is the weight.

Finally, we can get the encoder $VEC^T(\cdot)$ and $UEC^T(\cdot)$, which map the question and user features to the low-dimensional vector in the target, respectively. Similarly, the encoder $VEC^S(\cdot)$ and $UEC^S(\cdot)$ can be obtained in the source domain. $UEC^T(\cdot)$ and $UEC^S(\cdot)$ can map the user features to the same dimension d_u; $VEC^T(\cdot)$ and $VEC^S(\cdot)$ can map the question features to the same dimension d_v.

3.4.2. Question and Expert Feature Matching

With the SDAE from the previous subsection, we can obtain the feature vector of the questions and users for the same dimension in the target and source domains. Then, we find K questions that are similar to the target question in the source domain by calculating the cosine similarity between questions in both domains. The cosine similarity is calculated as follows:

$$\text{similarity}\left(v_i, v_j\right) = \frac{f_{v_i} \bullet f_{v_j}}{|f_{v_i}| \times \left|f_{v_j}\right|} \tag{16}$$

where v_i denotes the set of questions in the target domain, v_j represents the set of questions in the source domain, and the higher value indicates a higher similarity between questions.

After getting the list of similar K questions in the source domain, we use the recommendation model RecModel_S in Section 3.3 to recommend experts in the source domain for these K questions and get a recommendation list of length KN. Then, we sort the scores of the experts in this recommendation list, taking top-N experts to form the recommendation list *list_B*. Next, using cosine similarity calculation to find the most similar expert in the target domain for each expert in the expert recommendation *list_B*, we finally get the expert recommendation list *list_B*. Before the calculation, we use the SDAE to map the expert features in the source and target domains to the same dimension.

3.5. Expert List Merging

After obtaining the expert lists *list_A* and *list_C*, we merge them to obtain the final expert recommendation list of length N. The merging rules are as follows:

- The shared experts of *list_A* and *list_C* are added directly to the expert recommendation list *list_D*, which are arranged in the order of expert recommendation list *list_C*.
- If there are still less than N experts in the recommendation list *list_D* after the above processing, half of the total number of missing experts are taken from the recommendation list *list_C*. If the number is not divisible, the number of experts taken from *list_C* is rounded up, and the rest are taken from *list_A*. The experts are kept in the original order. For example, if *list_A* = {u_1, u_3, u_{10}, u_2, u_8}, *list_C* = {u_{10}, u_7, u_1, u_4, u_{15}}, then *list_D* = {u_{10}, u_1, u_7, u_4, u_3}.

3.6. Algorithm Description

Detailed Feature-alignment-based cross-platform expert recommendation algorithm is summarized in Algorithm 1.

Algorithm 1 Feature-alignment-based cross-platform expert recommendation

```
Input: expert dataset in the source and target domain $U^S$, $U^T$
       question dataset in the source and target domain $V^S$, $V^T$
       new questions dataset asked by the target domain $V^N$,
       questions–expertise rating matrix $R^T$, $R^S$
       the maximum number of iterations T.
Output: a list of experts for a new question $list\_D$.
Vectorize the question and expert representation of the target domain as $e_v^T$, $e_u^T$
Vectorize the question and expert representation of the source domain as $e_v^S$, $e_u^S$
Vectorize the question representation of the target domain as $e_v^N$
Construct rating matrix $R^T$, $R^S$
For $i = 1$ to $T$ do
    Feature stitching and construct training sample $(e_u \oplus e_v, r_{uv})$
    BP network $\rightarrow RecModel\_S, RecModel\_T$
    $e_v^T, e_v^S \rightarrow$ SDAE $\rightarrow VEC^T(\cdot), VEC^S(\cdot)$,   $e_u^T, e_u^S \rightarrow$ SDAE $\rightarrow UEC^T(\cdot), UEC^S(\cdot)$
    Minimize the loss of BP network: $\sum_{(u,v)\in D}(r_{uv} - \hat{r}_{uv})^2 + \lambda||\theta||^2$
    Minimize the loss of SDAE:
    $L_{DAE}(x,z) = \frac{1}{2}\sum_{i=1}^{d}\left|\left|x^{(i)} - z^{(i)}\right|\right|^2$   $L = L_{DAE}^S + L_{DAE}^T + \beta$MMD
End for
For $v^N$ in $V^N$
    $e_v^N \rightarrow RecModel\_T \rightarrow list\_A$
    $e_v^N \rightarrow VEC^T(\cdot) \rightarrow \widetilde{e}_v^N$ , $e_v^S \rightarrow VEC^S(\cdot) \rightarrow \widetilde{e}_v^S$
    Similarity calculation $(\widetilde{e}_v^N, \widetilde{e}_v^S) \rightarrow \{\widetilde{e}_{v_i}^S\}_{i=1}^k$
    For $\widetilde{e}_v^S$ in $\{\widetilde{e}_{v_i}^S\}_{i=1}^k$
        $\widetilde{e}_{v_i}^S \rightarrow RecModel\_S \rightarrow \{e_{u_i}^S\}_{i=1}^N$
      Aggregate $\{e_{u_i}^S\}_{i=1}^N \rightarrow \{e_{u_i}^S\}_{i=1}^{kN} \rightarrow list\_B$
    For $e_u^S$ in $list\_B$
            $e_u^S \rightarrow UEC^S(\cdot) \rightarrow \widetilde{e}_u^S$, $e_u^T \rightarrow UEC^T(\cdot) \rightarrow \widetilde{e}_u^T$
        Similarity calculation $(\widetilde{e}_u^S, \widetilde{e}_u^T) \rightarrow \{\widetilde{e}_{u_i}^T\}_{i=1}^k$
    Aggregate $\{\widetilde{e}_{u_i}^T\}_{i=1}^k \rightarrow \{\widetilde{e}_{u_i}^T\}_{i=1}^{kN} \rightarrow list\_C$
    Merge $list\_A$ and $list\_C \rightarrow list\_D$
End
```

3.7. Time Complexity Analysis of the Algorithm

Cross-domain expert recommendation for CQA includes a BP network, an SDAE, and similarity calculation. Let T, m, and f denote the maximum number of iterations in the BP network, the number of training samples, and the number of nodes in the hidden layer, respectively. The worst case is when the algorithm is executed T times. Therefore, the time complexity of the BP network is $O\ (T \times m \times f)$. For the SDAE, let h_1, h_2, and h_3 denote the number of neurons in the middle layer, Q is the number of questions or users, and U is the number of iterations. The time complexity of the SDAE is $O\ (Q \times U \times (h_1 + h_2 + h_3))$. The time complexity of similarity calculation is $O\ (Q_T \times (Q_S + U_T))$, where Q_T and Q_S denote the number of questions in the target and source domains, and U_T denotes the number of users in the target domain. In summary, the time complexity of this algorithm is $O\ (T \times m \times f + Q \times U \times (h_1 + h_2 + h_3) + Q_T \times (Q_S + U_T))$.

4. Experiments

In this section, we conduct experiments on the Toutiao and Zhihu datasets to verify the effectiveness of the proposed cross-domain expert recommendation algorithm for CQA and compare the proposed model with other expert recommendation algorithms. The environment configuration used in the experiments is 2.70 GHz Intel(R) Core(TM) i5-11,400 H CPU, 16 GB RAM, and Windows 11. The relevant algorithms are implemented in Python 3.0 with the scikitlearn and Keras libraries.

4.1. Experimental Data

This experiment uses the Toutiao dataset (from the 2016 ByteCup International Machine Learning Competition) and the Zhihu dataset (from the 2019 Zhiyuan-Kanzan Cup Expert Discovery Algorithm Competition).

The Zhihu dataset includes information about questions, user profiles, user response records, and user acceptance of question invitations. The time span of this corpus is 2 months, and there are 100,000 question-related topics. Each user also provides multiple features. According to the needs of the proposed algorithm, we select data such as interest topics, gender, active frequency, and salt value as expert features and data such as the title, question description, topic, number of likes, number of collections, and number of comments as question features. The Toutiao dataset includes information such as expert tag data, question data, and question distribution data. Similarly, we select the topic and expert description as expert features and the title, question description, topic, number of likes, and number of responses as question features.

The dataset used in this paper was not subjected to any cleaning for the training and testing data except for operations such as case conversion, the removal of punctuation, and disabling words for the original text. The following data processing was performed before model validation:

(1) For the Zhihu dataset, the questions with no answers and associated invitations were removed, while users with salt values greater than 700 in the dataset were selected as experts.

(2) For the Toutiao dataset, information with no question description and no expert description in the dataset was removed.

Table 4 shows the comparison results between the Toutiao dataset and the Zhihu dataset. From the table, it can be seen that the Zhihu dataset has significantly more questions and users than the Toutiao dataset, so the experiments select the Zhihu dataset as the source domain and the Toutiao dataset as the target domain. In the experiments, 80%, 60%, 40%, and 20% of the target domain dataset are extracted to construct the training set, which are noted as TR80, TR60, TR40, and TR20, respectively, and five-fold cross-validation is used for parameter tuning.

Table 4. Comparison between the datasets of Toutiao and Zhihu.

Question Answering Platforms	Number of Questions	Number of Users
Zhihu dataset	116,139	8019
Toutiao dataset	8095	2763

4.2. Evaluation Metrics

In this paper, the following four metrics are used to analyze the recommendation performance of the algorithm proposed in this paper relative to other recommendation algorithms, which include the evaluation of the accuracy of predicted rating and the coverage of recommendation results.

- The *MAE* and *RMSE* metrics are chosen to evaluate the accuracy of rating prediction.

$$MAE = \frac{1}{n}\sum_{i=1}^{\mathrm{n}}|\hat{y}_i - y_i| \tag{17}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{\mathrm{n}}(\hat{y}_i - y_i)^2} \tag{18}$$

where $\hat{y}_i$ and y_i denote the predicted and true ratings of the question answerers, respectively, and n denotes the size of the test set. A smaller value of *MAE* or *RMSE* represents higher accuracy of the recommendation and better recommendation results. In addition, two indicators, Precision and Recall, are selected to evaluate the accuracy of recommendation results.

- *Precision*: A list of experts is recommended for each question in the test set, and Precision represents the ratio of the number of accurate experts to the total number of recommended experts.

$$Precision = \frac{\sum\limits_{q\in Q}|R(q)\cap T(q)|}{\sum\limits_{q\in Q}|R(q)|} \tag{19}$$

where $R(q)$ denotes the N experts recommended for each question q in the test set, $T(q)$ denotes the real expert recommendation list for question q, and Q denotes the set of all questions in the test set.

- *Recall*: The proportion of the number of correctly recommended experts to the number of real experts. The formula is shown below:

$$Recall = \frac{\sum\limits_{q\in Q}|R(q)\cap T(q)|}{\sum\limits_{q\in Q}|T(q)|} \tag{20}$$

The meaning of the signs in the formula is the same as the above formula.

4.3. Comparison Methods

In order to verify the effectiveness of the proposed algorithm, we compare it with the current classical expert recommendation algorithm.

- CQARank [6] is a topic–experience model that jointly learns the topic and expertise by integrating a text content model and link structure analysis;
- GRMC [13] considers expert discovery from the perspective of missing value estimation. Then, a matrix complementation algorithm with graph regularization is used to infer the user model using the user's social network. Two efficient iterative processes, GRMC-EGM and GRMC-AGM, are further developed to solve the optimization problem;
- HSIN [18] not only encodes the question content and social interactions of the questioner to enhance the question embedding performance but also uses a random-walk-based learning method with RNN to match the similarities between new questions and historical questions asked by other users;

- EndCold [20] constructs an undirected heterogeneous graph encoding past questions and answers of users and text information of questions. The model can use the higher-order graph structure and content information to embed nodes in the input graph and then send questions to respondents with expertise.

4.4. Experimental Results and Analysis

4.4.1. Parameter Settings

In order to compare the performance of different methods on evaluation metrics, the parameters of the model in the experiments are selected by cross-validation for optimal values. For the BP network, the number of hidden layers is set to 1, the number of nodes f in the hidden layer is taken in the range of {30, 32, 34, 36, 38}, and the regularization parameter λ is taken in the range of {0.0001, 0.001, 0.01, 0.1, 1}. For the SDAE, the number of nodes in the first hidden layer and the second hidden layer are set to h_1 = {350, 320, 290, 260, 230} and h_2 = {190, 180, 170, 160, 150}. Tables 4 and 5 show the average MAE values for different combinations of parameters when TR60 is chosen as the training set, and the same operation can be performed for a training set with different sparsity levels.

Table 5. Average MAE for different values of f and λ.

λ f	**30**	**32**	**34**	**36**	**38**
0.0001	0.829	0.732	0.729	0.784	0.753
0.001	0.783	0.825	0.804	0.689	0.715
0.01	0.790	0.804	0.843	0.753	0.774
0.1	0.864	0.785	0.783	0.740	0.725
1	0.820	0.814	0.830	0.791	0.784

From Tables 5 and 6, it can be seen that when $f = 0.001, \lambda = 36, h_1 = 160, h_2 = 320$, the parameters take the values as their optimal values, and the MAE obtains the corresponding smallest values.

Table 6. Average MAE for different values of h_1 and h_2.

h_1 h_2	**350**	**320**	**290**	**260**	**230**
190	0.915	0.893	0.864	0.945	1.102
180	0.865	0.845	0.863	0.884	0.965
170	0.858	0.853	0.843	0.864	0.943
160	0.842	0.841	0.852	0.861	0.926
150	0.953	0.997	0.934	0.896	1.026

4.4.2. Performance Comparison

Table 7 shows the results of the algorithms in this paper and the comparison methods on MAE and RMSE. The analysis shows that the algorithm proposed in this paper outperforms all the other four expert recommendation algorithms, and the MAE and RMSE values are at least 0.02 and 0.12 lower on average than the other methods on differently scaled training sets. The comparative results of the above experiments fully demonstrate that the use of auxiliary domain information can alleviate the problem of sparse data within the platform and improve the performance of expert recommendation in CQA.

Figure 5 shows the comparison results of the proposed algorithm and the compared methods in terms of Precision. It can be seen that as the proportion of the training set increases, the knowledge migrates from the source domain to the target domain and becomes richer and more accurate, and the Precision keeps improving. The algorithm in this paper has a higher accuracy than the other four comparison methods. Among the four comparison methods, since CQARank relies on long-term user comments and historical answer data of questions in CQA, in the context of the new CQA platform in this paper, the data of both experts and questions are sparse, so the recommendation effect is the worst; the modeling of users by GRMC depends on the rating matrix and the past CQA data of users, ignoring the question information, which makes it difficult to achieve the ideal recommendation results; HSIN and EndCold are network-representation-based methods compared to CQARank and GRMC, but they only utilize information from a single domain, which is affected by the sparse data in the target domain. It is difficult to obtain enough information during the model training process to achieve the better recommendation effect.

Table 7. Comparison of MAE and RESE results.

Methods	MAE				RMSE			
	TR80	TR60	TR40	TR20	TR80	TR60	TR40	TR20
CQARank	0.852	0.871	0.909	0.985	1.081	1.107	1.233	1.270
GRMC	0.824	0.838	0.862	0.904	1.036	1.082	1.156	1.166
HSIN	0.802	0.821	0.841	0.889	0.952	0.972	1.057	1.062
EndCold	0.788	0.809	0.829	0.879	0.901	0.926	1.029	1.054
FA-CPQAER	0.801	0.807	0.812	0.822	0.829	0.842	0.854	0.912

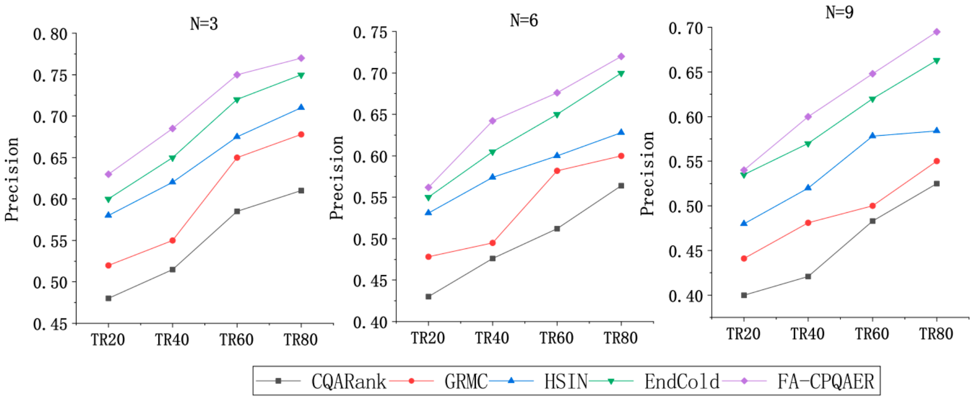

Figure 5. Comparison of the Prediction.

The experimental results of Recall are shown in Figure 6. FA-CPQAER performs better in Recall, and the performance of the other expert recommendation algorithms is ranked as follows: CQARank, GRMC, HSIN, EndCold. HSIN and EndCold, compared to CQARank and GRMC, make full use of the historical question–answer information and expert experience, thus reducing the impact of data sparsity and making the enhancement effect more obvious.

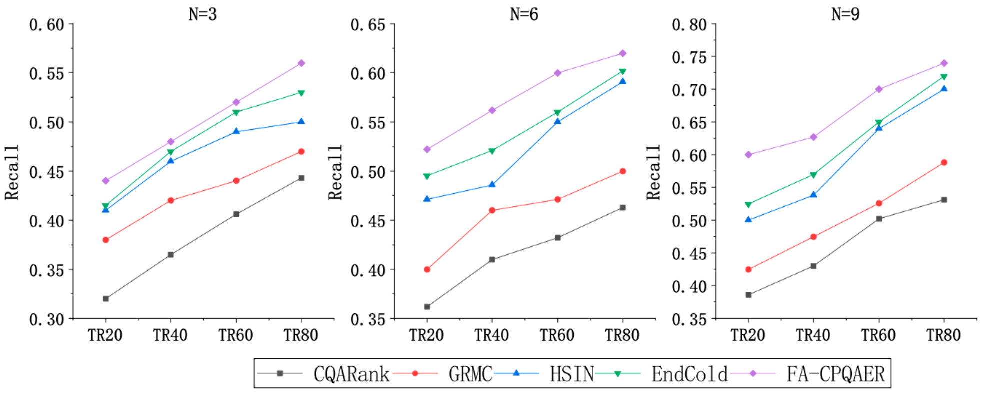

Figure 6. Comparison of the Recall.

4.4.3. Ablation Experiment

To verify whether MMD has an impact on the model performance, we conducted an ablation experiment. The comparison results of the experiments with and without MMD for feature alignment are presented in Figure 7, which shows the performance of the model on the evaluation metrics Precision and Recall for expert recommendation list length $N = 6$ and training set $TR = 40$.

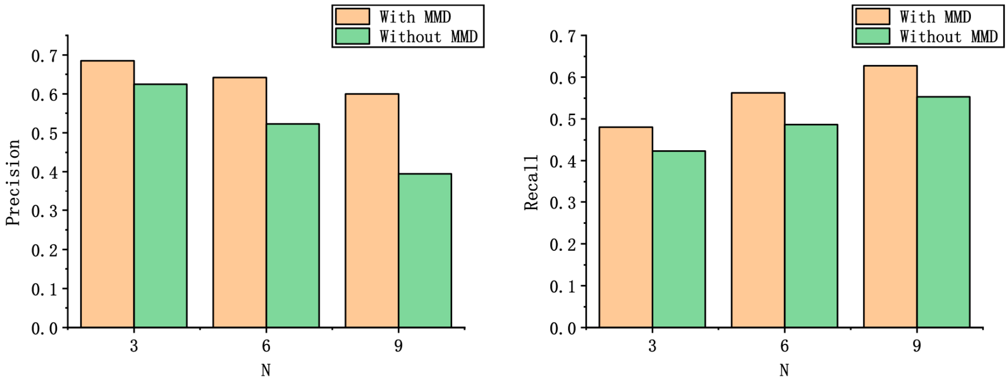

Figure 7. Comparison results with and without MMD.

The ablation experiment shows that using the MMD model improves the evaluation metrics Precision by 24.9% and Recall by 14.2% compared to not using the MMD, which indicates that MMD can achieve the alignment between the data distribution of the two domains through aligning the feature dimensions between the target and auxiliary domains, which has an indispensable role in the overall performance of the model.

5. Conclusions

Transferring the rich historical data from the source platform to the target platform can alleviate the data sparsity problem. However, due to the differences in platform operation mechanisms, the feature composition of questions and experts in different CQA platforms is inconsistent, which makes the feature dimensions and data distributions inconsistent. Thus, data transferring between platforms cannot be performed directly. To solve the above problems, this paper proposes a cross-platform expert recommendation algorithm for CQA, which utilizes the rich information in the source domain to improve the expert recommendation performance in the target domain and solves the problem with inconsistent feature dimensions and data distributions of experts and questions among different platforms by the SDAE. Experiments on two real datasets show that our method can solve the data sparsity problem in a single platform and obtain better recommendation performance.

In this paper, we only perform expert recommendation for new questions in the target platform, and in future work, we will extend this model to achieve matching between new experts and unanswered questions in the target platform. Meanwhile, we consider adding other types of features, such as the social network information of experts to build a more comprehensive feature model that further improves the performance of the recommendation algorithm.

Author Contributions: Conceptualization, B.T. and X.Y.; Investigation and Methodology, X.Y. and X.C.; Experiment and Software, Q.P. and Q.G.; Writing, X.Y. and Q.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number No. 62172249, the Natural Science Foundation of Shandong Province, grant number No. ZR2019MF014, and the Fundamental Research Funds for the Central Universities, JLU, grant number No. 93K172022K01.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets generated and/or analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wang, X.Z.; Huang, C.R.; Yao, L.N.; Beantallah, B.; Dong, M.Q. A survey on expert recommendation in community question answering. *J. Comput. Sci. Technol.* **2018**, *33*, 625–653. [CrossRef]
2. Li, B.; King, I. Routing questions to appropriate answerers in community question answering services. In Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Toronto, ON, Canada, 26–30 October 2010.
3. Du, L.; Buntine, W.; Jin, H. A segmented topic model based on the two-parameter Poisson-Dirichlet process. *Mach. Learn.* **2010**, *81*, 5–19. [CrossRef]
4. Li, B.; King, I.; Lyu, M.R. Question routing in community question answering: Putting category in its place. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, Glasgow, UK, 24–28 October 2011.
5. Xu, F.; Ji, Z.C.; Wang, B. Dual role model for question recommendation in community question answering. In Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval, Portland, OR, USA, 12–16 August 2012.
6. Liu, Y.; Qiu, M.H.; Gottipati, S.; Zhu, F.D.; Jiang, J.; Sun, H.P.; Chen, Z. Cqarank: Jointly model topics and expertise in community question answering. In Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, San Francisco, CA, USA, 27 October–1 November 2013.
7. Sahu, T.P.; Nagwani, N.K.; Verma, S. TagLDA based user persona model to identify topical experts for newly posted questions in community question answering sites. *Int. J. Appl. Eng. Res.* **2016**, *11*, 7072–7078.
8. Zhang, J.; Ackerman, M.S.; Adamic, L. Expertise networks in online communities: Structure and algorithms. In Proceedings of the 16th International Conference on World Wide Web, Banff Alberta, AB, Canada, 8–12 May 2007.
9. Liu, J.; Song, Y.I.; Lin, C.Y. Competition-based user expertise score estimation. In Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, Beijing, China, 24–28 July 2011.
10. Zhu, H.S.; Cao, H.H.; Chen, E.H.; Xiong, H. Towards expert finding by leveraging relevant categories in authority ranking. In Proceedings of the 20th ACM International Conference on Information and Knowledge Management, Glasgow, UK, 24–28 October 2011.
11. Zhu, H.S.; Chen, E.H.; Cao, H.H.; Tian, J.L.; Xiong, H. Ranking user authority with relevant knowledge categories for expert finding. In Proceedings of the 23nd International Conference on World Wide Web, Seoul, Republic of Korea, 7–11 April 2014.
12. Liu, D.R.; Chen, Y.H.; Kao, W.C.; Wang, H.W. Integrating expert profile, reputation and link analysis for expert finding in question-answering websites. *Inf. Process. Manag.* **2013**, *49*, 312–329. [CrossRef]
13. Zhao, Z.; Zhang, L.J.; He, X.F.; Ng, W. Expert finding for question answering via graph regularized matrix completion. *IEEE Trans. Knowl. Data Eng.* **2014**, *27*, 993–1004. [CrossRef]
14. He, T.Z.; Guo, C.L.; Chu, Y.F.; Yang, Y.; Wang, Y.J. Dynamic user modeling for expert recommendation in community question answering. *J. Intell. Fuzzy Syst.* **2020**, *39*, 7281–7292. [CrossRef]
15. Zhao, Z.; Yang, Q.F.; Cai, D.; He, X.F.; Zhuang, Y.T. Expert finding for community-based question answering via ranking metric network learning. In Proceedings of the 25th International Joint Conference on Artificial Intelligence, New York, NY, USA, 9–15 July 2016.
16. Huang, C.; Yao, L.; Wang, X. Expert as a service: Software expert recommendation via knowledge domain embeddings in stack overflow. In Proceedings of the 14th IEEE International Conference on Web Services (ICWS), Honolulu, HI, USA, 25–30 June 2017.
17. Wang, J.; Sun, J.Q.; Lin, H.F.; Dong, H.L.; Zhang, S.W. Convolutional neural networks for expert recommendation in community question answering. *Sci. China Inf. Sci.* **2017**, *60*, 110102. [CrossRef]
18. Chen, Z.Q.; Zhang, C.; Zhao, Z.; Yao, C.W.; Cai, D. Question retrieval for community-based question answering via heterogeneous social influential network. *Neurocomputing* **2018**, *285*, 117–124. [CrossRef]
19. Sun, J.K.; Vishnu, A.; Chakrabarti, A.; Siegel, C.; Parthasarathy, S. Coldroute: Effective routing of cold questions in stack exchange sites. *Neurocomputing* **2018**, *32*, 1339–1367. [CrossRef]
20. Sun, J.K.; Zhao, J.; Sun, H.; Parthasarathy, S. EndCold: An end-to-end framework for cold question routing in community question answering services. In Proceedings of the 29th International Joint Conference on Artificial Intelligence, Yokohama, Japan, 11–17 July 2020.
21. Berkovsky, S.; Kuflik, T.; Ricci, F. Cross-domain mediation in collaborative filtering. In Proceedings of the 11th International Conference on User Modeling, Corfu, Greece, 25–29 July 2007.
22. Singh, A.P.; Gordon, G.J. Relational learning via collective matrix factorization. In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, USA, 24–27 August 2008.

23. Hu, L.; Cao, J.; Xu, J.D.; Cao, L.B.; Gu, Z.P.; Zhu, C. Personalized recommendation via cross-domain triadic factorization. In Proceedings of the 22nd International Conference on World Wide Web, Rio de Janeiro, Brazil, 13–17 May 2013.
24. Li, B.; Yang, Q.; Xue, X.Y. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction. In Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, CA, USA, 11–17 July 2009.
25. Li, B.; Yang, Q.; Xue, X.Y. Transfer learning for collaborative filtering via a rating-matrix generative model. In Proceedings of the 26th Annual International Conference on Machine Learning, Montreal, QC, Canada, 14–18 June 2009.
26. Gao, S.; Luo, H.; Chen, D.; Li, S.T.; Gallinari, P.; Ma, Z.Y.; Guo, J. A cross-domain recommendation model for cyber-physical systems. *IEEE Trans. Emerg. Top. Comput.* **2013**, *1*, 384–393. [CrossRef]
27. Liu, X.Y.; Croft, W.B.; Koll, M. Finding experts in community-based question-answering services. In Proceedings of the 14th ACM International Conference on Information and Knowledge Management, Bremen, Germany, 31 October–5 November 2005.
28. Le, Q.; Mikolov, T. Distributed representations of sentences and documents. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.