Hindawi
International Journal of Intelligent Systems
Volume 2023, Article ID 7434058, 11 pages
https://doi.org/10.1155/2023/7434058

WILEY | Hindawi

Research Article

BamnetTL: Bidirectional Attention Memory Network with Transfer Learning for Question Answering Matching

Lei Su, Jiazhi Guo, Liping Wu, and Han Deng

School of Information Engineering and Automation, Kunming University of Science and Technology, Kunming, Yunnan 650504, China

Correspondence should be addressed to Lei Su; s28341@hotmail.com

Received 8 February 2023; Revised 19 July 2023; Accepted 26 July 2023; Published 3 August 2023

Academic Editor: Vasudevan Rajamohan

In KBQA (knowledge base question answering), questions are processed using NLP (natural language processing), and knowledge base technology is used to generate the corresponding answers. KBQA is one of the most challenging tasks in the field of NLP. Q&A (question and answer) matching is an important part of knowledge base QA (question answering), in which the correct answer is selected from candidate answers. At present, Q&A matching task faces the problem of lacking training data in new fields, which leads to poor performance and low efficiency of the question answering system. The paper puts forward a KBQA Q&A matching model for deep feature transfer based on a bidirectional attention memory network, BamnetTL. It uses biattention to collect information from the knowledge base and question sentences in both directions in order to improve the accuracy of Q&A matching and transfers knowledge from different fields through a deep dynamic adaptation network. BamnetTL improves the accuracy of Q&A matching in the target domain by transferring the knowledge in the source domain with more training resources to the target domain with fewer training resources. The experimental results show that the proposed method is effective.

1. Introduction

In KBQA (knowledge base question answering), Q&A (question and answer) matching plays an important role in question answering systems. It is the key to the success of KBQA. Q&A matching based on a knowledge base refers to finding the question entity associated with the KB (knowledge base) entity as a candidate answer, matching the question and candidate answers to obtain scores, and selecting the answer with the highest score. As shown in Figure 1, if we input the question "What is Nina Dobrev's nationality?", Nina Dobrev is the question entity. We select surrounding entities such as actors, Bulgaria, Canada, and other entities as candidate answers, input them into the KBQA model, and match them with the question to obtain the correct answer, "Canada." Moholkar and Patil proposed the RHAC-ABM [1] (recurrent hybrid ant colony and African buffalo model), which used ACO (ant colony optimization) and ABO (African buffalo optimization) to improve accuracy. They also proposed LA-GWO [2] (Lioness Adapting GWO) to improve the performance of QA. Song et al. proposed CVA [3], which applied a novel channel and spatial attention to object regions. Gao et al. proposed MTA [4] (multitask learning with adaptive attention), which fuses the answer options and question features and then adaptively attends to the visual features. Zhang et al. proposed KAN [5] (knowledge-based augmentation network), which extracted more visual information from images and introduced a knowledge graph to provide the necessary common sense or experience for the reasoning process.

At present, the KBQA method is becoming increasingly mature. Li et al. [6] proposed a multicolumn convolutional neural network based on deep learning, which calculated the similarity of questions and answers from three different aspects: answer path, answer context, and answer type. Hao et al. [7] calculated the weights of entities in candidate answers from answer path, answer context, and answer type using cross attention. Chen et al. [8] proposed the bidirectional attention memory networks (Bamnet), which

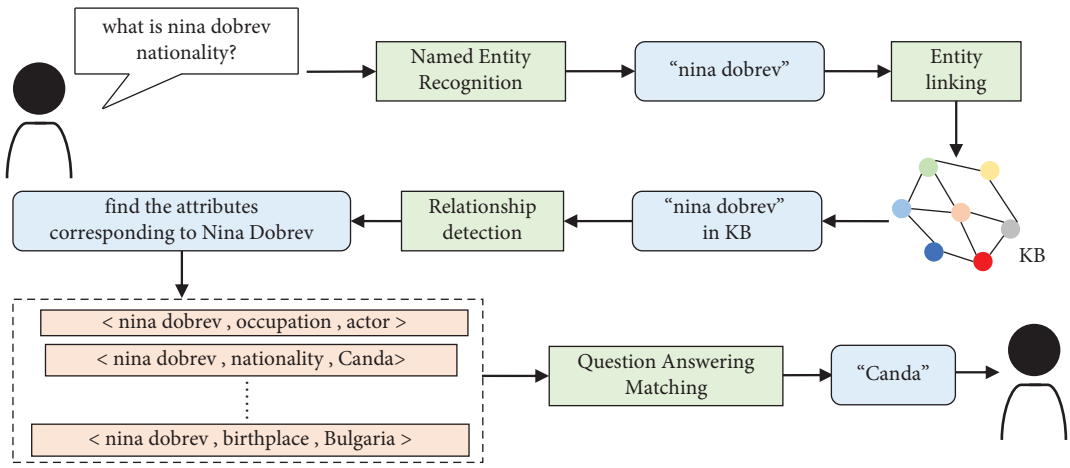


Figure 1: KBQA steps.

catch the correlation between questions and information in the knowledge base through biattention. Deep learning requires a lot of information, but there is less information in the knowledge base in new fields. It is difficult to obtain a better Q&A matching model because of lacking enough training corpus. Therefore, how to utilize a large amount of knowledge of the source domain in the knowledge base to help new domains with a small amount of knowledge to improve the accuracy for Q&A matching is a research difficulty in the KBQA.

The paper proposes KBQA Q&A matching based on deep feature transfer to solve the problem of insufficient accuracy of the trained model due to training data scarcity. Transfer learning uses the similarity between data, tasks, or models to transfer data from the source domain to the target domain to improve the task performance in the target domain. Liu et al. proposed BB-KBQA [9]. BB-KBQA uses the fine-tuning model BERT [10] to obtain a representation of context information learned from a priori knowledge in deep learning. Gu et al. proposed a new version of BERT based on the KBQA model for three-tier generalization [11]. In domain adaptation, Pan et al. proposed TCA [12]. TCA maps the data of the two domains into a high-dimensional reproducing kernel Hilbert space to preserve their internal attributes to the greatest extent possible while minimizing the distance between the source domain and the target domain. Different from TCA, JDA [13] can reduce the differences in the marginal probability distribution and conditional probability distribution between the source domain and target domain at the same time. Wang et al. proposed weighted balanced distribution adaptation (WBDA) [14] on the basis of JDA. WBDA adaptively changes the weight of each category because of the distribution adaptability between domains. The current fine-tuning models have the advantages of good training performance and high accuracy of the training results, but they are still unable to handle distributional differences between the training data and test data. Although the current domain adaptation methods set adaptive weights for domain distribution and categories, migration will be difficult if the data are distributed dispersedly, and there are many domain classes. Therefore, this paper proposes a KBQA Q&A matching model for deep transfer learning based on a biattention memory network: BamnetTL (bidirectional attention memory network for transfer learning). Compared with fine-tuning models, BamnetTL has fewer parameters, needs fewer resources, and is more suitable for most complex environments. Compared with the current domain adaptation methods, BamnetTL uses attention [15] and long short-term memory (LSTM) [16] to make the weight distribution more accurate. It can also achieve transfer in the face of many source domains. The results show that the matching accuracy of BamnetTL is 2.6% higher than that of Bamnet, 3.68% higher than that of the traditional domain adaptation method TCA (transfer component analysis), and 5% higher than that of the traditional domain adaptation method JDA (joint distribution adaptation), which proves that BamnetTL is effective.

Section 1 of the paper introduces the current research progress in KBQA and the significance of this study. Section 2 discusses the basic knowledge relevant to this study. Section 3 introduces the KBQA Q&A matching model based on deep transfer learning. Section 4 presents the results of a comparative experiment and ablation experiment to verify the effectiveness of BamnetTL. Finally, the full text is summarized.

2. Related Works

The methods proposed in this paper are based mainly on deep KBQA, transfer learning, and feature transfer. Next, we introduce relevant concepts and basic knowledge.

2.1. KBQA. Early KBQA methods were based on templates and semantic analysis. In these methods, the Q&A matching part uses manually weighted calculation, keyword matching, and other methods, which are expensive to implement. Cai and Yates [17] proposed a large knowledge base semantic analysis model based on a supervised learning algorithm, pattern matching algorithm, and pattern learning algorithm. Compared with the supervised learning method alone, it shows improved performance. However, it still cannot completely eliminate the dependence on manually annotated data.

Traditional Q&A matching methods have problems such as being time-consuming, having low performance, requiring a large number of manual annotations, and having high costs. To compensate for these shortcomings, researchers have combined traditional Q&A matching methods with CNNs [18], DNNs [19], RNNs [20], and other deep learning methods. Deep learning can automatically extract features, improve efficiency, and reduce costs. Dai et al. proposed CFO [21] (conditional-focused neural question answering with large-scale knowledge bases) for solving simple problems with large knowledge bases. For simple problems, CFO extracts the corresponding entities and relationships from the problems and then uses them to find answers in the knowledge bases. End-to-end deep learning in KBQA means finding the knowledge submap of the question entity and making the correct answer score greater than the wrong answer through training. Shen et al. proposed knowledge-aware attentive bidirectional long short-term memory KABLSTM [22]. KABLSTM uses input background knowledge to enrich representation learning in question answering systems. It uses CNN and an attention model guided by context to embed background knowledge into a sentence representation and uses knowledge-aware attention to correlate question–answer pairs. Deep learning requires a considerable amount of data, but there is less relevant knowledge in the knowledge bases of new fields. Therefore, it is difficult to obtain a better Q&A matching model because of the lack of training knowledge. Therefore, using source fields with large amounts of knowledge in their knowledge bases to help match questions and answers in new fields with small amounts of knowledge to improve the model accuracy is a research difficulty in KBQA.

2.2. Feature Transfer. Feature transfer transforms features after extraction from knowledge to reduce the distance between the source domain and target domain or transforms the features of the source domain and target domain into a feature space for recognition. Blitzer proposed the SCL (structural corresponding learning) method [23] based on structure correspondence. It transforms unique features in one space into axis features in other spaces. Long et al. proposed combining instance and feature transfer, that is, adding transfer joint matching (TJM) [24] to minimize the distribution distance. Zhang et al. proposed having the source domain and target domain each train different transformation matrices [25].

2.3. Deep Transfer Learning. Deep transfer learning directly learns from the original data, automatically extracts more expressive features, and meets the end-to-end learning requirements in practical applications. When transforming features, the main strategy of deep transfer learning is to select some layers of the deep learning network to add domain adaptation methods to enhance the learning level and generalization ability of the network. The loss in deep transfer learning is calculated as follows:

$$\ell = \ell c\left(D_t, y_t\right) + \lambda \ell A\left(D_s, D_t\right), \tag{1}$$

where ℓ is the final loss of the network, $\ell c(D_s, y_s)$ is the general classification loss of the network, $\ell A(D_s, D_t)$ is the adaptive loss of the network, D_t denotes samples from the target domain, D_s denotes samples from the source domain, y_t denotes labels from the target domain, and λ is the weight parameter for weighting the two parts. Yosinski et al. [26] proposed that the learning effect of the network increasingly depends on downstream tasks with the deepening of the network layers. The shallow layers are only a general feature of learning. In different tasks, the features of the shallow layers are universal; that is, the optimal selection adds domain adaptation to the deep layers. For example, Tzeng et al. proposed DDC [27] (deep domain confusion). DDC adds the MMD (maximum mean discrepancy) distance to the penultimate layer of AlexNet to reduce the distance between the source domain and the target domain. Long and Wang proposed DAN [28] (deep adaptation network). Different from DDC, DAN can adapt to multilayer networks and uses MMD, which assigns different weights to Gaussian distributions and linear distributions when calculating mean differences. JAN [29] (joint adaptation network) makes further efforts to add the joint probability distributions of features and labels.

3. Methods

Due to the strong feature extraction ability and generalization ability of deep feature transfer, the paper proposes a KBQA Q&A matching model for deep feature transfer based on a biattention memory network: BamnetTL (bidirectional attention memory network for transfer learning). The BamnetTL model consists of four components, which are the FSL (feature selection layer), FEL (feature enhancement layer), Q&AML (Q&A matching layer), and FTL (feature transfer layer), as shown in Figure 2.

3.1. Feature Selection Layer (FSL). First, the question and candidate answers are words embedded in the FSL. The question is divided into question sentences and question words. The question words $QV = \{qv_i\}_{i=1}^{|QV|}$ are embedded by a dictionary (each word corresponds to a number) to obtain the question word embedded representation V^Q. Then, the question $Q = \{q_i\}_{i=1}^{|Q|}$ is encoded through LSTM to obtain H^Q, and question features and candidate answer features are further selected through self-attention:

$$att^Q = \text{self} - \text{attention}\left(H^Q\right). \tag{2}$$

All entities in the knowledge base can be candidate answers. However, because the number of entities in the knowledge base is very large, the computational cost is enormous. Thus, the entity that is associated with the entity in the question is selected. Embedding candidate answers to obtain $A = \{A_i\}_{i=1}^{|A|}$ and using KV-MemNNs [30] (key-value memory networks) save three attributes of the candidate answers: the answer type $HA_i^{k_t}$, answer path $HA_i^{k_p}$, and

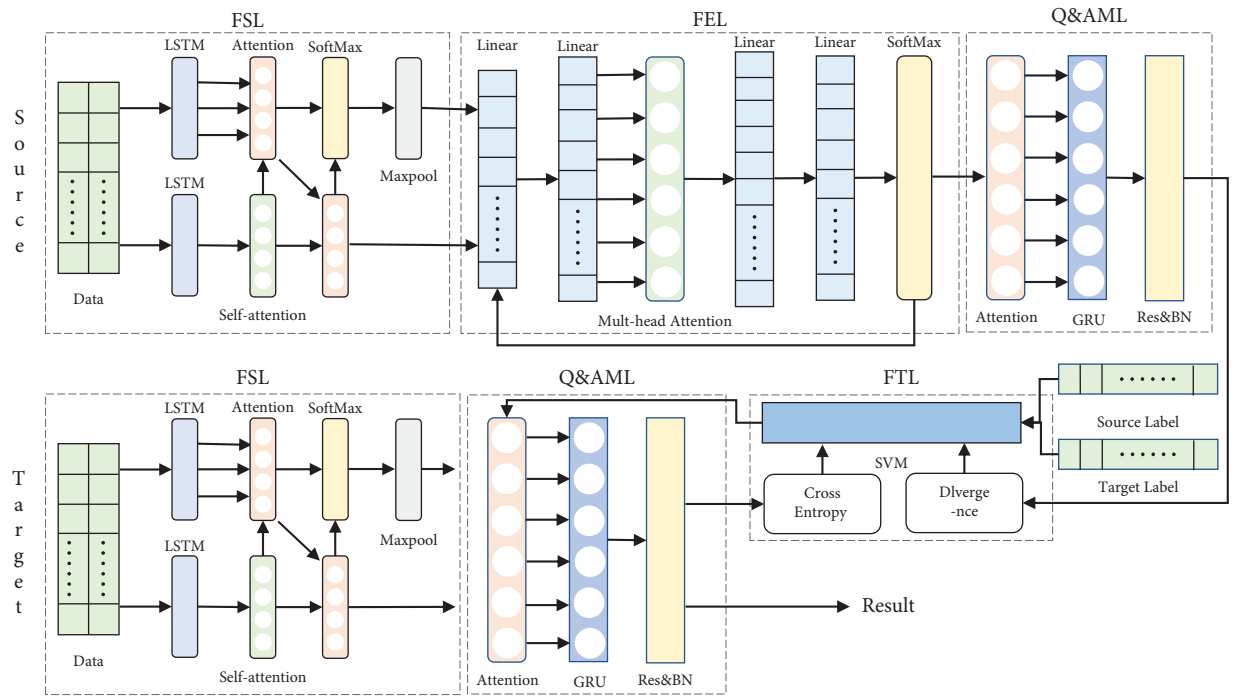


Figure 2: Overall architecture of the BamnetTL model.

answer context $HA_i^{k_c}$. k_t is the type of candidate answer that is derived from the knowledge base using knowledge-aware attention. k_p is the path of candidate answers that are derived from the knowledge base using knowledge-aware attention. k_n is the context of candidate answers that are derived from the knowledge base using knowledge-aware attention. The question word corresponds to the same type of answer, which plays an important role in sorting the candidate's answers. For example, the question "when" always corresponds to an answer of type "time"; the answer to the question "when did the New York Knicks win a championship?" is "the 1970 NBA Finals." The answer path is a collection of relationships from topic entities to candidate answers. The answer context is the topic entities around the candidate's answers. LSTM can overcome the influence of short-term memory and can choose to save or forget information. They are encoded using LSTM and linear projection:

$$\begin{aligned} M_i^{k_t} &= \int_t^k \left(HA_i^{k_t}\right), \\ M_i^{v_t} &= \int_t^v \left(HA_i^{k_t}\right), \end{aligned} \tag{3}$$

where $M_i^{k_t}$ and $M_i^{v_t}$ are the keys and values, respectively, of the answer type $HA_i^{k_t}$. Similarly, the answer path and answer context also have keys and values $M_i^{k_p}$, $M_i^{v_p}$ and $M_i^{k_c}$, $M_i^{v_c}$. In a KV-MemNNs named M, each line is $M_i = \left\{M_i^k, M_i^v\right\}$. $M_i^{\mathrm{k}} = [M_i^{k_t}, M_i^{k_p}, M_i^{k_c}]$ denotes the keys in M, and $M_i^{\mathrm{v}} = [M_i^{v_t}, M_i^{v_p}, M_i^{v_c}]$ denotes the values of the keys in M.

As shown in Figure 3, a feature is input into the attention to capture the two-way interaction question feature and the answer feature. The weight for each answer to the question and the weight assigned to each question for the answer are obtained. Then, Maxpool and Softmax are used to remove redundant information and compress the features:

$$\begin{aligned} M^k &= \left\{M_i^{k_t}, M_i^{k_p}, M_i^{k_n}\right\}_1^{|A|}, \\ Att_k^a &= \mathrm{attention}\left(att^Q, M_i^k\right), \\ \overline{Att_k^a} &= \mathrm{Softmax}\left(Att_k^a\right), \\ \overline{att^Q} &= \mathrm{MaxPool}\left(\overline{Att_k^a} att^Q M_i^v\right), \\ Att^Q &= \mathrm{MaxPool}\left(\overline{att^Q}\right). \end{aligned} \tag{4}$$

3.2. Feature Enhancement Layer (FEL). The feature enhancement layer (FEL) consists of multihead attention [15] and linear [31] layers for selecting high-dimensional features from the features' output by the FSL. The FEL can enhance the features from the previous step to obtain more accurate and effective information. First, each input feature is transformed linearly for uniform integration:

$$A_1, Att_1^Q, V_1^Q, M_1^k, M_1^v, = x\left(A, Att^Q, V^Q, M^k, M^{kv}\right) + b, \tag{5}$$

where x and b are the parameters of linear. The calculated feature is input into multihead attention. In contrast to attention, multihead attention can focus on different

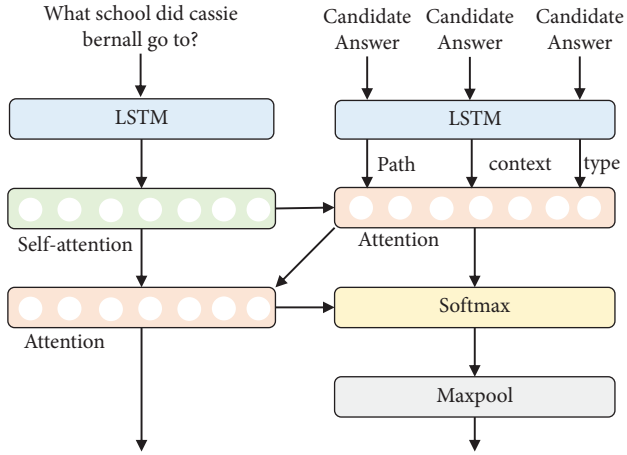


Figure 3: Architecture of the FSL.

information from subspaces in different parts. Each head of multihead attention focuses on only one subspace in the final output sequence, independent of the others. The core strategy is to use multihead attention to make each attention part optimize the different feature parts of each word and select richer feature information. Therefore, the features obtained by the FSL are input into the model.

$$Z = \text{softmax}\left(\frac{QK^t}{\sqrt{d_k}}\right)V. \tag{6}$$

As shown in Figure 4, the feature information is input into the multihead attention, and the output is transformed through two linear layers. Then, a ReLU [32] function is used to obtain the selected features:

$$\tilde{z} = x(\max\ \ (0, xz + b)) + b, \tag{7}$$

where x and b are the parameters of linear. z is the information feature of the previous layer. After an appropriate number of rounds, we obtain enhanced features: $\widetilde{A}$, $\widetilde{Att^Q}$, $\widetilde{V^Q}$, $\widetilde{M^k}$, and $\widetilde{M^v}$.

3.3. Question and Answer Matching Layer (Q&AML). Q&A matching matches each question with the most suitable answer. The question and answer matching layer Q&AML can calculate according to the features of questions and answers and select the candidate answer with the shortest distance from the question feature vector as the answer. As shown in Figure 5, the attention of Q&AML can select the answer through other factors, and a GRU [33] (gated recurrent unity) can analyse sequence data. The Q&AML structure is shown in Figure 4. First, Maxpool is used to reduce the feature dimensions of the problem representation $\widetilde{Att^Q}$, the keys of the candidate answers $\widetilde{M^k}$, and the values of the candidate answers $\widetilde{M^v}$.

$$\begin{aligned} Att^{Q'}, M^{v'}, M^{k'} &= \text{MaxPool}\left(\widetilde{Att^Q}, \widetilde{M^v}, \widetilde{M^k}\right), \\ \text{Answer1} &= \text{Attention}\left(Att^{q'}, M^{k'}, M^{v'}\right), \\ \text{Answer2} &= \text{GRU}(\text{Answer1}), \\ \text{Answer3} &= \text{RES\&BN}(\text{Answer2}), \end{aligned} \tag{8}$$

where $Att^{Q'}$ is input into the Q&AML, and attention is used to obtain the value Answer1. Then, sequence information is extracted to update the vector through GRU to obtain Answer2. RES [34] and BN [35] are used to make the updating process converge to obtain Answer3. The loss is calculated using $Att^{Q'}$ for Answer1, Answer2, and Answer3 separately.

3.4. Feature Transfer Layer (FTL). Wang Jindong proposed the dynamic distribution adaptation network (DDAN) [36] on the basis of dynamic distribution adaptation (DDA) [36] DDAN uses the backbone network to learn useful features, and DDA to perform domain adaptation. As shown in Figure 6, the feature transfer layer (FTL) adapts the features of NLP to reduce the distance between the source domain and the target domain. $\Omega \in R^d$ is the input space with dimension d. $\Omega_s = \{\text{Answer}^s, y^s\}$ is the source domain, where Answer^s denotes the candidate answers, y^s denotes their labels, and Ω_t is the target domain. The purpose of transfer learning is to make the source domain closer to the target domain in space, which can be expressed as follows:

$$f = \min_{\Theta} \sum_{i=1}^{n} J\left(f\left(\text{Answer}_i^s\right), y_i^s\right) + \lambda \overline{D_f}\left(\Omega_s, \Omega_t\right) + \rho R_f\left(\Omega_s, \Omega_t\right), \tag{9}$$

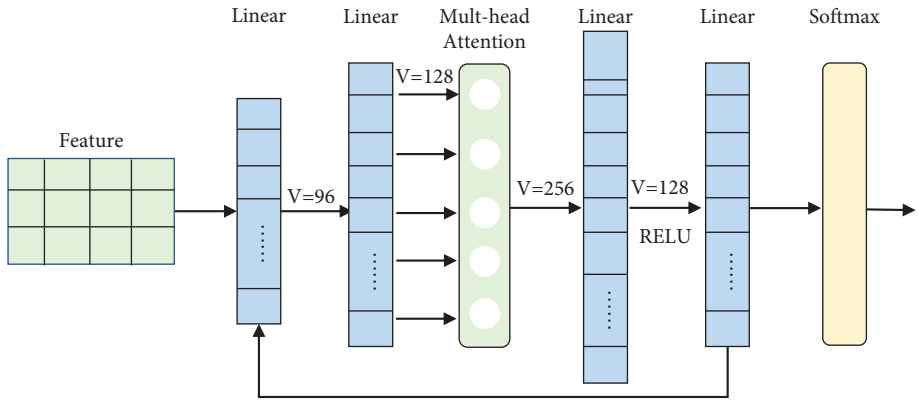


Figure 4: Architecture of the FEL.

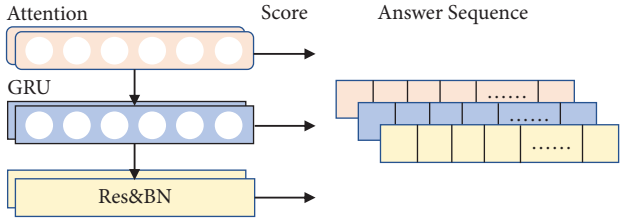


Figure 5: Architecture of the Q&AML.

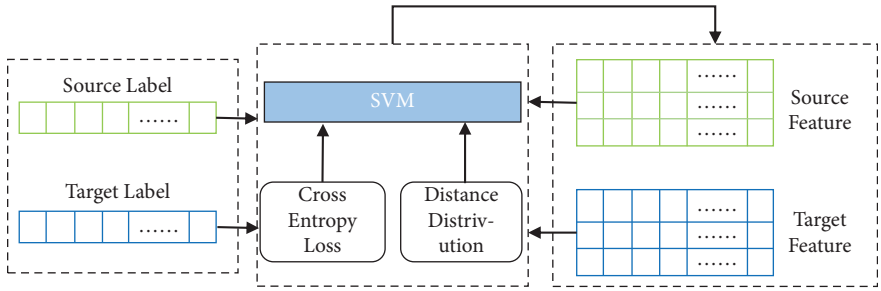


Figure 6: Architecture of the FTL.

where $J(\cdot,\cdot)$ is the cross-entropy loss function, $\Theta = \{w, b\}$ includes the weights and bias parameters of the neural network, λ and ρ are the trade-off parameters, $D_f(\Omega_s, \Omega_t)$ represents the distributional divergence between the source domain and target domain, and R_f denotes a regularization function. We do not fit the data of the whole source domain with the data of the target domain. Instead, the computational cost is reduced by comparing each training batch.

4. Experiments

4.1. Datasets. The dataset used in this paper is a Q&A matching dataset, WebQuestionSP [37], which is based on the knowledge base "Freebase" [38]. WebquestionSP includes question IDs, question contents, and best answers. We extract one-hop entities (the attributes of entities for query in a knowledge base are called one-hop entities) from Freebase as candidate answers. The dataset is divided into five domains: location, people, sports, film, and base. Their quantities are shown in Table 1.

We experiment with different domain data on different models. For example, we use the location domain as the target domain and No_Location as the source location for KBQA. No_Location contains a people domain, sports domain, etc. The base domain includes all domains except the Location_People domain.

4.2. Experimental Setup. We embed the datasets of the source domain and target domain through the dictionary and find the corresponding entity of each question in Freebase. We take its one-hop entities as candidate answers. We set the number of candidate answers to 96 to include the maximum number of candidate answers. Since each question has an uncertain number of correct answers, we set the number of correct answers for each question to 13, which can include the maximum number of correct answers.

Table 1: Experimental datasets.

Target domain	Train	Valid	Test
Location	758	188	519
People	762	190	540
Sports	220	69	139
Film	183	40	122
Base	1503	377	973

All the models in the experiment are composed of neural networks, and the method is implemented in Python 3.6 and Python 1.12.1. The classification loss function multilabel hinge loss or MultLabelMarginLoss is applicable to the case where a sample corresponds to multiple labels:

$$\text{loss}(x, y) = \sum_{ij} \frac{\max\ (0, 1 - x[y[i]] - x[i])}{x.\text{size}(0)}, \tag{10}$$

where x is the predicted value and y is the label of the data. We add the MultLabelMarginLoss and transfer loss function to obtain the final loss:

$$\text{loss} = \text{loss}(x, y) + f. \tag{11}$$

The paper applies the Adam [39] optimizer to BamnetTL. The Adam optimizer has an adaptive learning rate mechanism, so it can allocate different learning rates to different parameters, which can increase the speed of optimization.

4.3. Evaluation Measure. The paper uses the evaluation metric F1 to evaluate the performance of the model, which is widely used in Q&A. F1 is composed of precision and recall. It is too one-sided to only consider the precision or recall as an evaluation indicator. F1 can be compatible with precision and recall:

$$\begin{aligned} F1 &= \frac{2 * (\text{precision} * \text{recall})}{\text{precision} + \text{recall}} \\ &= \frac{2 * \text{TP}}{2 * \text{TP} + \text{FN} + \text{FP}}, \end{aligned} \tag{12}$$

where TP (true positive) represents the number of correct predictions, FP (false positive) represents the number of instances belonging to other categories that are predicted to belong to this category, and FN (false negative) represents the number of instances belonging to this category that are predicted to belong to other categories.

4.4. Experimental Result

4.4.1. Q&A Matching for Deep Transfer Learning. To evaluate whether the BamnetTL model achieves satisfactory results, we carry out the following experiments. We test different models on five different datasets. We set the model hyperparameters to the same values. We set the number of iterations to 100 and the initial learning rate to 0.01.

As shown in Table 2, Bamnet (bidirectional attentive memory network) ranks first among all Q&A matching models, so we select Bamnet as the baseline. Bamnet + FEL represents the baseline Bamnet plus the FTL. Compared with Bamnet, BamnetTL shows performance improvements of 2.6%, 2.35%, 1.23%, 0.58%, and 1.17% in the five areas of location, people, sports, film, and base, respectively. This proves the superiority of BamnetTL in KBQA. Compared with Bamnet, Bamnet + FEL shows performance improvements of 2.03%, 0.78%, 0.45%, and 0.58% in the four areas of location, people, sports, and base, respectively, and shows reduced performance in only the film domain. This proves that the FEL is effective.

The performance of BamnetTL in the people and location domains is better than that in the sports and film domains because the people and location domains have more data than the other domains. However, BamnetTL has the worst performance in the base domain. The data distribution in the base domain shows that this domain is not a single domain but a combination of many domains. Therefore, we draw the following conclusion: the simpler the domain is, the more data there are, and the better the transfer effect is.

4.4.2. Comparison with Traditional Transfer Learning Models. To prove that the transfer effect of BamnetTL is better than that of traditional transfer learning models, we test different transfer learning models on Q&A matching. Under the same experimental conditions, we compare the experimental results of the traditional transfer model with those of BamnetTL.

As shown in Table 3, BamnetTL is more efficient than TCA and JDA. Compared with TCA, BamnetTL shows an average F1 increase of 5.17%. Compared with JDA, BamnetTL shows an average F1 increase of 3.854%. Compared with only traditional transfer learning, we find that using FEL with traditional transfer learning significantly improves the transfer performance. Therefore, we draw the following conclusion: the feedback from FEL improves the transfer effect, and BamnetTL is superior to traditional transfer learning.

Figure 7 shows the F1 values obtained by traditional transfer learning, traditional deep transfer learning, and BamnetTL on five datasets through 100 iterations. BamnetTL is more stable and effective than the other models in all Q&A matching experiments.

4.4.3. FEL Impact on the Model. The function of the FEL layer is to enhance the text features and extract deep text features from them. Too few FEL layers will result in inadequate extraction of information, which will have

Table 2: F1 values on various datasets for different models (%).

Model	Location	People	Sports	Film	Base
Bamnet	60.55	60.74	52.84	53.51	46.63
Bamnet + FEL	62.58	61.52	53.29	52.19	47.21
BamnetTL	63.15	63.09	54.07	54.09	47.8

Table 3: F1 values on various datasets for traditional transfer learning models and BamnetTL (%).

Model	Location	People	Sports	Film	Base
Bamnet + TCA	58.35	57.62	51.54	47.63	43.46
Bamnet + JDA	56.94	54.39	52.33	49.55	44.26
Bamnet + FEL + TCA	59.47	56.45	48.13	49.72	45.61
Bamnet + FEL + JDA	57.14	57.48	53.37	52.14	42.80
BamnetTL	63.15	63.09	54.07	54.09	47.8

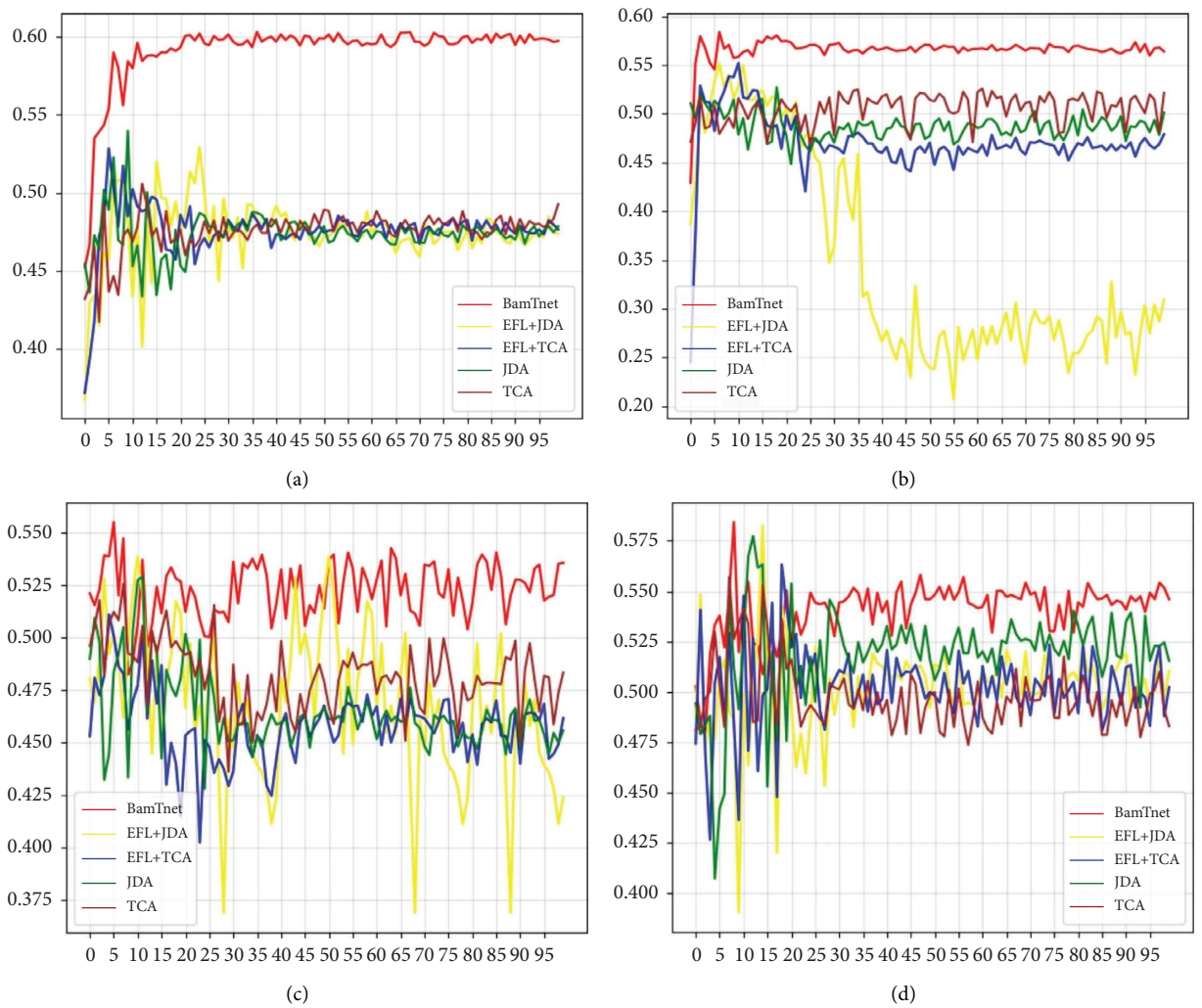
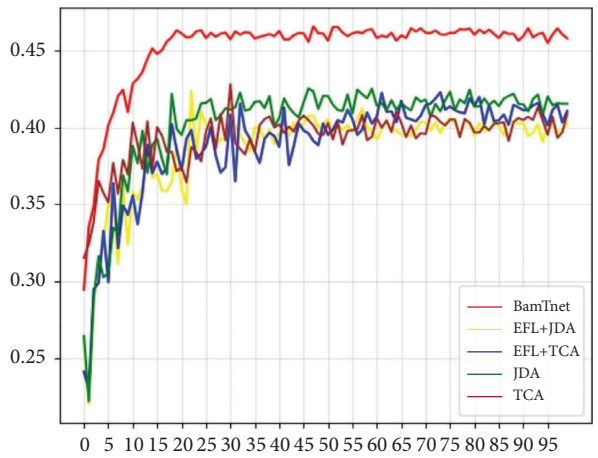


Figure 7: Continued.



(e)

Figure 7: Comparing the effects of different transfer learning models on different data: (a) people, (b) location, (c) sports, (d) film, and (e) base.

Table 4: Influence of the number of FEL layers on the performance of BamnetTL (%).

Layers	Location	People	Sports	Film	Base
1	59.61	62.34	53.05	51.44	46.91
2	61.37	61.34	53.53	50.75	45.64
3	63.15	63.09	54.07	54.09	47.8
4	60.13	61.66	52.92	51.58	47.10
5	60.37	60.28	50.61	54.01	46.15
6	59.87	61.41	51.83	51.87	46.02

a negative impact on downstream tasks. Too many FEL layers will cause parameter redundancy, which will greatly reduce the model performance. Therefore, to find the appropriate number of FEL layers, we perform the following experiment. We test the effects of different numbers of FEL layers on BamnetTL to find the best number of FEL layers. We follow the control variates method. Except for the number of layers, no parameters are changed.

Table 4 shows that when the number of FEL layers is 3, the performance is better than that with other numbers of layers. By comparing 1, 2, and 3 FEL layers, we find that F1 is positively correlated with the number of FEL layers and reaches the best value when the number of layers is 3. We compare 3, 4, 5, and 6 FEL layers and find that F1 is negatively correlated with the number of FEL layers. We draw the following conclusion: when the number of FEL layers is 3, the BamnetTL model achieves the best performance. When the number of FEL layers is less than 3, the informative features of the model are not sufficiently extracted. When the number of FEL layers is higher than 3, the redundancy of parameters in the model results in lower model performance.

4.4.4. Impact of Multihead Attention on the Model. Multihead attention can help the model to expand its ability to focus on different locations. In NLP, it can be used to locate "tokens" or features. We use it in the FEL layer to increase the spatial feature selection ability. The number of heads in the multihead attention will affect the selection of spatial features, and too few heads will lead to insufficient feature selection, whereas too many heads will lead to parameter redundancy and reduce the performance of the model. Therefore, we test the impacts of different numbers of multihead attention heads on BamnetTL. We follow the control variates method. Except for the number of multihead attention heads, no parameters are changed.

As shown in Table 5, when the number of heads is 16, the feature selection effect of the multihead attention is the highest, and the impact on the model is the best. Too many heads will lead to weight dispersion, and too few heads will focus too much on their own positions, which will reduce the enhancement effect of FEL or even have a negative effect, leading to a reduction in the transfer performance of BamnetTL.

4.4.5. Impact of Different Networks on the Model. LSTM, GRU, and attention all can undertake certain tasks. However, different networks will affect the overall performance of the model. Therefore, we have done the following experiments to prove the contribution of each network to the model. All experimental environments are the same.

As shown in Table 6, when the LSTM in the FSL layers is replaced by GRU, the performance of BamnetTL will decline, which proves the effectiveness of LSTM. In the FEL layers, we removed multihead attention to prove its effectiveness. In the Q&AML layers, we removed GRU and attention modules, respectively, which also proved that GRU and attention contributed positively to the BamnetTL.

Table 5: Influence of the number of heads on the performance of BamnetTL (%).

Layers	Location	People	Sports	Film	Base
8	62.23	59.92	52.76	49.83	45.14
16	63.15	63.09	54.07	54.09	47.8
32	61.42	60.96	51.99	52.01	45.63
64	61.00	61.55	51.90	51.39	45.16

Table 6: Influence of different networks on the performance of BamnetTL (%).

Layers	Model	Location	People	Sports	Film	Base
FSL	LSTM->GRU	60.93	60.98	53.05	48.22	45.87
FEL	No_multihead attention	62.43	62	53.91	52.15	46.16
Q&AML	No_GRU	61.58	60.44	53.33	50.37	45.99
	No_Attention	60.28	60.33	51.7	52.76	45.5
BamnetTL	Ours	63.15	63.09	54.07	54.09	47.8

5. Conclusions and Future Works

Nowadays, with the rapid development of science and technology, many new fields have emerged. In the new field with little knowledge, there are few models that can undertake the search task of users. This paper proposes a KBQA Q&A matching model for deep feature transfer based on a biattention memory network, BamnetTL, to solve the problem that the existing transfer learning model has difficulty achieving the desired effect when facing new domains with less knowledge. In this paper, we use biattention for end-to-end learning and feature transfer to shorten the distances between multiple source domains and target domains. Compared with different transfer learning models and other deep learning models, BamnetTL shows powerful generalization ability and improved performance in KBQA. BamnetTL has a high performance in the face of new fields, which solves the problem of insufficient effect in the face of new fields. Future work will focus on continuing to shorten the domain distances and further research on feature enhancement.

Data Availability

The data used to support the findings of this study are included within the article [31]. The dataset "WebQuestionsSP" can be derived from Microsoft Download Center (https://www.microsoft.com/en-us/download/details.aspx?id=52763).

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The work was supported by the National Science Foundation of China under Grant no. 62166021.

References

[1] K. Moholkar and S. Patil, "A Hybrid optimized deep learning framework to enhance question answering system," *Neural Processing Letters*, vol. 54, no. 6, pp. 4711–4734, Article ID 21022, 2022.

[2] K. Moholkar and S. Patil, "Lioness adapted GWO-based deep belief network enabled with multiple features for a novel question answering system," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 30, no. 01, pp. 93–114, 2022.

[3] J. Song, P. Zeng, L. Gao, and H. T. Shen, "From pixels to objects: cubic visual attention for visual question answering," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pp. 906–912, IJCAI, Stockholm, Sweden, July 2018.

[4] L. Gao, P. Zeng, J. Song, X. Liu, and H. T. Shen, "Examine before You Answer: Multi-Task Learning with Adaptive-Attentions for Multiple-Choice VQA," in *Proceedings of the 26th ACM international conference on Multimedia*, Seoul, Republic of Korea, October 2018.

[5] L. Zhang, S. Liu, D. Liu et al., "Rich visual knowledge-based augmentation network for visual question answering," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 10, pp. 4362–4373, 2021.

[6] D. Li, F. Wei, Z. Ming, and X. Ke, "Question answering over freebase with multi-column convolutional neural networks," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, pp. 260–269, Beijing, China, July 2015.

[7] Y. Hao, Y. Zhang, L. Kang, S. He, and J. Zhao, "An end-to-end model for question answering over knowledge base with cross-attention combining global knowledge," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pp. 221–231, Vancouver, Canada, July 2017.

[8] Y. Chen, L. Wu, and M. J. Zaki, "Bidirectional attentive memory networks for question answering over knowledge bases," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2913–2923, Minneapolis, Minnesota, June 2019.

[9] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, pp. 4171–4186, Minneapolis, Minnesota, June 2018.

[10] A. Liu, Z. Huang, H. Lu, X. Wang, C. Yuan, and B. B. Kbqa, "BERT-Based Knowledge Base Question Answering," in *Proceedings of the Chinese Computational Linguistics: 18th China National Conference*, pp. 81–92, Cham, Switzerland, October 2019.

[11] Y. Gu, S. Kase, M. Vanni et al., "Beyond IID: three levels of generalization for question answering on knowledge bases," in *Proceedings of the Web Conference 2021*, pp. 3477–3488, Ljubljana, Slovenia, June 2021.

[12] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Transactions on Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.

[13] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer feature learning with joint distribution adaptation," in *Proceedings of the 2013 IEEE International Conference on Computer Vision*, pp. 2200–2207, Sydney, NSW, Australia, December 2013.

[14] J. Wang, Y. Chen, S. Hao, W. Feng, and Z. Shen, "Balanced Distribution Adaptation for Transfer Learning," in *Proceedings of the 2017 IEEE international conference on data mining (ICDM)*, pp. 1129–1134, New Orleans, LA, USA, November 2017.

[15] A. Vaswani, N. Shazeer, N. Parmar et al., "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] Q. Cai and A. Yates, "Large-scale semantic parsing via schema matching and lexicon extension," in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pp. 423–433, Sofia, Bulgaria, August 2013.

[18] J. Bouvrie, *Notes on Convolutional Neural Networks*, Center for Biological and Computational Learning, Massachusetts, MA, USA, 2006.

[19] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[20] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[21] Z. Dai, L. Li, and W. Xu, "CFO: conditional focused neural question answering with large-scale knowledge bases," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, vol. 1, pp. 800–810, Berlin, Germany, August 2016.

[22] Y. Shen, Y. Deng, M. Yang et al., "Knowledge-aware attentive neural network for ranking question answer pairs," in *Processing of the 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 901–904, Ann Arbor, MI, USA, June 2018.

[23] J. Blitzer, R. McDonald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pp. 120–128, Sydney, Australia, July 2006.

[24] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, "Transfer joint matching for unsupervised domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1410–1417, Columbus, OH, USA, June 2014.

[25] J. Zhang, W. Li, and P. Ogunbona, "Joint geometrical and statistical alignment for visual domain adaptation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1859–1867, CVPR2017: Honolulu, HI, USA, July 2017.

[26] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" *Advances in Neural Information Processing Systems*, vol. 27, pp. 3320–3328, 2014.

[27] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep Domain Confusion: Maximizing for Domain Invariance," 2014, https://arxiv.org/abs/1412.3474.

[28] M. Long and J. Wang, "Learning Transferable Features with Deep Adaptation Networks," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, pp. 97–105, JMLR, Lille, France, July 2015.

[29] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep Transfer Learning with Joint Adaptation Networks," in *Proceedings of the 34th International Conference on Machine Learning*, pp. 2208–2217, PMLR, Sydney, NSW, Australia, August 2017.

[30] A. Miller, A. Fisch, J. Dodge, A. H. Karimi, A. Bordes, and J. Weston, "Key-value memory networks for directly reading documents," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 1400–1409, Austin, Texas, November 2016.

[31] B. W. White and F. Rosenblatt, "Principles of neurodynamics: perceptrons and the theory of brain mechanisms," *American Journal of Psychology*, vol. 76, no. 4, p. 705, 1963.

[32] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," *Journal of Machine Learning Research*, vol. 15, pp. 315–323, 2011.

[33] K. Cho, B. Van Merriënboer, C. Gulcehre et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proceedings of the2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724–1734, EMNLP, Doha, Qatar, October 2014.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, CVPR2016: Las Vegas, NV, USA, June 2016.

[35] S. Ioffe and C. Szegedy, "Batch normalization: accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, pp. 448–456, ICML, Lille, France, July 2015.

[36] J. Wang, Y. Chen, W. Feng, H. Yu, M. Huang, and Q. Yang, "Transfer learning with dynamic distribution adaptation," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 11, no. 1, pp. 1–25, 2020.

[37] T. Y. Wen, M. Richardson, C. Meek, M. W. Chang, and J. Suh, "The value of semantic parse labeling for knowledge base question answering," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pp. 201–206, ACL, Berlin, Germany, August 2016.

[38] K. D. Bollacker, C. Evans, P. Paritoch, T. Sturge, and J. Taylor, "Freebase: a collaboratively created graph database for structuring human knowledge," in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pp. 247–1250, Vancouver, Canada, June 2008.

[39] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference for Learning Representations*, San Diego, CA, USA, January 2015.