

Received September 12, 2020, accepted October 8, 2020, date of publication October 30, 2020, date of current version September 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2020.3034920

Answer Category-Aware Answer Selection for Question Answering

WEIJING WU¹, YANG DENG^{1,2}, YUZHILANG³, AND KAI LEI¹, (Member, IEEE)

¹ICNLAB, School of Electronics and Computer Engineering (SECE), Peking University, Shenzhen 518055, China

²Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong

³School of Electronics and Computer Engineering (SECE), Peking University Shenzhen Graduate School, Shenzhen 518055, China

Corresponding author: Yuzhi Liang (liangyz@pkusz.edu.cn)

This work was partially supported by the Natural Science Foundation of Guangdong Province (No. 2021A1515011602, 2019A1515111203), Shenzhen Innovation and Entrepreneurship Plan (No. JSGG20191129110603831), and Natural Science Foundation of China (No. 62072012).

ABSTRACT As a key problem in artificial intelligence, question answering (QA) has always been a topic of intensive research. Most existing methods cast question answering as an answer selection task. The size of the candidate answer pool is usually very large, so it is difficult to accurately select the correct answer. One of the solutions is to narrow the range of candidate answer pool based on the category labels of the answers. However, QA tasks in reality usually only provide the category label of the question but not the category label of the answer. Based on this observation, we propose an Answer Category-Aware Answer Selection system (ACAAS), which jointly leverage unlabelled answer data and labelled question category data to generate answer category pseudo-labels in a joint embedding space. Experimental results on two public QA datasets demonstrate the effectiveness of the proposed method.

INDEX TERMS Answer selection, label transfer, question answering.

I. INTRODUCTION

Question answering refers to making the machine understand natural language questions and deliver answers automatically [1]. As one of the core tasks of artificial intelligence, question answering technology has received more and more attention due to its wide application in natural language processing and information retrieval. There are many existing machine learning-based question answering methods. Most question answering systems convert question answering tasks into answer selection tasks, such as [2], [3]. That is, given a question and a set of candidate answers, the system automatically selects the correct answer from the pool of candidate answers [4]. In the answer selection-based question answering systems, a popular method is to encode the question and each candidate answer as a continuous vector, then calculate the matching score of the two, and finally select the answer with the highest matching score as the answer to the question [5]–[7]. One of the main challenges in the answer selection task is that the size of the candidate answer pool is usually very large, so it is difficult to accurately select the correct answer from it. Intuitively, if we can know the category of the answer, then in general, we can quickly locate the target category, thereby reducing the size of the answer pool. However, in most cases, the answer category

information is not available, and there is no existing method for answer categorization. On the other hand, we note that the category information of the problem is usually available [8]. Moreover, even if the category information of the question is not available, there are existing methods that can accurately categorize questions [9].

Based on the above observations, we propose the Answer Category-aware Answer Selection System (ACAAS). The system improves the accuracy of the question answering system through the fusion of question category information and answer data. Specifically, the system first uses a two-way attention mechanism to encode the target question and answer so that the information of the question and answer can interact with each other in the representation computation. Then, we designed the Attention-Based Shared Label Embedding Network (AB-SLEN), which generates pseudo-labels in the latent space based on question categories and unlabeled answers. The pseudo label of the answer can help the system locate the correct answer more accurately. Finally, ACAAS selects answers based on the encoded questions, answers, and pseudo-labels. We evaluated the performance of ACAAS on three public QA datasets, and the experimental results demonstrate the effectiveness of our model.

II. RELATED WORK

A. ANSWER SELECTION

After being proposed by [1], answer selection became popular as a task. An important component of this task is to encode

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

question texts and answer texts as vector representations and calculate the matching scores based on the representations. Thus, Convolutional neural network (CNN) [10]–[13] and recurrent models like the long short-term memory (LSTM) [11], [12] can be used to encode the input sentences.

There are many different neural network to model question and answer sentence pairs e.g., the attentive pooling with BiLSTM or CNN network [12]; dual Bi-LSTM models with holographic composition approach [14]; a model using hyperbolic representations instead of Euclidean space [15]; a LSTM-based model using temporal gates for question and answer pairs to jointly influence the learned representations in a pairwise manner [16]; a knowledge-based model using a knowledge module to learn a more general representation [17]. In addition, [13] employed a bigram CNN model with average pooling. [10] developed the LSTM model with noise-contrastive estimation approach.

In NLP tasks, many recent works leverage external information to enrich the sentence representation learning of deep learning models [8], [18]. Reference [6] used a positional attention-based RNN model to incorporate the positional context of questions into the answers' representations. Reference [19] used external knowledge bases (KBs) to learn continuous representations, which can enhance the learning of RNN model for machine reading. Reference [20] used a novel attention mechanism from external information sources to enrich the representations learned by transformer networks in conversational search. Similar to these researches, our work aims to jointly leverage the unlabelled candidate answer and the question category label information for answer selection task.

B. ATTENTION-BASED MODELS

Attention-based

models have shown great performance on a range of tasks, such as computer vision and speech recognition [21], [22]. For computer vision tasks, [23] applied attention in RNN models to integrate information over time from an image or video by sequential decision process. [24] combined a novel spatial attention mechanism with a sequential variational auto-encoding framework for image generation. For speech recognition tasks, [25] used an attention mechanism in bi-directional RNN to do the alignment between the input and output sequences. [26] extended the attention mechanism with features needed for speech recognition.

Attention-based models have been applied to NLP tasks after their success in computer vision and speech recognition. To introduce extra source of information for NLP tasks, most prior works used attention mechanism on top of the CNN or RNN model to guide the extraction of sentence embedding. Reference [27] firstly introduced attention mechanism into question answering in RNN models. Reference [28] took the lead in exploring attention mechanisms in CNNs. Reference [29] performed soft-attention alignment by first measuring the matching scores between each word in question and answer.

Most of the aforementioned methods create an attention vector for each of its hidden states during the recurrent iteration, which lead to the models focused on lexical correlations between a certain word and its previous words. Different from them, we propose an attention mechanism, which is only performed once and performed on top of an LSTM layer in our model. It is more focused on the semantics of the whole sentence that each word contributes to, less on relations between words.

C. QUESTION CLASSIFICATION

Question classification is the task that analyzes a question and labels the question based on its expected category information of answer [30]. In answer selection task, we expect it not only provides a semantic constraint on the answer, which allow further processing to precisely locate and verify the answer, but also provides some information for downstream processes in determining answer selection strategies, which may be answer category specific, rather than uniform [31]. Recent works on question classification task focus mainly on machine learning approaches. Reference [32] presented a feature-based machine learning model for question classification. Reference [9] employed a deep learning model and use question categories information to enhance the datasets with highlighting entities in it by a question classification API.¹

To date, the studies on QC are mainly based on the text classification techniques. Many efforts have been made to approach text classification using statistical models based on bag-of-words (BOW) features, such as [33], [34]. But because BOW features lack of semantic meaning representation, neural networks started to be applied to learn word-level representations, such as word2vec. There are also related works that use character-level features for language processing. [35] proposed an alternative low-level representation based on character n-grams. [36], [37] incorporated character-level features to CNNs. In our article, we explore using both word-level and character-level representation and some other kinds of representation of text to introduce more helpful information.

III. PROBLEM STATEMENT

Given a question q , the category of the question c and a set of candidate answers $A = \{a_1, a_2, \dots, a_n\}$, the goal of our system is to select the correct answer of q from A . Concretely, for each question-answer pair $\{q, a_i\}$, the system takes q , c , and a_i as the input, and outputs a scalar that indicating the matching score of the question-answer pair. We learn a function $f(q, a_i, c)$ to compute the matching scores. The scores are then used to rank a list of possible candidates. The higher the matching score is, the corresponding answer is more likely to be the correct answer of the question.

IV. METHODOLOGY

In the proposed method, we compute a matching score against each question answer pair for answer selection. The question category information is used to improve the scores of

¹www.harishmadabushi.com/research/questionclassification/

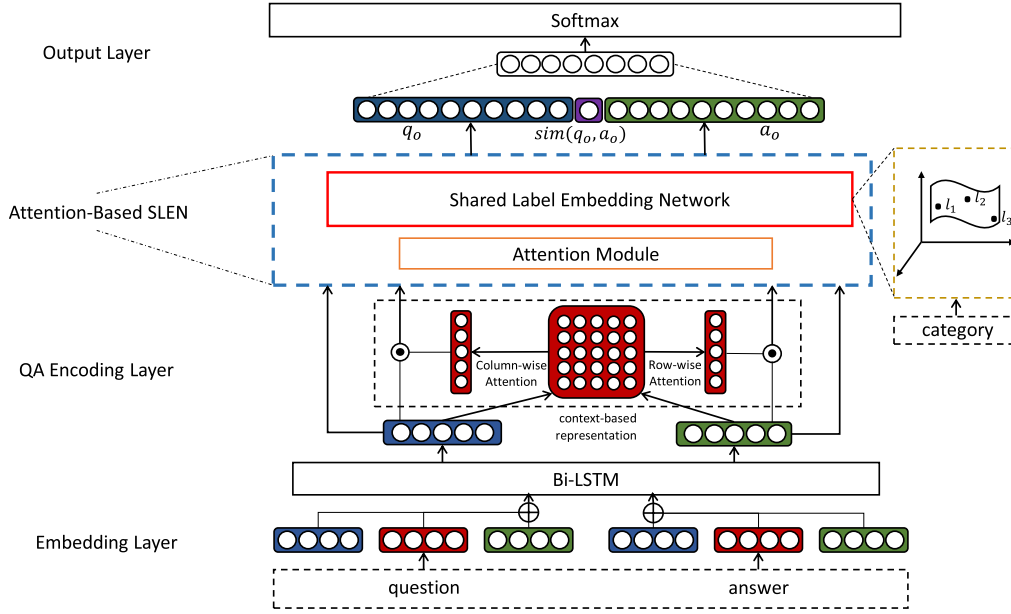


FIGURE 1. Answer Category-Aware Answer Selection system.

positive question-answer pairs and reduce the scores of negative question-answer pairs, individually. Considering answers are unlabelled in the answer selection datasets, we leverage the predictions of question classification module to estimate a category label for the answer. Concretely, we first use an embedding layer to convert questions and answers into vector representations (Section IV-A). Then we adopt bidirectional long short-term memory (BiLSTM) and a two-way attention mechanism to learn a context-based sentence representation of questions and answers in QA encoding layer (Section IV-B). Afterwards, an Attention-Based Shared Label Embedding Network (AB-SLEN) is designed to learn the category-based attentive sentence representation by embedding labels in latent space and providing pseudo-labels for unlabeled answer sentences (Section IV-C). Finally, we compute the bilinear matching score between the category-based attentive question and answer vectors and pass into a fully connected hidden layer before a 2-class softmax layer in output layer (Section IV-D). The overview of our model is shown in Fig. 1. In this section, we describe our answer category-aware answer selection system (ACAAS) layer-by-layer. Note that, we use the subscripts q, a to indicate whether a parameter belongs to question or answer respectively.

A. EMBEDDING LAYER

Given a question q and a set of candidate answers $A = \{a_1, a_2, \dots, a_n\}$, we first use an embedding layer to convert them into vector representations, then input these embedding vectors into the our answer selection system. For a given sentence, it is represented by the embeddings of the words it consists of. The character feature and position feature are also transformed into real-valued representation. The question representation is given by:

$$E = E_{word} \oplus E_{char} \oplus E_{pos}, \quad (1)$$

where $\oplus$ indicates the concatenation operation. $E \in \mathbb{R}^{m \times d}$, m denotes the length of padded sequence and d is the dimension of the embedding.

B. QA ENCODING LAYER

In QA encoding layer, we adopt bidirectional long short-term memory (BiLSTM) as a shared encoder to encode questions and answers. BiLSTM not only has access to future contexts but also captures information from past contexts. The input of BiLSTM layer is a sequence of word embeddings E . The output of the i -th word is represented by $h_i = [\vec{h}_i : \overleftarrow{h}_i]$, in which $\vec{h}_i$ and $\overleftarrow{h}_i$ are the outputs of the forward network and the backward network, respectively. Accordingly, the initial contextual sentence representation $H \in \mathbb{R}^{2u \times m}$ for both question and answer will be:

$$H_q = \text{BiLSTM}(E_q); \quad (2)$$

$$H_a = \text{BiLSTM}(E_a). \quad (3)$$

Based on these initial contextual sentence representations, we employ a two-way attention mechanism to interactively learn the attentive representation of question and answer. With question sentence representation H_q and answer sentence representation H_a , we compute the correlation matrix $F \in \mathbb{R}^{m \times m}$ by introducing an attentive matrix $U \in \mathbb{R}^{2u \times 2u}$ as follows:

$$F = \tanh(H_q^\top U H_a). \quad (4)$$

Afterwards, we conduct max pooling operation along rows and columns of F to generate context-based attention vectors for question f^q and answer f^a separately, where $f^q, f^a \in \mathbb{R}^m$. The co-attentive context-based representation of questions and answers can be computed as:

$$q_{out} = H_q f^q, \quad (5)$$

$$a_{out} = H_a f^a. \quad (6)$$

C. ATTENTION-BASED SHARED LABEL EMBEDDING NETWORK

Attention-Based Shared Label Embedding Network (AB-SLEN) has two components: Attention Module and Shared Label Embedding Network (SLEN).

Attention Module uses attention mechanisms to enhance the presentation of questions and answers. Specifically, we use a self-attention mechanism to avoid computing an annotation vector over step each time, which more focus on the semantics of the whole sentence. The attention mechanism takes the initial contextual sentence representation H_a , H_q as input, and outputs a matrix of weights:

$$W_q = \text{softmax}(W_{c2} \tanh(W_{c1} H_q)). \quad (7)$$

$$W_a = \text{softmax}(W_{c2} \tanh(W_{c1} H_a)). \quad (8)$$

where $W_q, W_a \in \mathbb{R}^{d_a \times 2u}$ is a weight matrix, u is the hidden units size. And $W_{c2} \in \mathbb{R}^{r \times d_a}$ is a weight matrix of parameters, where d_a and r are hyperparameter we can set arbitrarily. In the conventional attention mechanism, W_{c2} is usually a vector belonging to $\mathbb{R}^{d_a}$. Therefore, the final representation tends to focus on specific components of the sentence, such as a special set of related words or phrases. Considering that a sentence may contain multiple semantic components that make up the entire sentence, we introduce W_{c2} by extending r .

Accordingly, the attentive question representation Q and answer representation A instead of H as input of similarity calculation object in (14):

$$Q_{att} = W_q H_q. \quad (9)$$

$$A_{att} = W_a H_a. \quad (10)$$

Then we use a fully connected layer to obtain the attentive representation $h_q \in \mathbb{R}^{d_l}$ and $h_a \in \mathbb{R}^{d_l}$:

$$h_q = W_{f1} \text{flat}(Q_{att}) + b_{f1}. \quad (11)$$

$$h_a = W_{f2} \text{flat}(A_{att}) + b_{f2}. \quad (12)$$

where *flat* indicates the flatten operation.

The Shared Label Embedding Network uses the question category label as the training signal to find a joint space of category label and answer, and then projects the representation of the unlabelled answer to the joint space to generate the answer category pseudo label. Concretely, we first use a trainable matrix L for the representation of the question category labels, where $L \in \mathbb{R}^{t \times d_l}$, t is the number of labels, d_l is the dimension of the label embedding. In order to let the question representation h_q and answer representation h_a interact with L , we introduce a similarity calculation object to measure the similarity between the embedded label and the input representation, which is similar to the Universal Schema Latent Feature Model introduced by [38]:

$$s_q = L \cdot h_q, \quad (13)$$

$$s_a = L \cdot h_a, \quad (14)$$

where $(\cdot)$ is the dot product.

Then we use a softmax function to obtain probability distribution:

$$p_q = \text{softmax}(s_q) \quad (15)$$

$$p_a = \text{softmax}(s_a) \quad (16)$$

The pseudo-labelled embedding of question and answer are computed by:

$$o_q = \sum_{j=1}^t p_{qj} l_j, \quad (17)$$

$$o_a = \sum_{j=1}^t p_{aj} l_j, \quad (18)$$

The loss function of AB-SLEN is the squared error between pseudo-labelled embedding o_q and the embedding of the question category l_c in the label embedding matrix L :

$$L_{pseudo} = \text{MSE}(o_q, l_c) = \|o_q - l_c\|^2 \quad (19)$$

By replacing the question text with the answer text as input, we can also obtain the pseudo-labelled embedding of answer as (18). To take full use of the label information, we consider these label embeddings as the input of matching score computation in 22 and concatenate them to the existing vectors as detailed in (20), (21):

$$q_o = [q_{out} : o_q]. \quad (20)$$

$$a_o = [a_{out} : o_a]. \quad (21)$$

The category label information of question and answer is included in the representation, which helps to reduce the search space of potential answers, making the discovery of answers more effective and accurate.

D. OUTPUT LAYER AND OPTIMIZATION

Finally, we compute the bilinear matching score between the category-based attentive question and answer vectors:

$$\text{Score}(q_o; a_o) = q_o^\top W_s a_o, \quad (22)$$

where $W_s \in \mathbb{R}^{2u \times 2u}$ is a matching matrix to be learned. Note that, ACAAS without AB-SLEN take q_{out} and a_{out} instead of q_o and a_o as the input of matching score computation.

Given the matching score $\text{Score}(q_o; a_o)$ and the category-based representation q_o, a_o , we concatenate them together $X = [q_o : \text{Score}(q_o; a_o) : a_o]$ as the input of the hidden layer. Then the output of the hidden layer goes through a softmax layer for binary classification:

$$y_a = \text{softmax}(WX + b), \quad (23)$$

where W and b are the parameters to be learned. The model is trained to minimize the overall loss function:

$$L_{all} = L_{pseudo} + L_a. \quad (24)$$

As mentioned above, the first optimization objective L_{pseudo} denotes the squared error in (19). The second optimization objective L_a denotes the cross-entropy loss function:

$$\mathcal{L}_a = \sum_{i=1}^N [y_{a_i} \log p_{a_i} + (1 - y_{a_i}) \log (1 - p_{a_i})] + \lambda \|\theta\|_2^2, \quad (25)$$

TABLE 1. Summary statistics of datasets.

Dataset (train/dev/test)	TREC QA	Yahoo QA	Wiki QA
#Questions	1,229/82/100	50.1K/6.2K/6.2K	873/126/243
#QA Pairs	53,417/1,148/1,517	253K/31.7K/31.7K	20,360/1,130/2,352
%Correct	0.12/0.19/0.19	0.20/0.20/0.20	0.12/0.12/0.13

where p_{a_i} is the output of the softmax layer. θ contains all the parameters of the network and $\lambda \|\theta\|_2^2$ is the L_2 regularization. The training loss function measures how predictive the model is on the training data and the regularization term penalizes the complexity of the model, which helps to avoid overfitting. The parameters of the model can be updated by the Adam optimizer.

V. EXPERIMENTS

A. DATASETS AND METRICS

We evaluate the performance of ACAAS on three popular public datasets: TREC QA [39],² Yahoo QA [14] and Wiki QA [13]. The statistics of the AS datasets used are described in Table 1. Note that our proposed method needs to use the information of question categories, but only TREC QA has the question category labels in [9]. For the rest of the datasets, we use the same question classification method as [9] to categorize the questions. The questions are classified into ABBR, DESC, ENTY, HUM, LOC, NUM. We add an “UNK” label to signify a category which is not in the above category list.

- **TREC QA** [39], is a widely-adopted QA ranking benchmark obtained from the TREC QA Tracks 8-13. TREC QA is a factoid question answering dataset. Following previous works [16], [17], we experiment on the raw TREC QA dataset. We adopt the official evaluation metrics of the mean average precision (MAP) and mean reciprocal rank (MRR) as our evaluation metrics. Since the evaluation metrics are commonplace in ranking tasks, we omit any further details for the sake of brevity.
- **Yahoo QA** [14], is an open-domain community-based dataset collected from CQA platform, which can be used to learn and validate answer selection tasks. Yahoo QA is a moderately large dataset containing 253K QA pairs. For this dataset, we use the same evaluation metrics as that in [14], [16], including Precision@1 and Mean Reciprocal Rank (MRR).
- **Wiki QA** [13], is an open-domain factoid answer selection benchmark. Each question was sampled from real queries of Bing without editorial revision. The candidate sentences were chosen from the summary paragraph of relevant Wikipedia pages directly. We adopt the official evaluation metrics of MAP and MRR as our evaluation metrics.

²Following previous works [16], [17], we experiment on the raw TREC QA dataset.

B. EXPERIMENTAL SETTINGS

For question and answer sentences, we use GloVe embeddings³ of 300 dimensions to initialize the word embedding, which is concatenated with character embedding and position embedding to generate the input of our model. We use position embedding generated by sinusoids of varying frequency as [40] and character embedding are randomly initialized with size 30. The dimension of the label embedding d_l is set to be 100. The maximum length of a sentence is set to be 30.

For the baseline models we compared with, we followed exactly the same parameter settings as those in their original papers. As for our model, the hidden layer size of BiLSTM and the final hidden layer size are both set to be 200. The hyperparameter d_a is set to 400 and r is set to a wide range of values between 10 to 30. The dropout rate is set to be 0.5 and L_2 regularization is set to 0.0001. Our model is trained in batches with a size of 64 and a learning rate of 0.005. All other parameters are randomly initialized from $[-0.1, 0.1]$.

C. EXPERIMENTAL RESULTS

For TREC QA and Yahoo QA, we compare ACAAS against 5 baseline methods: (1) the attentive pooling with BiLSTM or CNN network [12]; (2) dual Bi-LSTM models with holographic composition approach [14]; (3) a model using hyperbolic representations instead of Euclidean space [15]; (4) a LSTM-based model using temporal gates for question and answer pairs to jointly influence the learned representations in a pairwise manner [16]; (5) a knowledge-based model using a knowledge module to learn a more general representation [17]. The experimental results are shown in Table 2.

For Wiki QA dataset, in addition to [12], we compare our method with two more strong baselines: (1) the average pooling with a bigram CNN model [13]; (2) the attentive pooling with BiLSTM or CNN network [12]; (3) a LSTM-based model with noise-contrastive estimation approach [10]. The experimental results are shown in Table 3.

Table 2 and Table 3 report the experimental results of different models on TREC QA, Yahoo QA and Wiki QA. The performances of our baseline models achieve the state of the art. For all reported results, the best result is in boldface and the second best is underlined. There are multiple interesting observations from Table 2 as follows:

(1) We observe that our answer category-aware method ACAAS outperforms a myriad of complex neural architectures. Notably, our answer category-aware method ACAAS outperforms the state-of-the-art results on TREC QA and

³<http://nlp.stanford.edu/data/glove.6B.zip>

TABLE 2. Experimental results on TREC QA and Yahoo QA.

Model	TREC QA		Yahoo QA	
	MAP	MRR	P@1	MRR
AP-CNN [12]	0.753	<u>0.851</u>	0.560	0.726
AP-BiLSTM [12]	0.713	<u>0.803</u>	0.568	0.731
HD-LSTM [14]	0.750	0.815	0.557	0.735
HyperQA [15]	0.770	0.825	0.683	0.801
CTRN [16]	0.771	0.838	0.601	0.755
KAN (AP-BiLSTM) [17]	<u>0.797</u>	0.850	<u>0.744</u>	<u>0.840</u>
ACAAS	0.813	0.878	0.865	0.915

TABLE 3. Experimental results on Wiki QA.

Model	MAP	MRR
CNN-Cnt [13]	0.652	0.665
AP-CNN [12]	0.689	0.696
AP-BiLSTM [12]	0.671	0.684
NCE-DNN [10]	<u>0.701</u>	<u>0.718</u>
ACAAS	0.705	0.719

Yahoo QA datasets by at least about 2% and 11% respectively. We obtain a clear performance gain of 2% - 3% in terms of MAP/MRR on TREC QA and 7% - 11% in terms of P@1/MRR on Yahoo QA against the second-best result of KAN (AP-BiLSTM).

(2) The result of ACAAS significantly outperforms other baselines not leveraging external information. Among these results, AP-BiLSTM performs much worse than KAN (AP-BiLSTM) and ACAAS, which demonstrates the effectiveness of incorporating external information into our model. In general, our model has shown promise by incorporating attentive and category information.

(3) We observe that ACAAS outperforms KAN (AP-BiLSTM), demonstrating that ACAAS will introduce more accurate external information (category information). This is within our expectation since ACAAS introduces category information to precisely improve the computation of the matching score. While KAN (AP-BiLSTM) learn representations by using the knowledge base to enable knowledge transfer from the source domain to the target domains. It will interfere the training of the knowledge module and bring a negative effect on the performance if the target dataset is completely different from the source dataset in domain.

(4) These results show that ACAAS is more robust to larger input texts than others. The relative performance of ACAAS is significantly better on large datasets, e.g., Yahoo QA (253K training pairs) as opposed to smaller ones TREC QA (53K training pairs). We believe that this is due to the fact that our representation in Eq.1 introduces more information with the increasing input texts to enrich overall sentence representations.

From Table 3, we observe a similar result as above. The proposed method shows comparative result on Wiki QA. For these baselines, it makes a significant performance boost to incorporate external information into the overall architecture.

TABLE 4. Comparisons between different categories on TREC QA.

Category	%Correct	A/Q	MAP	MRR
HUM	0.19	16	81.8 (-5.2)	86.6 (-1.9)
ENTY	0.20	11	79.0 (-2.4)	87.4 (-7.6)
NUM	0.15	23	76.7 (-6.1)	84.2 (-3.7)
ABBR	-	-	-	-
LOC	0.37	9	87.4 (-3.8)	90.0 (-4.4)
DESC	0.56	2	100 (-5.3)	100 (-8.3)
ALL	0.19	16	81.3 (-4.5)	87.8 (-4.1)

Form Table 4, we observe more detailed experimental results of incorporating AB-SLEN into the overall architecture. The number in the parenthesis indicates the accuracy decrease over ACAAS without AB-SLEN model on different category data. A/Q indicates the average number of answers to a question. The performance increases at least about 2% on each category data. We observe that ACAAS performs well on data that belong to LOC and DESC since the proportion of correct answers is higher, and the average number of answers to a question is smaller than others. Besides, it is obvious that the accuracy increase over ACAAS without AB-SLEN model on data that belong to LOC and DESC is higher than others, which demonstrates that the smaller searching space of potential answers, the more effective incorporating external information.

D. ABLATION EXPERIMENTAL RESULTS

Ablation Study For a thorough comparison, we report the ablation test on TREC QA, Yahoo QA and Wiki QA to analyze the improvements contributed by each part of our model. The results are shown in Table 5, where (i) **ACAAS without AB-SLEN** stand for the answer selection model without AB-SLEN. (ii) **ACAAS with SLEN** and **ACAAS with AB-SLEN** are the SLEN-based model of answer selection task. (iii) **ACAAS with AB-SLEN** is a model conducted with AB-SLEN. Generally, both factors contribute, and it makes larger performance boosting to integrate category information. Even ACAAS with SLEN achieves competitive results with these strong baselines in 2, which demonstrates the effectiveness of leveraging external category information for answer selection task. This is within our expectation since the answer category-aware representation helps reduce the search space of potential answers, making the discovery of answers more effective and accurate.

TABLE 5. Ablation experimental results.

Model	TREC QA		Yahoo QA		Wiki QA	
	MAP	MRR	P@1	MRR	MAP	MRR
AP-BiLSTM [12]	0.713	0.803	0.670	0.684	0.608	0.761
ACAAS without AB-SLEN	0.768	0.837	0.854	0.907	0.682	0.697
ACAAS with SLEN	<u>0.803</u>	<u>0.868</u>	0.875	0.922	0.699	0.712
ACAAS with AB-SLEN	0.813	0.878	<u>0.865</u>	<u>0.915</u>	0.705	0.719

Question: **how do i start using asp.net?**

AP-BiLSTM: you can check out learnvisualstudio.com for some video tutorials on many topics including asp.net

CAAS: you can check out learnvisualstudio.com for some video tutorials on many topics including asp.net

FIGURE 2. An example of the attention visualization of the answer selection task.

Question: **how much wood would a woodchuck chuck, if a woodchuck chuck wood?**

AP-BiLSTM: none, unless you paid him. nothing is free anymore.

CAAS: none, unless you paid him. nothing is free anymore.

FIGURE 3. An example of the attention visualization of the answer selection task.

E. CASE STUDY OF OUR ATTENTION

The category label information used in ACAAS can reweight the attention weight from Equation 20 and Equation 21 in answer selection task. Due to the limited space, we randomly choose two question-answer pairs and visualize their attention weights. The color depth indicates the importance degree of the words, the darker the more important.

The question in Fig. 2 “how do i start using asp.net?” is categorised under “DESC” (DESCRIPTION). From Fig. 2, we can observe that AP-BiLSTM pays much attention to those words that are similar to the question, such as “asp.net”, while due to our attention mechanism, ACAAS can be used to alleviate the limitation of that with more context and semantic information.

The question in Fig. 3 “how much wood would a woodchuck chuck, if a woodchuck chuck wood?” is categorized under “NUM”. From Fig. 3, we can observe that AP-BiLSTM pays much attention to those words that are not related to numbers, such as “nothing is free”. This limitation can be alleviated decently by category-aware attention mechanism, since there is a strong correlation between “NUM” and “none” in question classification.

The results indicate that leveraging question category label information can aid in attending more valuable information in the answer selection task.

VI. CONCLUSION

In this paper, we propose an answer category-aware answer selection method for question answering. Specifically, we jointly leverage question category labels and unlabeled

answers by using an embedding space between the disparate label spaces and learning transfer functions between the label embeddings. Experimental results on three public datasets show the effectiveness of our proposed method. In the future, we will explore the hidden relations beyond the context by leveraging external knowledge from a text corpus to enrich the representational learning of questions and answers.

REFERENCES

- [1] V. Punyakanok, D. Roth, and W.-T. Yih, “Natural language inference via dependency tree mapping: An application to question answering,” Tech. Rep. [Online]. Available: <http://hdl.handle.net/2142/11100>
- [2] T. P. Sahu, N. K. Nagwani, and S. Verma, “Selecting best answer: An empirical analysis on community question answering sites,” *IEEE Access*, vol. 4, pp. 4797–4808, 2016, doi: [10.1109/ACCESS.2016.2600622](https://doi.org/10.1109/ACCESS.2016.2600622).
- [3] T. P. Sahu, N. K. Nagwani, and S. Verma, “Multivariate beta mixture model for automatic identification of topical authoritative users in community question answering sites,” *IEEE Access*, vol. 4, pp. 5343–5355, 2016, doi: [10.1109/ACCESS.2016.2609279](https://doi.org/10.1109/ACCESS.2016.2609279).
- [4] W. Bian, S. Li, Z. Yang, G. Chen, and Z. Lin, “A compare-aggregate model with dynamic-clip attention for answer selection,” in *Proc. ACM Conf. Inf. Knowl. Manage.*, Nov. 2017, pp. 1987–1990, doi: [10.1145/3132847.3133089](https://doi.org/10.1145/3132847.3133089).
- [5] M. Tan, C. dos Santos, B. Xiang, and B. Zhou, “Improved representation learning for question answer matching,” in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics (Long Papers)*, vol. 1, 2016, pp. 464–473.
- [6] Q. Chen, Q. Hu, J. X. Huang, L. He, and W. An, “Enhancing recurrent neural networks with positional attention for question answering,” in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 993–996, doi: [10.1145/3077136.3080699](https://doi.org/10.1145/3077136.3080699).
- [7] M. Tan, C. dos Santos, B. Xiang, and B. Zhou, “LSTM-based deep learning models for non-factoid answer selection,” 2015, *arXiv:1511.04108*. [Online]. Available: <http://arxiv.org/abs/1511.04108>
- [8] G. Zhou, T. He, J. Zhao, and P. Hu, “Learning continuous word embedding with metadata for question retrieval in community question answering,” in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process. Asian Fed. Natural Lang. Process., ACL*, vol. 1, Beijing, China: The Association for Computer Linguistics, Jul. 2015, pp. 250–259, doi: [10.3115/v1/p15-1025](https://doi.org/10.3115/v1/p15-1025).
- [9] H. T. Madabushi, M. Lee, and J. Barnden, “Integrating question classification and deep learning for improved answer selection,” in *Proc. 27th Int. Conf. Comput. Linguistics, COLING*, Santa Fe, NM, USA, Aug. 2018, pp. 3283–3294. [Online]. Available: <https://aclanthology.info/papers/C18-1278/c18-1278>
- [10] J. Rao, H. He, and J. Lin, “Noise-contrastive estimation for answer selection with deep neural networks,” in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 1913–1916, doi: [10.1145/2983323.2983872](https://doi.org/10.1145/2983323.2983872).
- [11] J. Rao, L. Liu, Y. Tay, W. Yang, P. Shi, and J. Lin, “Bridging the gap between relevance matching and semantic matching for short text similarity modeling,” in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process. (EMNLP-IJCNLP)*, 2019, pp. 5369–5380, doi: [10.18653/v1/D19-1540](https://doi.org/10.18653/v1/D19-1540).
- [12] C. dos Santos, M. Tan, B. Xiang, and B. Zhou, “Attentive pooling networks,” 2016, *arXiv:1602.03609*. [Online]. Available: <http://arxiv.org/abs/1602.03609>
- [13] Y. Yang, W.-T. Yih, and C. Meek, “WikiQA: A challenge dataset for open-domain question answering,” in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 2013–2018, doi: [10.18653/v1/d15-1237](https://doi.org/10.18653/v1/d15-1237).

- [14] Y. Tay, M. C. Phan, L. A. Tuan, and S. C. Hui, "Learning to rank question answer pairs with holographic dual LSTM architecture," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Aug. 2017, pp. 695–704, doi: [10.1145/3077136.3080790](https://doi.org/10.1145/3077136.3080790).
- [15] Y. Tay, L. A. Tuan, and S. C. Hui, "Hyperbolic representation learning for fast and efficient neural question answering," in *Proc. 11th ACM Int. Conf. Web Search Data Mining - WSDM*, 2018, pp. 583–591, doi: [10.1145/3159652.3159664](https://doi.org/10.1145/3159652.3159664).
- [16] Y. Tay, L. A. Tuan, and S. C. Hui, "Cross temporal recurrent networks for ranking question answer pairs," in *Proc. 32nd AAAI Conf. Artif. Intell., (AAAI), 30th Innovative Appl. Artif. Intell. (IAAI), 8th AAAI Symp. Educ. Adv. Artif. Intell. (EAAI)*, New Orleans, LA, USA, Feb. 2018, pp. 5512–5519. [Online]. Available: <https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16632>
- [17] Y. Deng, Y. Shen, M. Yang, Y. Li, N. Du, W. Fan, and K. Lei, "Knowledge as a bridge: Improving cross-domain answer selection with external knowledge," in *Proc. 27th Int. Conf. Comput. Linguistics, COLING*, Santa Fe, NM, USA, Aug. 2018, pp. 3295–3305. [Online]. Available: <https://aclanthology.info/papers/C18-1279/c18-1279>
- [18] G. Zhou and J. X. Huang, "Modeling and learning distributed word representation with metadata for question retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 6, pp. 1226–1239, Jun. 2017, doi: [10.1109/TKDE.2017.2665625](https://doi.org/10.1109/TKDE.2017.2665625).
- [19] B. Yang and T. Mitchell, "Leveraging knowledge bases in LSTMs for improving machine reading," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics, Long Papers*, vol. 1, 2017, pp. 1436–1446, doi: [10.18653/v1/P17-1132](https://doi.org/10.18653/v1/P17-1132).
- [20] H. Hashemi, H. Zamani, and W. Bruce Croft, "Guided transformer: Leveraging multiple external sources for representation learning in conversational search," 2020, *arXiv:2006.07548*. [Online]. Available: <http://arxiv.org/abs/2006.07548>
- [21] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Represent., ICLR*, Y. Bengio and Y. LeCun, Eds. San Diego, CA, USA, May 2015, pp. 1–15. [Online]. Available: <http://arxiv.org/abs/1409.0473>
- [22] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst., Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds.* Montreal, QC, Canada, Dec. 2014, pp. 3104–3112. [Online]. Available: <http://papers.nips.cc/paper/5346-sequence-to-sequence-learning-with-neural-networks>
- [23] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 2204–2212. [Online]. Available: <http://papers.nips.cc/paper/5542-recurrent-models-of-visual-attention>
- [24] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra, "DRAW: A recurrent neural network for image generation," in *Proc. ICML*, Jul. 2015, pp. 1462–1471. [Online]. Available: <http://proceedings.mlr.press/v37/gregor15.html>
- [25] J. Chorowski, D. Bahdanau, K. Cho, and Y. Bengio, "End-to-end continuous speech recognition using attention-based recurrent NN: First results," 2014, *arXiv:1412.1602*. [Online]. Available: <http://arxiv.org/abs/1412.1602>
- [26] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2015, pp. 577–585. [Online]. Available: <http://papers.nips.cc/paper/5847-attention-based-models-for-speech-recognition>
- [27] K. M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst., C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds.* Montreal, QC, Canada, 2015, pp. 1693–1701. [Online]. Available: <http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend>
- [28] W. Yin, H. Schütze, B. Xiang, and B. Zhou, "ABCNN: Attention-based convolutional neural network for modeling sentence pairs," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 259–272, Dec. 2016. [Online]. Available: <https://tacl2013.cs.columbia.edu/ojs/index.php/tacl/article/view/831>
- [29] L. Yang, Q. Ai, J. Guo, and W. B. Croft, "ANMM: Ranking short answer texts with attention-based neural matching model," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manage.*, Oct. 2016, pp. 287–296, doi: [10.1145/2983323.2983818](https://doi.org/10.1145/2983323.2983818).
- [30] H. T. Madabushi and M. Lee, "High accuracy rule-based question classification using question syntax and semantics," in *Proc. COLING, 26th Int. Conf. Comput. Linguistics, Conf., Tech. Papers*, N. Calzolari, Y. Matsumoto, and R. Prasad, Eds. Osaka, Japan: ACL, Dec. 2016, pp. 1220–1230. [Online]. Available: <https://www.aclweb.org/anthology/C16-1116/>
- [31] X. Li and D. Roth, "Learning question classifiers," in *Proc. 19th Int. Conf. Comput. Linguistics, COLING, Howard Int. House Academia Sinica*, Taipei, Taiwan, Aug. 2002, pp. 1–7. [Online]. Available: <https://www.aclweb.org/anthology/C02-1150/>
- [32] X. Li and D. Roth, "Learning question classifiers: The role of semantic information," *Natural Lang. Eng.*, vol. 12, no. 3, pp. 229–249, Sep. 2006, doi: [10.1017/S1351324905003955](https://doi.org/10.1017/S1351324905003955).
- [33] D. Yan, K. Li, S. Gu, and L. Yang, "Network-based bag-of-words model for text classification," *IEEE Access*, vol. 8, pp. 82641–82652, 2020, doi: [10.1109/ACCESS.2020.2991074](https://doi.org/10.1109/ACCESS.2020.2991074).
- [34] Y. H. Lee, D. W. Kim, and M. T. Lim, "A two-level recurrent neural network language model based on the continuous Bag-of-Words model for sentence classification," *Int. J. Artif. Intell. Tools*, vol. 28, no. 1, Feb. 2019, Art. no. 1950002, doi: [10.1142/S0218213019500027](https://doi.org/10.1142/S0218213019500027).
- [35] I. Kanaris, K. Kanaris, I. Houvardas, and E. Stamatatos, "Words versus character n-grams for anti-spam filtering," *Int. J. Artif. Intell. Tools*, vol. 16, no. 6, pp. 1047–1067, Dec. 2007, doi: [10.1142/S0218213007003692](https://doi.org/10.1142/S0218213007003692).
- [36] C. N. dos Santos and B. Zadrozny, "Learning character-level representations for part-of-speech tagging," in *Proc. 31th Int. Conf. Mach. Learn., ICML*, Beijing, China, Jun. 2014, pp. 1818–1826. [Online]. Available: <http://jmlr.org/proceedings/papers/v32/santos14.html>
- [37] Y. Shen, X. He, J. Gao, L. Deng, and G. Mesnil, "A latent semantic model with convolutional-pooling structure for information retrieval," in *Proc. 23rd ACM Int. Conf. Conf. Inf. Knowl. Manage. - CIKM*, 2014, pp. 101–110, doi: [10.1145/2661829.2661935](https://doi.org/10.1145/2661829.2661935).
- [38] S. Riedel, L. Yao, A. McCallum, and B. M. Marlin, "Relation extraction with matrix factorization and universal schemas," in *Proc. Hum. Lang. Technol., Conf. North Amer. Chapter Assoc. Comput. Linguistics*, L. Vanderwende, H. D. III, and K. Kirchhoff, Eds. Atlanta, GA, USA: Westin Peachtree Plaza Hotel, Jun. 2013, pp. 74–84. [Online]. Available: <https://www.aclweb.org/anthology/N13-1008/>
- [39] M. Wang, N. A. Smith, and T. Mitamura, "What is the jeopardy model? A quasi-synchronous grammar for QA," in *Proc. EMNLP-CoNLL, Joint Conf. Empirical Methods Natural Lang. Process. Comput. Natural Lang. Learn.*, Prague, Czech Republic, Jun. 2007, pp. 22–32. [Online]. Available: <http://www.aclweb.org/anthology/D07-1003>
- [40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst., Annu. Conf. Neural Inf. Process. Syst.*, Long Beach, CA, USA, Dec. 2017, pp. 5998–6008. [Online]. Available: <http://papers.nips.cc/paper/7181-attention-is-all-you-need>



WEIJING WU received the B.S. degree in computer science from Peking University, China, in 2017, where she is currently pursuing the M.S. degree. Her current research interests include deep learning and question answering.



YANG DENG received the B.S. degree in computer science from the Beijing University of Posts and Telecommunications, in 2016, and the M.S. degree in computer science from Peking University, in 2019. He is currently pursuing the Ph.D. degree in computer science with The Chinese University of Hong Kong. His research interests include deep learning and question answering.



YUZH LIANG received the Ph.D. degree from The University of Hong Kong, in 2017. She is currently a Research Associate with the School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School. Her research interests include artificial intelligence and natural language processing.



KAI LEI (Member, IEEE) received the B.Sc. degree in computer science from Peking University, China, in 1998, the M.Sc. degree in computer science from Columbia University, in 1999, and the Ph.D. degree in computer science from Peking University, in 2015. From 1999 to 2004, he was with several companies, including IBM T.J. Watson Research Center, Citigroup, Oracle, and Google. He is currently an Associate Professor with the School of Electronic and Computer Engineering, Peking University. His research interests include knowledge graph, named data networking, and blockchain.

• • •