Representation Learning for Question Classification via Topic Sparse Autoencoder and Entity Embedding

Dingcheng Li, Jingyuan Zhang and Ping Li
Cognitive Computing Lab (CCL), Baidu Research, USA
{*lidingcheng, zhangjingyuan03, liping11*}*@baidu.com*

***Abstract*—Deep learning models have achieved great successes these days. There are intensive studies of word representation learning for question classification. As questions are typically short texts, existing techniques are often not effective for extracting discriminative representations of questions just from a limited number of words. This motivates us to exploit additional information beyond words in order to improve the representation learning of questions. On one hand, topic modeling often captures meaningful semantic structures from the question corpus. Such global topical information should be helpful for question representations. On the other hand, entities extracted from question themselves provide more auxiliary information for short texts from a local viewpoint. Together with words, topics and entities, question representations can be substantially improved.**

In this paper, we propose a unified neural network framework by integrating Topic modeling, Word embedding and Entity Embedding (TWEE) for question representation learning. Concretely, we introduce a novel topic sparse autoencoder to incorporate discriminative topics into the representation learning of questions. In addition, both words and entity related information are embedded into the network to help learn a more comprehensive question representation. Empirical experiments show that the proposed TWEE framework outperforms the state-of-the-art methods on different datasets.

***Keywords*-Representation Learning; Topic Sparse Autoencoder; Entity Embedding; Question Classification**

I. Introduction

Question answering (QA) is the basic activity of daily human communications. Over the past years, online question answering websites, such as **quora.com** and **stackoverflow.com**, have become increasingly popular for sharing knowledge on a wide range of subjects. People can ask questions in diverse categories through these platforms. Due to the large volumes of questions arriving every second, the first and key step is to effectively understand questions. A better question understanding will help build a more efficient online communication systems. The problem of question understanding and classification has received considerable attention in the last few years [1–6].

The conventional approaches focus on representation learning for question classification as shown in Figure 1. *Bag-of-words representation models* simply construct language models with words or tokens, including the deep average network [1, 7], word autoencoders [8], etc. These methods ignore word orders during the learning process.

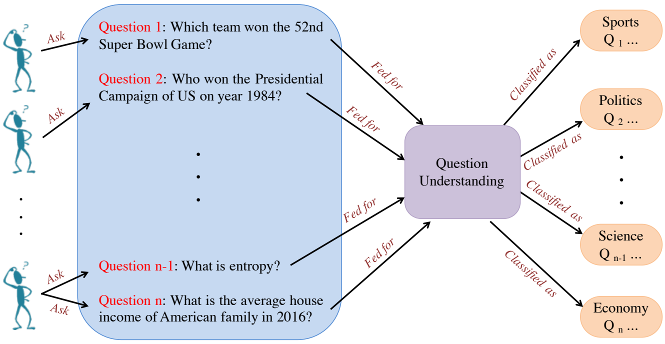
Ask
Ask
Ask
Ask
Question 1: Which team won the 52nd Super Bowl Game?
Question 2: Who won the Presidential Campaign of US on year 1984?
Question n-1: What is entropy?
Question n: What is the average house income of American family in 2016?
Fed for
Fed for
Fed for
Fed for
Question Understanding
Classified as
Classified as
Classified as
Classified as
Sports Q 1 ...
Politics Q 2 ...
Science Q n-1 ...
Economy Q n ...

Figure 1. A Scenario of question understanding and classification where the left plate are a series of questions asked by users. The middle small plate is procedure of question processing, we call question understanding. On the right, diverse questions are classified into predefined categories.

Sequence representation models, such as Dynamic convolutional neural networks (dynamic CNN) [2] and recurrent neural networks of long short-term memory (LSTM) are proposed to take word orders into consideration. Later, *structured representation models* are proposed to learn question representations. For example, a tree-structured LSTM [4] generalizes LSTMs to tree-structured network topologies. It captures both word orders and internal structures of questions. Furthermore, *attention-based representation models* use the attention mechanism to build representations by scoring words and sentences differently [6].

For learning question representations, one inherent challenge is that questions are typically short texts. The existing approaches often cannot effectively extract discriminative representations of questions from a limited number of words. This motivates us to exploit both entity and topic modeling to improve the representation learning of questions. It is known that topic modeling [9, 10] can capture meaningful semantic structures from the question corpus.

In this paper, we propose a unified neural network framework by integrating Topic modeling, Word embedding and Entity Embedding (TWEE) for question representation learning. In particular, we introduce a Topic Sparse AutoEncoder (TSAE) by integrating a probabilistic topic modeling algorithm into a sparse autoencoder. Topic distributions of questions are generated from a global viewpoint and are utilized to enable autoencoder to learn topical repre-

Table I
LIST OF NOTATIONS

Notation	Definition and description
V, N, M and K	Numbers of words, entities, questions and topics
D_t, D_e and D_w	Dimensions for topic, entity and word related embeddings
$\mathbf{x} \in \mathcal{R}^V$ and $\mathbf{a} \in \mathcal{R}^{D_t}$	Bag-of-word for a question and vector for the topic related embedding
$\mathbf{u}_e \in \mathcal{R}^K$ and $\mathbf{e}_e \in \mathcal{R}^{D_e}$	One-hot vector for an entity e and vector for the entity embedding
$\mathbf{v}_w \in \mathcal{R}^V$ and $\mathbf{e}_w \in \mathcal{R}^{D_w}$	One-hot vector for a word w and vector for the word embedding
$\mathbf{T}_w \in \mathcal{R}^{V \times K}$ and $\mathbf{T}_q \in \mathcal{R}^{M \times K}$	Topic distributions over words and questions
$\mathbf{h} \in \mathcal{R}^{D_t \times K}$	Topic distribution for the topic related embedding
$\mathbf{W} \in \mathcal{R}^{D_t \times V}$	Weight matrix for Topic Sparse Autoencoder (TSAE)
$\mathbf{b} \in \mathcal{R}^{D_t}$ and $\mathbf{c} \in \mathcal{R}^{D_t}$	Bias vectors for encoder and decoder in TSAE
γ	Regularization parameter for TSAE to prevent over-fitting
ρ and θ	Sparsity parameter and the topic sparsity parameter in TSAE
α and β	Weights of the sparsity term and the topic guidance term in TSAE
$\hat{\mathbf{x}} \in \mathcal{R}^V$	Decoding representation for a question in TSAE
$\hat{\rho}_j$	Average activation of the j-th topic related embedding
$\hat{\theta}_k$	Average activation of topic related embedding for the k-th topic

sentations. A sparsity constraint is added to ensure the most discriminative representations are related to question topics. In addition, both words and entity related information are embedded into the network from different local viewpoints. Together with topic modeling, word embedding and entity embedding, the proposed TWEE model not only explores information from local contexts of words and entities, but also incorporates global topical structures for a more comprehensive representation learning.

In summary, our contributions are the following:

- We propose a unified neural network TWEE for question representation learning by embedding topics, words and entity-related information together.
- We design a novel topic sparse autoencoder (TSAE) to incorporate topic information into a sparse autoencoder for the representation learning process.
- We introduce an interactive mechanism between TSAE, word embedding and entity embedding to coordinate global topics and local contexts of questions.
- We demonstrate the effectiveness of the proposed TWEE model by comparing it with several state-of-the-art methods on question classification.

II. NOTATIONS AND PROBLEM DEFINITIONS

We first introduce the notations used in this paper. We use bold uppercase letters such as $\mathbf{Z}$ to represent matrices, bold lowercase letters such as $\mathbf{h}$ to represent vectors or embeddings, regular upper case letters such as H to represent scalar constants, and regular lowercase letters such as $z^i_{t,h}$ to represent scalar variables. Table I lists notations which are used throughout this paper.

Given a question, we denote its bag-of-word representation as $\mathbf{x} \in \mathcal{R}^V$, where V is the number of words in the question set. We denote D_t, D_e and D_w to be the dimensions for topic, entity and word related embeddings, respectively. We assume the total number of topics is K and each question focuses on only a small amount of topics. Given M questions, a classic topic model, such as LDA [11], can help extract topic distributions $\mathbf{T}_w \in \mathcal{R}^{V \times K}$ over words and $\mathbf{T}_q \in \mathcal{R}^{M \times K}$ over questions. The proposed TSAE will incorporate the topic information $\mathbf{T}_w$ into a sparse autoencoder and learn a topic-related embedding $\mathbf{a} \in \mathcal{R}^{D_t}$. In addition, given N entities extracted from the questions, we apply the skip-gram model [12] to learn an entity-related embedding $\mathbf{e}_e \in \mathcal{R}^{D_e}$. Entity types (e.g., location, person or media) are used for embeddings since they are more relevant and important to the process of question understandings. Similarly, a word embedding $\mathbf{e}_w \in \mathcal{R}^{D_w}$ is learned via the skip-gram model. With the representations $\mathbf{a}$, $\mathbf{e}_e$ and $\mathbf{e}_w$, the proposed TWEE framework coordinates global topics and local contexts of a question to learn its representation for question classifications. The full architecture of the proposed TWEE framework is illustrated in Figure 2.

III. METHODOLOGY

In this section, we introduce the details of the proposed TWEE framework, which integrates topic modeling, word embedding and entity embedding for question representation learning. Firstly, a topic sparse autoencoder (TSAE) incorporates a probabilistic topic modeling algorithm into a sparse autoencoder. The global topical representations of questions will be learned. Then we introduce how word

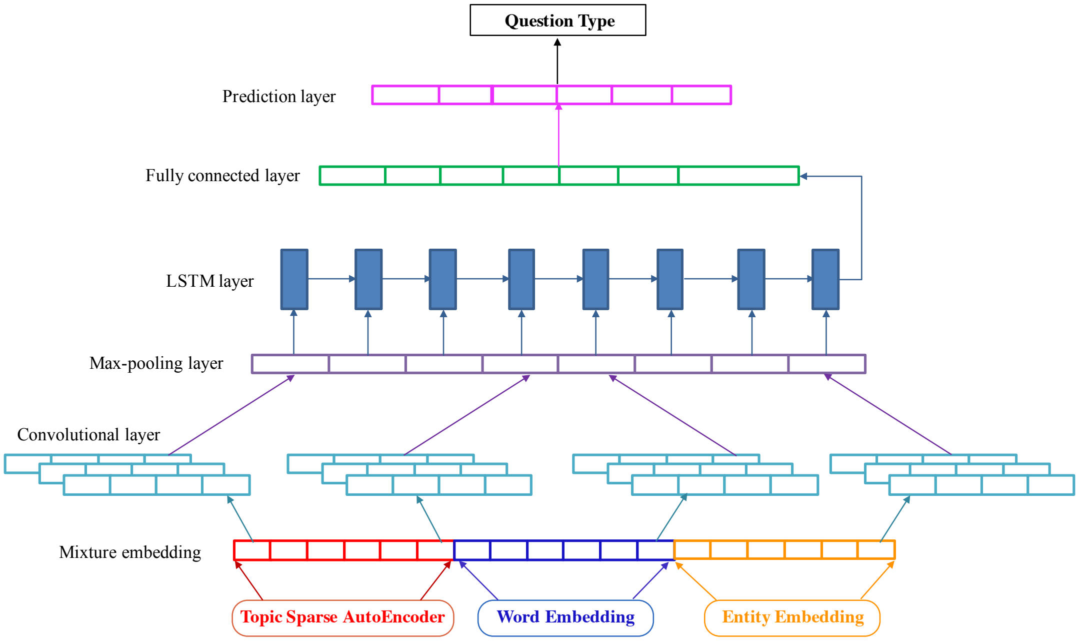

Figure 2. The network structure of TWEE, which is constructed by three input components, namely, the topic sparse autoencoder, the word embedding and the entity embedding. They are concatenated into convolutional layers and passed into LSTM to train a classifier for question types.

embeddings are learned from questions to capture the local context information. Furthermore, we explain how to get entity embeddings to improve the representation learning of questions. Finally, we show how the proposed TWEE framework is built for a more comprehensive representation learning of questions.

A. Topic Sparse Autoencoder (TSAE)

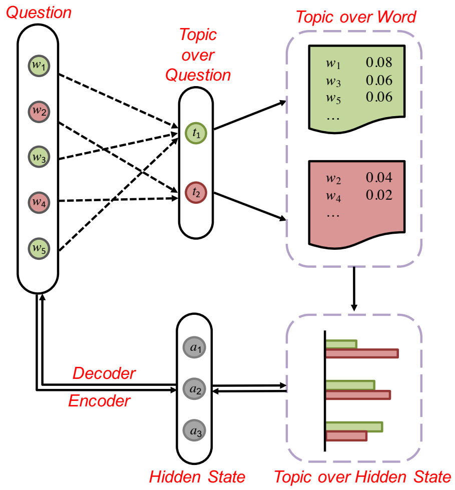

Figure 3. Topic sparse autoencoder for input questions. The encoder is enhanced with topics learned from topic modeling. Topic distributions for words are fed into hidden states so that the representation learning is more discriminative.

In order to learn topic-related representations of questions, we adopt the classic sparse autoencoder (SAE) using the self-reconstruction criterion [13–16]. Autoencoder is an unsupervised feedforward neural network that applies backpropagation by fitting the input using the reconstructed output. It is often used to reduce high-dimensional features and pre-train deep learning models. Basically, SAE encodes the i-th input question $\mathbf{x}_i$ to a hidden representation $\mathbf{a}_i \in \mathcal{R}^{D_t}$ by a feedforward propagation

$$\mathbf{a}_i = f(\mathbf{W}\mathbf{x}_i + \mathbf{b}).$$

Here $\mathbf{a}_i$ is the topic related embeddings for the i-th question. $\mathbf{W} \in \mathcal{R}^{D_t \times V}$ is a weight matrix and $\mathbf{b} \in \mathcal{R}^{D_t}$ is a hidden bias vector. $f(\cdot)$ is the activation function (e.g., the sigmoid function $f(x) = \frac{1}{1+\exp(x)}$ or ReLU). After the feedforward pass, $\mathbf{a}_i$ is decoded to a representation

$$\hat{\mathbf{x}}_i = f(\mathbf{W}^\top \mathbf{a}_i + \mathbf{c}).$$

$\mathbf{c} \in \mathcal{R}^{D_t}$ is a bias vector for the decoder. A sparsity constraint is imposed on the hidden representation of $\mathbf{a}_i$ to reduce noise in SAE. The overall cost function of SAE is

$$\begin{aligned}\mathcal{L}_{SAE}(\mathbf{W}, \mathbf{b}) =& \frac{1}{M}\sum_{i=1}^{M}\frac{1}{2}||\hat{\mathbf{x}}_i - \mathbf{x}_i||^2 \\ &+ \frac{\gamma}{2}||\mathbf{W}||^2 + \alpha\sum_{j=1}^{D_t} KL(\rho||\hat{\rho}_j),\end{aligned}$$

where the first term is the average of reconstruction loss on all questions with sum-of-squares. The second term is a regularization term to prevent over-fitting, where γ is the regularization parameter. It aims to control the sparsity of the weight and bias parameters $\mathbf{W}$ and $\mathbf{b}$. The third term is the

Kullback-Leibler (KL) divergence between two Bernoulli random variables with mean ρ and $\hat{\rho}_j$, respectively:

$$KL(\rho||\hat{\rho}_j) = \rho \log \frac{\rho}{\hat{\rho}_j} + (1-\rho) \log \frac{1-\rho}{1-\hat{\rho}_j}$$

for penalizing $\hat{\rho}_j$ deviating significantly from the sparsity parameter ρ. α is the weight of the sparsity term in the overall cost function. We let

$$\hat{\rho}_j = \frac{1}{M} \sum_{i=1}^{M} a_{ij}$$

be the average activation of the j-th hidden representation. $a_{ij} \in \mathbf{a}_i$ is the j-th hidden unit for the i-th question.

As questions are typically short texts, directly applying SAE to short questions often cannot effectively extract discriminative representations from a limited number of words. Thus, we take advantage of the topical information hidden in questions to improve the representation learning of questions as shown in Figure 3. Our aim is to encapsulate topical information into the overall cost function of SAE so that the learned hidden representations also reflect the topic distributions of questions. In order to achieve this goal, we propose to add the fourth term as a topic guidance term and the goal of the TSAE (topic sparse autoencoder) is to minimize the following objective function:

$$\begin{aligned}\mathcal{L}_{TSAE}(\mathbf{W}, \mathbf{b}) =& \frac{1}{M} \sum_{i=1}^{M} \frac{1}{2} ||\hat{\mathbf{x}}_i - \mathbf{x}_i||^2 + \frac{\gamma}{2} ||\mathbf{W}||^2 \\ &+ \alpha \sum_{j=1}^{D_t} KL(\rho||\hat{\rho}_j) + \beta \sum_{k=1}^{K} KL(\theta||\hat{\theta}_k),\end{aligned}$$

where θ is the topic sparsity parameter for the hidden representations and β is the weight of the topic guidance term in the overall objective function. $\hat{\theta}_k$ is the average activation of the hidden layer for the k-th topic:

$$\hat{\theta}_k = \frac{1}{MD_t} \sum_{i=1}^{M} \sum_{j=1}^{D_t} ||h_{jk}^i||^2,$$

where $h_{jk}^i \in \mathbf{h}_i \in \mathcal{R}^{D_t \times K}$ is the topic distribution of the j-th hidden state over the k-th topic for the i-th question.

$$\mathbf{h}_i = \mathbf{a}_i \mathbf{x}_i^\top \mathbf{T}_w$$

is the topic distribution for the hidden representation $\mathbf{a}_i$.

The topic guidance term is designed for hidden representations learning of $\mathbf{a}$. It reflects the global topical information of questions. The KL divergence $KL(\theta||\hat{\theta}_k)$ helps reconstruct the input with the activation that is related to the most discriminative topics. Figure 3 shows the learning process of TSAE. The global topical information is incorporated into the sparse autoencoder for the representation learning of $\mathbf{a}$.

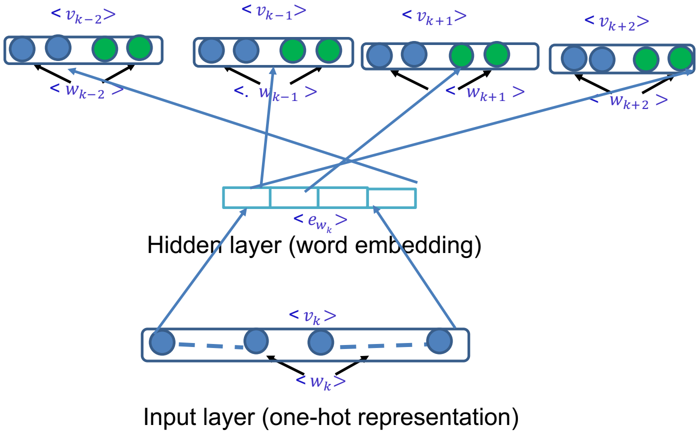

Figure 4. The network for both word and entity embedding learning. Input one-hot representation of words is transformed into low-dimensional vectors. The predictions are context words

B. Word Embedding

The embedding $\mathbf{a}$ learned from the above TSAE module reflects global topical information of questions while the word embedding considers local context information. In this paper, we apply the skip-gram method [12] to learn word embeddings $\mathbf{e}_w$, considering that, as our corpus is composed of short texts and in medical field, there are quite a few number of rare words or phrases. The training objective of the skip-gram model is to learn word representations that are helpful for predicting the nearby words. Given a sequence of training words $S_w = \{w_1, w_2, \cdots, w_s\}$ extracted from questions, the formal objective of the skip-gram model is to maximize the average log probability

$$\mathcal{L}_{word} = \frac{1}{|S_w|} \sum_{w_i \in S_w} \sum_{w_j \in C(w_i)} \log p(w_j|w_i),$$

where w_i is a target word and $C(w_i)$ represents the context words of w_i. $C(w_i)$ is generated by collecting a window of words to the left and to the right of the target word w_i. We use c to denote the window size. The conditional probability $p(w_j|w_i)$ is defined as

$$p(w_j|w_i) = \frac{\exp(\mathbf{e}_{w_j}^\top \mathbf{v}_{w_i})}{\sum_{k=1}^{V} \exp(\mathbf{e}_{w_k}^\top \mathbf{v}_{w_i})},$$

where $\mathbf{v}_w$ is the input one-hot vector representation of word w and $\mathbf{e}_w$ is the corresponding embedding vector representation. V is the number of words in the question vocabulary. Since the cost of computing the derivative of $\log p(w_j|w_i)$ is proportional to V, the stochastic gradient descent with negative sampling [12, 17, 18] is deployed to the skip-gram model. Figure 4 illustrates the learning process of word embeddings.

C. Entity Embedding

The embeddings $\mathbf{a}$ and $\mathbf{e}_w$ learn both global topical information and local contexts from questions. They are helpful for question understandings. Moreover, entities in questions often provide more auxiliary information for short texts from a different local viewpoint. By incorporating entity information into the representation learning process, the understanding of questions can be further improved.

Similar to the word embedding process, we apply the skip-gram method [12] to learn entity embeddings $\mathbf{e}_e$. By maximizing an average log probability, entity embeddings can be learned to help predict nearby entities. The formal objective can be formulated as

$$\mathcal{L}_{entity} = \frac{1}{|S_e|} \sum_{e_i \in S_e} \sum_{e_j \in Q(e_i)} \log p(e_j|e_i),$$

where $S_e = \{e_1, e_2, \cdots, e_s\}$ is a sequence of training entities extracted from questions. e_i is a target entity and $Q(e_i)$ represents the co-occurred entities with e_i in questions. $Q(e_i)$ is generated by collecting a window of entities to the left and to the right of the target entity e_i. We still use c to denote the window size. The conditional probability $p(e_j|e_i)$ can be defined in a similar way as

$$p(e_j|e_i) = \frac{\exp(\mathbf{e}_{e_j}^\top \mathbf{u}_{e_i})}{\sum_{k=1}^{N} \exp(\mathbf{e}_{e_k}^\top \mathbf{u}_{e_i})},$$

where $\mathbf{u}_e$ is the input one-hot vector representation of entity e and $\mathbf{e}_e$ is the corresponding embedding vector representation. N is the number of entities in the questions. The stochastic gradient descent with negative sampling is deployed to speed up the computational time. Its learning process is quite similar to word embeddings and thus we illustrate the learning network with Figure 4 as well.

D. The Full Architecture of TWEE

Together with the topic, word and entity embeddings, the proposed TWEE framework trains a neural classifier for the question type classification as shown in Figure 2. For simplicity, TWEE concatenates the three types of embedding representations together and feeds them into a convolutional layer where multiple filter vectors slide over the embedding sequence to detect features at different positions. The ReLU function is employed on the detected feature maps to do the nonlinear transformations. With n filters, TWEE obtains a successive high-order window representation, which concatenates the feature maps of n filters by column. A max-pooling is applied on the representation to select the most important features. Then LSTM [19] is employed for sequential processing. In the LSTM layer, a range of repeated modules for each time step are defined. Namely, at each time step, the output of the module is a function of the old hidden state and the input of the current time step.

The output is controlled by a set of gates at the current time step, including the forget gate, the input gate and the output gate. These gates collectively decide how to update the current memory cell and the current hidden state. After the LSTM layer is processed, the output of the hidden state at the last time step of LSTM is fed into a fully connected layer for a compact representation of a question. Then a prediction layer using softmax is on the top of the fully connected layer. The cross entropy loss is calculated to make classifications on question types. Back propagations are made at each epoch for the optimal solution of the TWEE framework.

IV. Experiments

In this section, we report extensive experiments to evaluate the proposed TWEE framework.

A. Datasets and Experimental Setup

Two datasets are used in the experiment for the question classification. One is a Chinese medical QA dataset on how patients with diabetes or hypertensions manage daily life. The other dataset is the frequently used TREC dataset [20] for factoid question type classification. Our experiments show that TWEE perform well in both the Medical QA dataset (Chinese) and the Trec dataset (English). We should also explain that the medical QA dataset focuses on the specific topic of diseases while the TREC dataset is more general with diverse topics.

For the medical QA dataset, we aim at classifying the questions into three types, i.e.,"yes-no", "selection" and "description". We collected a total of 100,024 questions and labeled their types by three domain experts with 99% of inter-annotator agreements. A popular text segmentation tool Jieba [1] is used to tokenize the questions and detect entities. The total number of tokens is 37,875. Since the disease related entities are the most important for the medical QA dataset, we map the recognized entities with several medical knowledge resources. The embeddings of entity-related information are trained with random initialization in skip-gram. The word embeddings are initialized with the 300 dimensional pretrained vector representations learned from a large Chinese medical corpus via the GloVe model [21].

For the TREC dataset, there are 5,952 questions with 9,592 words. The questions are divided into 6 categories, including "human", "entity", "location", "description", "abbreviation" and "numeric". The support verbs and lexical answer types are considered as entities for the TREC dataset. They are extracted from the questions and mapped with WordNet [2]. The embeddings of entity-related information are also trained with skip-gram. The word embeddings are initialized with the 300 dimensional pretrained vectors [3] from the Common Crawl of 840 billion tokens and 2.2

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

[2] https://wordnet.princeton.edu/

[3] https://nlp.stanford.edu/projects/glove/

Table II
STATISTICS OF THE DATASETS, WHERE WE USE STANDARD DIVISION OF DATA INTO TRAINING, VALIDATION AND TESTING SETS. CHINESE MEDICAL QA DATASET AND TREC HAVE QUITE DIFFERENT DISTRIBUTIONS IN WORDS (37,875 VS 9592 WORDS) AND ENTITIES (20 VS 2400 ENTITY TYPES)

Datasets	#Classes	#Questions	#Training	#Validation	#Testing	#Words	#Entity Types
Chinese medical QA	3	100,024	70,130	10,045	20,039	37,875	20
TREC	6	5,952	5,000	452	500	9,592	2,400

million vocabularies. The statistics of the datasets are summarized in Table II.

In the experiment, we test the embedding dimensions ranging from 50 to 300. TWEE achieves the best performance when the embedding size is 50 and 100 for the TREC and medical QA datasets, respectively. The number of topics is set as 10 for the TREC dataset and 7 for the medical QA dataset. The regularization parameter γ is set to 0.01 for both datasets. The sparsity parameter ρ and the topic sparsity parameter θ are both set to 0.05 in the experiment. The weights α for the sparsity term and β for the topic guidance term are both set to 0.1.

B. Experimental Results

The results for the medical QA dataset are reported in Table III. For comparisons, we ran two models (CNN and LSTM based) [22] after making little adaptation for question classification to get two groups of results as seen in the first two rows. In Table III, From the fourth row to the end, representation learning for words are obtained with sparse autoencoder (SAE), topic sparse autoencoder (TSAE), integration of TSAE and skip-gram word embedding and finally our proposed TWEE with the integration of TSAE, skip-gram WV and entity embedding (EE) respectively (TSAE+WV+EE+CNN-LSTM). We make use of CNN and CNN-LSTM to train the classifiers to show how much difference between CNN and CNN-LSTM can bring under the context of TSAE. The results show a few trends: topic sparse autoencoder achieves better results than sparse autoencoder; the integration of TSAE and WV boosts the performance; the propose TWEE with the integration of TSAE, WV and entity embedding further improves the classification results.

Table III
THE PERFORMANCE ON THE MEDICAL QA DATASET. WE FOCUS ON DEEP LEARNING METHODS PLUS DIFFERENT WORD VECTORS AIMING AT HIGHLIGHTING THE EFFECTIVENESS OF OUR PROPOSED TSAE METHODS. IT IS CLEAR TO SEE THE INCREMENTAL TREND.

Model	Acc (%)	Pre (%)	Rec (%)	F1 (%)
WV+CNN	92.0	91.5	92.2	91.8
WV+CNN-LSTM	94.1	93.3	92.9	93.1
AE+CNN	51.0	49.2	47.3	48.2
SAE+CNN	78.2	75.5	77.6	76.5
TSAE+CNN	84.5	83.3	84.2	83.7
TSAE+CNN-LSTM	86.0	84.5	85.4	84.9
TSAE+WV+CNN-LSTM	95.0	94.2	93.3	93.7
TWEE	**96.2**	**95.4**	**96.5**	**95.4**

The results for TREC are reported in Table IV, where e compare TWEE with a variety of models. Traditional approaches construct a classifier over a large number of manually engineered features and hand-coded resources. The best classification results with that approach comes from [23]. They trained an SVM classifier with unigrams, bigrams, wh-word, head word, POS tags and hypernyms, WordNet synsets and 60 hand-coded rules and achieved 95% accuracy. Besides SVM, we list the classification performance of other baselines related to CNN or LSTM in Table IV. TWEE consistently outperforms all published neural baseline models. Our result is also better than that of the state-of-the-art SVM that depends on highly engineered features. Such engineered features not only demands human laboring but also leads to the error propagation in the existing NLP tools. With the ability of automatically learning semantic sentence representations, our framework does not require any human-designed features and has a better scalability. Without doubt, entity embedding plays an essential role for the final win-out over that of SVM.

Table IV
THE PERFORMANCE ON THE TREC DATA. BESIDES HIGHLIGHTING EFFECTIVENESS OF PROPOSED TSAE METHOD, WE MADE COMPARISONS WITH RESULTS WHICH ARE AVAILABLE FROM REFERENCED PAPERS.

Model	Acc (%)
SVM [23]	95.0
DCNN [2]	93.0
Group Sparse CNNs [24]	94.2
D-LSTM [19]	94.8
WV+CNN	91.8
WV+CNN-LSTM	93.6
AE+CNN	65.5
SAE+CNN	83.4
TSAE+CNN	87.5
TSAE+CNN-LSTM	92.0
TSAE+WV+CNN-LSTM	94.0
TWEE	**96.5**

C. Parameter Analysis

We provide a study on how the number of topics influence the performance of the proposed TWEE framework. Intuitively speaking, questions belonging to the same category focus on a certain topic. Therefore, the number of topics should be larger than or equal to the number of classes. Figure 5 shows the classification accuracy of TWEE on the TREC and medical QA datasets. For the TREC dataset, we analyze the results with topic numbers of 4, 6, 8, 10 and 12. The best performance is achieved when the topic number is 10. Since the number of classes is 6 for TREC, 10 topics will help distinguish the semantic information from different classes. On the other hand, the performance of TWEE drops when the topic number is larger than 10, which in part reflects the fact that questions are usually short texts. The longest sentence in TREC has only 37 words. More topics cannot help TWEE learn discriminative embeddings from short sentences. Therefore, in the experiment, we set the number of topics as 10 for the TREC dataset.

For the medical QA dataset, we select the number of topics from 3, 5, 7 and 9. Figure 5 (right panel) shows the accuracy of TWEE, which shows that the performance is best when the number of topics is 7, which is larger than 3, the number of classes in the medical QA dataset. Thus in the experiment, we set 7 for the number of topics.

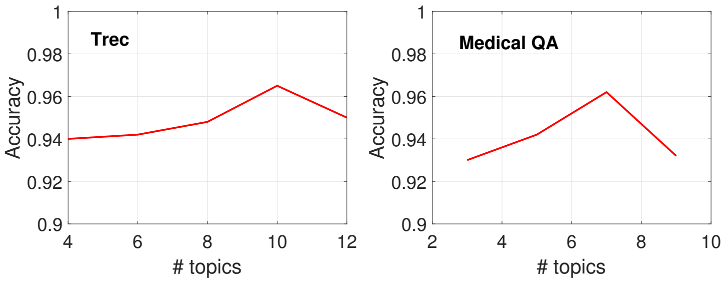

Figure 5. Performance with different topic numbers. As we see, for both the TREC dataset and Medical QA dataset, best performances come from some topic numbers which are higher than the number of classes. This shows some correlations between the class numbers and the topic numbers

V. Related Work

The main theme of this work is to improve the question classification with representation learning. Therefore, the relevant works involve the following two aspects: (1) representation learning; and (2) question classification.

A. *Representation Learning*

Representation learning has been intensively studied and plays an important role for diverse machine learning tasks, classification in particular. The success of deep learning, to a large degree, lies in its embedded capacity of doing representation learning. Word embedding, for example, resolved the issues that deep learning is a framework suitable for continuous data, like image processing while NLP is internally a task of handling discrete data. However, representation learning of words can be further improved if we can usher in the combinations of global context and local context. Word embedding based on skip gram or continuous bag of words is local context focused learning while topic modeling or autoencoder aims at global context learning. There is no existing work which incorporates global topics and local context for representation learning in question classification.

Topic modeling: As illustrated in Subsection III-A, the fourth term of $\mathcal{L}_{TSAE}(\mathbf{W}, \mathbf{b})$ in TSAE aims at adding topic sparsity to autoencoder. In theory, topic sparsity can be derived from diverse clustering results. However, the selection of clustering methods plays an important role in guaranteeing the model robustness. LDA, as a representative of topic modeling, is a powerful unsupervised tool to reveal the latent semantic structure from a text corpus based on its global document-word context information. As a soft-clustering model, LDA does not seek a hard clustering on the documents and the words. Instead, it only assigns topic distributions to them. In the process of back-propagation, LDA generates more suitable clustering constraints to SAE.

B. *Question Classification*

The traditional methods for question classification basically make use of linear classifiers and preprocessed feature vectors to construct classification models. The more recent algorithms construct neural networks, with lower layers focusing on feature extractions and representation learning and the final layer for classification.

Traditional Question Classification: Traditional methods to question classification, like any other traditional machine learning tasks, heavily depend on feature engineering and hand-coded rules before adopting some machine learning models, such as logistic regression or support vector machines [23] or boosted trees [25]. Even though such approaches can construct highly accurate classifiers, they are not robust to unseen datasets. In order to extract discriminative features, those approaches make full use of external resources, including domain ontologies and relevant knowledge graphs. For example, wordNet, the lexical database for English, has been used for synset extractions in the question classification for TREC dataset [26].

Deep Learning Based Question Classification: The first success in deep learning based question classification came from the work of [27], where pre-trained word vectors are fed into a CNN models. As it is known, CNN-based question classification uses linear feature mapping in its convolution operation. Group sparse CNNs [24] is proposed for question classification by making use of information from answer set. CNNs are good at capturing local invariant regularities, but it has the limitation of ignoring word sequence information. On the contrary, recurrent neural network (RNN) represents word sequence with their ordering information. Therefore, quite a few RNN-based works fill this gap. Due to the superior ability to memorize long distance dependencies, LSTMs have been applied to extract the sentence-level continuous representation [28]. The combination of CNNs and LSTMs achieves good performances [29].

VI. Conclusion

We propose TWEE for the task of question classification, by integrating topic modeling, word embedding and entity embedding into a unified neural network framework. The work is inspired by: (1) Topic modeling often captures meaningful semantic structures from the question corpus. Such global topical information are helpful for question representations; (2) Entities extracted from question themselves provide more auxiliary information for short texts from a local viewpoint. In TWEE, we introduce a novel topic sparse autoencoder to incorporate discriminative topics into the representation learning of questions. A sparsity constraint is added to ensure the most discriminative representations are related to question topics. In addition, both words and entities are embedded into the network to help learn a comprehensive question representation. Our extensive empirical experiments on two representative datasets clearly demonstrate that TWEE outperforms the state-of-the-art methods.

References

[1] M. Iyyer, V. Manjunatha, J. Boyd-Graber, and H. Daumé III, "Deep unordered composition rivals syntactic methods for text classification," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, vol. 1, 2015, pp. 1681–1691.

[2] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A convolutional neural network for modelling sentences," *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, 2014.

[3] J. Zhang, X. Kong, R. J. Luo, Y. Chang, and P. S. Yu, "Ncr: A scalable network-based approach to co-ranking in question-and-answer sites," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*. ACM, 2014, pp. 709–718.

[4] K. S. Tai, R. Socher, and C. D. Manning, "Improved semantic representations from tree-structured long short-term memory networks," *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, 2015.

[5] X. Huang, J. Zhang, D. Li, and P. Li, "Knowledge graph embedding based question answering," in *Proceedings of the 12th ACM International Conference on Web Search and Data Mining*. ACM, 2019.

[6] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[7] E. Grave, T. Mikolov, A. Joulin, and P. Bojanowski, "Bag of tricks for efficient text classification," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017, pp. 427–431.

[8] B. Liu, M. Huang, J. Sun, and X. Zhu, "Incorporating domain and sentiment supervision in representation learning for domain adaptation." in *International Joint Conference on Artificial Intelligence*, 2015, pp. 1277–1283.

[9] D. Li, S. Somasundaran, and A. Chakraborty, "A combination of topic models with max-margin learning for relation detection," in *Proceedings of TextGraphs-6: Graph-based Methods for Natural Language Processing*. Association for Computational Linguistics, 2011, pp. 1–9.

[10] D. Li, S. Liu, M. Rastegar-Mojarad, Y. Wang, V. Chaudhary, T. Therneau, and H. Liu, "A topic-modeling based framework for drug-drug interaction classification from biomedical text," in *AMIA Annual Symposium Proceedings*, vol. 2016. American Medical Informatics Association, 2016, p. 789.

[11] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *Journal of Machine Learning Research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in Neural Information Processing systems*, 2013, pp. 3111–3119.

[13] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems*, 2007, pp. 153–160.

[14] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 1096–1103.

[15] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1.

[16] J. Zhang, B. Cao, S. Xie, C.-T. Lu, P. S. Yu, and A. B. Ragin, "Identifying connectivity patterns for brain diseases via multi-side-view guided deep architectures," in *Proceedings of the 2016 SIAM International Conference on Data Mining*. SIAM, 2016, pp. 36–44.

[17] M. U. Gutmann and A. Hyvärinen, "Noise-contrastive estimation of unnormalized statistical models, with applications to natural image statistics," *Journal of Machine Learning Research*, vol. 13, no. Feb, pp. 307–361, 2012.

[18] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," *Proceedings of the 29th International Conference on Machine Learning*, 2012.

[19] Y. Shi, K. Yao, L. Tian, and D. Jiang, "Deep lstm based feature mapping for query classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1501–1511.

[20] X. Li and D. Roth, "Learning question classifiers," in *Proceedings of the 19th International Conference on Computational Linguistics*. Association for Computational Linguistics, 2002, pp. 1–7.

[21] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014, pp. 1532–1543.

[22] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning." in *Symposium on Operating Systems Design and Implementation*, vol. 16, 2016, pp. 265–283.

[23] J. Silva, L. Coheur, A. C. Mendes, and A. Wichert, "From symbolic to sub-symbolic information in question classification," *Artificial Intelligence Review*, vol. 35, no. 2, pp. 137–154, 2011.

[24] M. Ma, L. Huang, B. Xiang, and B. Zhou, "Dependency-based convolutional neural networks for sentence embedding," *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics*, 2015.

[25] P. Li, "Abc-boost: Adaptive base class boost for multi-class classification," in *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM, 2009, pp. 625–632.

[26] E. Brill, J. J. Lin, M. Banko, S. T. Dumais, A. Y. Ng *et al.*, "Data-intensive question answering." in *Text Retrieval Conference*, vol. 56, 2001, p. 90.

[27] Y. Kim, "Convolutional neural networks for sentence classification," *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 2014.

[28] S. Ravuri and A. Stoicke, "A comparative study of neural network models for lexical intent classification," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 368–374.

[29] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," in *Proceedings of the 2015 conference on Empirical Methods in Natural Language Processing*, 2015, pp. 1422–1432.