

# One-shot Learning for Question-Answering in Gaokao History Challenge

Zhuosheng Zhang<sup>1,2</sup>, Hai Zhao<sup>1,2,\*</sup>

<sup>1</sup>Department of Computer Science and Engineering, Shanghai Jiao Tong University

<sup>2</sup>Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, 200240, China

zhangzs@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn

## Abstract

Answering questions from university admission exams (Gaokao in Chinese) is a challenging AI task since it requires effective representation to capture complicated semantic relations between questions and answers. In this work, we propose a hybrid neural model for deep question-answering task from history examinations. Our model employs a cooperative gated neural network to retrieve answers with the assistance of extra labels given by a neural turing machine labeler. Empirical study shows that the labeler works well with only a small training dataset and the gated mechanism is good at fetching the semantic representation of lengthy answers. Experiments on question answering demonstrate the proposed model obtains substantial performance gains over various neural model baselines in terms of multiple evaluation metrics.

## 1 Introduction

Teaching a machine to pass admission exams is a hot AI challenge which has aroused a growing number of research (Guo et al., 2017; Cheng et al., 2016; Clark, 2015; Fujita et al., 2014; Henaff et al., 2016). Among these studies, deep question-answering (QA) task (Ferrucci et al., 2010; Wang et al., 2014) is especially difficult, aiming to answer complex questions via deep feature learning from multi-source datasets. Recently, a highly challenging deep QA task is from the Chinese University Admission Examination (*Gaokao* in Chinese), which is a national-wide standard examination for all senior middle school students in China and has been known as its large scale and strictness. This work focuses on *comprehensive question-answering* in Gaokao history exams as shown in Table 1<sup>1</sup>, which accounts for the major proportion in total scoring and is extremely difficult in the exams. (Cheng et al., 2016) made a preliminary attempt to take up the Gaokao challenge, trying to solve *multiple-choice questions* via retrieving and ranking evidence from *Wikipedia* articles to determine the right choices. Differently, this task is to solve comprehensive questions and has to be based on knowledge representation and semantic computation rather than word form matching in the previous work.

Although deep learning methods shine at various natural language processing tasks (Wang et al., 2015; Zhang et al., 2016b; Qin et al., 2017; Cai et al., 2018; Zhang et al., 2018b; Huang et al., 2018; Zhang et al., 2018a; He et al., 2018; Zhang et al., 2018c; Li et al., 2018), they usually rely on a large scale of dataset for effective learning. The concerned task, unfortunately, cannot receive sufficient training data under ordinary circumstances. Different from previous typical QA tasks such as community QA (Zhang et al., 2016a) which can enjoy the advantage of holding a very large known QA pair set, the concerned task is equal to retrieving a proper answer from textbooks organized as plain texts with guidelines of very limited number of known QA pairs. In addition, the questions are usually given in a quite indirect way to ask students to dig the exactly expected perspective of the concerned facts. If such kind of perspective

---

\*Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100), National Natural Science Foundation of China (No. 61672343 and No. 61733011), Key Project of National Society Science Foundation of China (No. 15-ZDA041), The Art and Science Interdisciplinary Funds of Shanghai Jiao Tong University (No. 14JCRZ04).

This work is licensed under a Creative Commons Attribution 4.0 International License. License details: <http://creativecommons.org/licenses/by/4.0/>

<sup>1</sup>These examples are chosen as they are short, the average number of words in questions and answers are 22 and 47.

fails to fall into the feature representation for either question or answer, the retrieval will hardly be successful.

**Q:** “农民可能充当一种极端保守的角色，也可能充当一种具有高度革命性的角色。”试结合有关史实评析这一观点。

“Peasants may act as an extremely conservative role, or may be highly revolutionary.” Try to analyze this view in the light of the relevant historical facts.

**A:** 农民阶级的经济地位和所处的时代条件决定了它可能充当一种具有高度革命性的角色。以太平天国运动为例，在斗争过程中颁布的《天朝田亩制度》就突出反映了农民阶级要求废除封建土地所有制的强烈愿望，表现出了高度的革命性。农民阶级存在的这种两面性是由其经济地位即受地主阶级压迫和小生产者的地位所决定的。

The economic and social conditions of peasants determine that they may act as a highly revolutionary role. Taking the Taiping Heavenly Kingdom movement as an example, the promulgated regulation “heavenly land system” reflected the peasant class’s demands and strong desire of abolishing the system of feudal land ownership, which shows a strong degree of revolution. The dual nature of the peasant class is also the result of its economic status, that is, under the oppression of the landlord class and the status of the small producers.

**Q:** 有人说“文艺复兴”是一场复古运动，你如何看待？如何评价其意义？

Some people think the Renaissance is a retro campaign. What’s your opinion and how to evaluate its historic significance?

**A:** 文艺复兴运动从表面的含义来看，是一种复兴古希腊罗马时期的哲学、文学和艺术的活动，是一种复古运动，但从其深层的含义看，它却是一场思想解放运动，是一次思想领域里的变革。

The Renaissance is seemingly regarded as a retro movement of philosophy, literature and art of ancient Greece and Rome. However, it is actually an ideological liberation movement and a revolution of thoughts from a deeper inspection.

Table 1: Comprehensive question examples from Gaokao history exams.

Generally speaking, for the Gaokao challenge, knowledge sources are extensive and no sufficient structured dataset is available, while most existing work on knowledge representation focused on structured and semi-structured types (Khot et al., 2017; Khashabi et al., 2016; Vieira, 2016). With regard to the answer retrieval, most current research focused on the factoid QA (Dai et al., 2016; Yin et al., 2016), community QA (Zhang et al., 2017; Lu and Kong, 2017) and episodic QA (Samothrakis et al., 2017; Xiong et al., 2016; Vinyals et al., 2016). Compared with these existing studies, the concerned task is more compositive and more comprehensive and has to be done from unstructured knowledge sources like textbooks. Moreover, the answers of our Gaokao challenge QA task are always a group of detail-riddled sentences with various lengths rather than merely short sentences as focused on in previous work (Yin et al., 2016; Yih et al., 2014).

Recent research has turned to semi-supervised methods to leverage unlabeled texts to enhance the performance of QA tasks via deep neural networks (Yang et al., 2017; Lei et al., 2016). This task is somewhat different from previous ones that the expected extra labels are difficult to be annotated and the entire unlabeled data is kept in a very small scale, so that semi-supervised methods cannot be conveniently applied. Notably, one-shot learning has been proved effective for image recognition with few samples (Li et al., 2006), which is a strategy similar to people learning concept. As an implementation of one-shot learning, neural turing machine (NTM) was proposed (Santoro et al., 2016; Vinyals et al., 2016) and showed great potential by learning effective features from a small amount of data, which caters to our mission requirements. Inspired by the latest advance of one-shot learning, we train a weakly supervised classifier to annotate salient labels for questions using a small amount of examples. For question answering, we propose a cooperative gated neural network (CGNN) to learn the semantic representation and corresponding relations between questions and answers. We also release a Chinese comprehensive deep question answering dataset to facilitate the research.

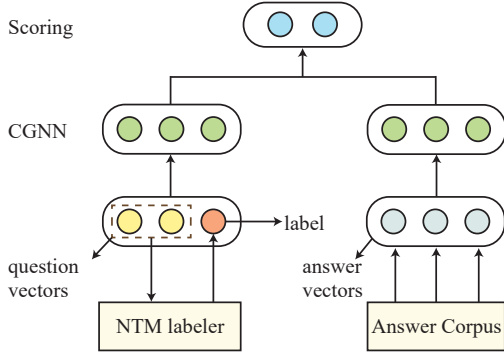


Figure 2: Model architecture.

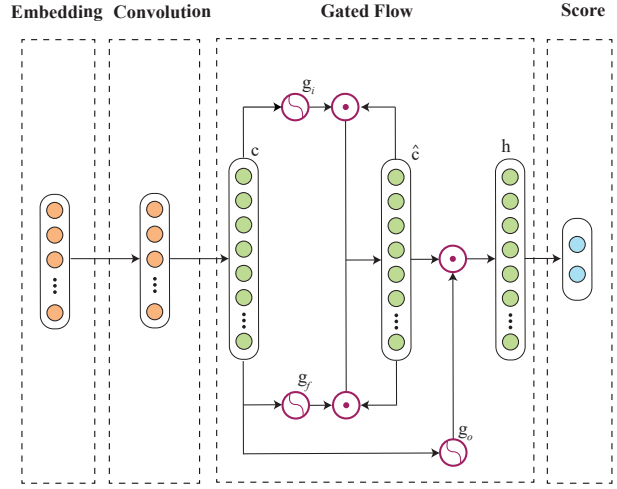


Figure 3: CGNN architecture.

The rest of this paper is organized as follows. The next section introduces our models, including retrieval model, CGNN, and NTM labeler. Task details and experimental results are reported in Section 3, followed by reviews of related work in Section 4 and conclusion in Section 5.

## 2 Model

The proposed hybrid neural model is composed of two main parts: cooperative gated neural network (CGNN) for feature representation and answer retrieval, and NTM as one-shot learning module for extra labeling. As shown in Figure 2, we use NTM to classify the question type and annotate the corresponding label. Then, the concatenated label vectors and question representation are fed to CGNN for jointly scoring candidate answers.

### 2.1 Main framework

For the QA task, the key problem is how to effectively capture the semantic connections between a question and the corresponding answer. However, the questions and answers are lengthy noisy, resulting in poor feature extraction. Inspired by the success of the popular gated mechanism (Wang et al., 2017a; Chen et al., 2016a; Lei et al., 2016) and the gradient-easing Highway Network (Srivastava et al., 2015), a CGNN is proposed to score the correspondence of inputted QA pairs. The architecture is shown in Figure 3.

**Embedding** Our model reads each word of questions and answers as the input. For an input sequence, the embedding is represented as  $\mathbf{M} \in \mathbb{R}^{d \times n}$  where  $d$  is dimension of word vector and  $n$  is the maximum length. When using the NTM module for question type labeling, the question embedding will be refined as the concatenation of the label vectors and its original embedding. Considering the calculation efficiency, we specialize a max number of words for the input and apply truncating or zero-padding when needed.

**Convolutional Layer** Filter matrices  $[\mathbf{W}_1, \mathbf{W}_2, \dots, \mathbf{W}_k]$  with several variable sizes  $[l_1, l_2, \dots, l_k]$  are utilized to perform the convolution operations for input embeddings. Via parameter sharing, this feature extraction procedure is kept the same for questions and answers. For the sake of simplicity, we will explain the procedure for only one embedding sequence. The embedding will be transformed to sequences  $\mathbf{c}_j (j \in [1, k])$ :

$$\mathbf{c}_j = [\dots; \tanh(\mathbf{W}_j \cdot \mathbf{M}_{[i:i+l_j-1]} + \mathbf{b}_j); \dots]$$

where  $[i : i + l_j - 1]$  indexes the convolution window. Additionally, we apply wide convolution operation between embedding layer and filter matrices, because it ensures that all weights in the filters reach the entire sentence, including the words at the margins.

**Gated Flow** To highlight the key information and ignore irrelevant ones during convolution, we adopt an adaptive gated decay mechanism for gated information flow. Three gates are added to optimize the feature representation. These gates are only influenced by the original input through different parameters. Let  $\hat{\mathbf{c}}_j^n$  denote  $n$ -gram features in  $c_j$ , the gates are formulated as

$$\begin{aligned}\mathbf{g}_i &= \sigma(\mathbf{W}_i \hat{\mathbf{c}}_j^n + \mathbf{b}_i) \\ \mathbf{g}_f &= \sigma(\mathbf{W}_f \hat{\mathbf{c}}_j^n + \mathbf{b}_f) \\ \mathbf{g}_o &= \sigma(\mathbf{W}_o \hat{\mathbf{c}}_j^n + \mathbf{b}_o)\end{aligned}$$

where  $\sigma$  denotes sigmoid function which guarantees the values in the gates are in  $[0,1]$  and  $\mathbf{W}_i, \mathbf{W}_f, \mathbf{W}_o, \mathbf{b}_i, \mathbf{b}_f, \mathbf{b}_o$  are model parameters. The transformed inner cell is iteratively calculated by:

$$\hat{\mathbf{c}}_j^n = \hat{\mathbf{c}}_{j-1}^n \odot \mathbf{g}_i + \mathbf{g}_f \odot (\hat{\mathbf{c}}_{j-1}^{n-1} + \mathbf{W}_n)$$

where  $\odot$  denotes element-wise multiplication and  $\mathbf{W}^o$  is the parameter. Through gated flow, the vector  $\mathbf{c}_t$  captures the filtered information. The output of our CGNN is  $\mathbf{h}_j = \tanh(\hat{\mathbf{c}}_j^n) \odot \mathbf{g}_o$ . The gates are used to route information through the flow. Although these gates are generated independently, they work cooperatively since they jointly control the information flow of inner-cells. This procedure will help select salient features. When there is one gate in our network, the model works similarly to the original highway network (Srivastava et al., 2015).

A *one-max-pooling* operation is adopted after the gated flow and the current state vector  $\mathbf{s}_j$  is obtained through concatenating all the mappings for those  $k$  filters, we have  $\mathbf{s}_j = \mathbf{max}(\mathbf{h}_j)$ .

For discriminative training, we use a max-margin framework for learning (or fine-tuning) parameters  $\theta$ . Specifically, a scoring function  $\varphi(\cdot, \cdot; \theta)$  is defined to measure the similarity between the corresponding representations of questions and answers. In this work, we use cosine similarity for the measurement. Let  $p = \{p_1, \dots, p_n\}$  denote the answer corpus and  $a \in p$  is the answer to the question  $q_i$ , the optimal parameters  $\theta$  are learned by minimizing the max-margin loss:

$$\max\{\varphi(q_i, p_i; \theta) - \varphi(q_i, a; \theta) + \delta(p_i, a)\}$$

where  $\delta(\cdot, \cdot)$  denotes a non-negative margin and  $\delta(p_i, a)$  is a small constant when  $a \neq p_i$  and 0 otherwise.

## 2.2 Question Type Labeling

Since examinees studying for exams are required to understand the history knowledge from different sides, we exactly consider using especially-annotated labels as extra indicators for machines to capture such features. With external memory, NTM has shown great potential for one-shot learning (Santoro et al., 2016). As in Figure 3, an NTM consists of two primary components, controller and memory. The controller, often implemented as a recurrent neural network, interacts with an external memory module using soft *read heads* to retrieve representation from memory and *write heads* to store memory, respectively.

**Controller** The controller of NTMs can be implemented as either a recurrent neural network or a feed-forward neural network. Our model adopts LSTM cells as the implementation for the best performance (Graves et al., 2014).

$$\begin{aligned}\mathbf{i}_t &= \sigma(\mathbf{W}_w^i \mathbf{w}_t + \mathbf{W}_h^i \mathbf{h}_{t-1} + \mathbf{b}_i) \\ \mathbf{f}_t &= \sigma(\mathbf{W}_w^f \mathbf{w}_t + \mathbf{W}_h^f \mathbf{h}_{t-1} + \mathbf{b}_f) \\ \mathbf{u}_t &= \sigma(\mathbf{W}_w^u \mathbf{w}_t + \mathbf{W}_h^u \mathbf{h}_{t-1} + \mathbf{b}_u) \\ \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_w^c \mathbf{w}_t + \mathbf{W}_h^c \mathbf{h}_{t-1} + \mathbf{b}_c) \\ \mathbf{h}_t &= \tanh(\mathbf{c}_t) \odot \mathbf{u}_t \\ \mathbf{o}_t &= (\mathbf{h}_t \oplus \mathbf{m}_t)\end{aligned}$$

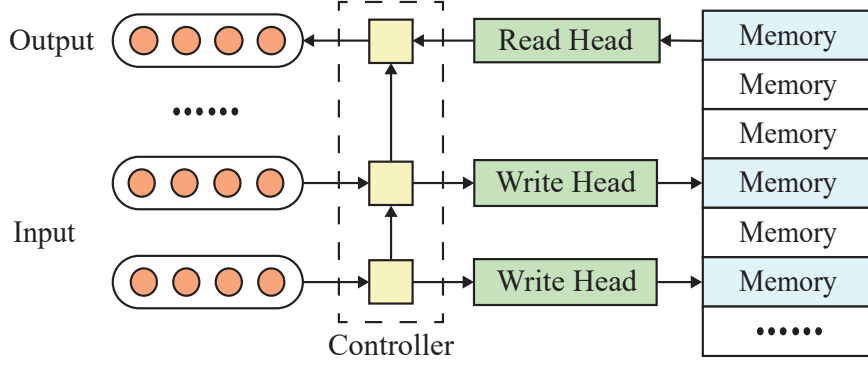


Figure 3: NTM architecture

where  $\oplus$  denotes vector concatenation, and  $\mathbf{i}_t, \mathbf{f}_t, \mathbf{u}_t, \mathbf{c}_t, \mathbf{h}_t$  are the input gates, forget gates, output gates, cell state and hidden state, respectively.

Given the input sequence, the controller computes the concatenated state sequence  $\mathbf{o}_t$  by applying the formulation for each time step. Each word of the sequence is represented as word embedding. In order to simplify the calculation, we apply *one-max-pooling* to each input. As a result, each sequence is represented as a vector with the same dimension of word embedding.

**Memory Manipulation** We use a rectangular matrix  $M \in \mathbb{R}^{N \times M}$  to denote the memory module. Given the read vector  $\mathbf{r}_t$ , the memory  $\mathbf{m}_t$  is retrieved by:

$$\mathbf{m}_t = \mathbf{r}_t \mathbf{M}_t, \mathbf{r}_t = \frac{\exp(\mathbf{K}(\mathbf{k}_t, \mathbf{M}_t))}{\sum_j \exp(\mathbf{K}(\mathbf{k}_t, \mathbf{M}_t))}$$

where  $\mathbf{r}_t$  is read vector and  $\mathbf{K}$  is the similarity score. For writing, the memory is updated by

$$\mathbf{M}_t = \mathbf{M}_{t-1} \cdot (1 - \mathbf{w}_t \mathbf{e}_t) + \mathbf{w}_t \mathbf{c}_t$$

where  $\mathbf{w}_t, \mathbf{e}_t$  and  $\mathbf{c}_t$  represent the write vectors, erase vectors and content vectors respectively.

There are two categories for memory addressing mechanism of the original NTM: content-based addressing comparing each memory locations by some key  $\mathbf{k}_t$  and a location-based addressing by shifting the heads, akin to running along a tape. However, the location-based addressing is not optimal for conjunctive coding of information independent of sequence. Following (Santoro et al., 2016), we use an addressing strategy called the Least Recently Used Access (LRUA).

**Addressing** The weight  $\mathbf{u}_t$  is defined as follows,

$$\mathbf{u}_t = \gamma \mathbf{u}_{t-1} + \mathbf{r}_t + \mathbf{w}_t$$

where  $\gamma$  is a decay parameter. The least-used vectors,  $\mathbf{v}_t$ , are boolean variables. For a given time-step, each element of  $\mathbf{v}_t$  is set to 0 if it is greater than the smallest element of  $\mathbf{u}_t$ , otherwise is 1. The write weights are obtained by

$$\mathbf{w}_t = \sigma(g_t) \mathbf{r}_{t-1} + (1 - \sigma(g_t)) \mathbf{v}_{t-1}$$

where  $g_t$  is a scalar interpolation in the range (0,1) that blends the weights of previous read weights and previous least-used weights.

**Learning** For our labeling task, we define a cost function as the negative log-likelihood:

$$L(\theta) = - \sum_{n=1}^N \mathbf{y}_n \log(\text{Pr}(\hat{\mathbf{y}}_n))$$

where  $\mathbf{y}_n$  is the label and  $\hat{\mathbf{y}}_n$  is the predicted one. After perspective detection, the label is then fed to the CGNN module as one-hot vectors and concatenated with the question representation.

	Train	Test
# QA pairs	964	965
Avg # words in questions	18	25
Avg # words in answers	46	48
Max # words in questions	478	482
Max # words in answers	4,652	4,387
# Candidate answer set	1,929	
# Vocabulary	10,806	

Table 2: Data statistics of the comprehensive question-answering corpus.

### 3 Experiments

**Corpus** Gaokao challenge is an imitation of open-book examination for computers. Therefore, only quite a limited number of resources can be used, which makes common semi-supervised methods unlikely beneficial from large scale unlabeled data. In addition, all our source for the answer should come from standard textbook which contains unstructured plain texts as we initialized our system building.

The corpus is from the standard history textbook<sup>2</sup>. First, to compose an answer set, 1,929 text fragments<sup>3</sup> were extracted by human experts. Then equal numbers of questions were collected from past Gaokao exam papers, and their answers were manually assigned from the answer set to give 1,929 annotated QA pairs, which were equally split for training and test. All text fragments in either questions or answers are segmented into words using *BaseSeg* (Zhao et al., 2006). We publish it to research communities to facilitate the research<sup>4</sup>. Data statistics are in Table 2.

Embedding	Max number of words	$n = 100$
	Word embedding size	$d = 200$
CGNN	Hidden unit number	$h = 200$
	Initial learning rate	$lr = 10^{-3}$
	Regularization	$\nabla = 10^{-5}$
	Dropout	$p = 0.2$
	Filter Width	$k = 4$
NTM	Controller size	$s = 200$
	Read heads	$r = 4$
	Memory shape	$m = (128, 100)$
	Initial learning rate	$lr = 10^{-3}$

Table 3: Hyper-parameters of our model.

<sup>2</sup>Standard Middle-school History Textbook (Vol. 1-3), published by the People’s Education Press in May, 2015.

<sup>3</sup>1,929 is the number of all must-to-be-mastered history facts required by national education quality control.

<sup>4</sup>Our source is available at: <https://github.com/cooelf/OneshotQA>.

Class Labels	Answers
背景 Background	运用我们从古代诗文、戏曲、民间传说中已经学到的知识，举例说明中国古代自给自足的自然经济的状况。 Using the knowledge learned from ancient poetic proses, operas, and folklores, exemplify the situation of self-sufficiency of natural economy in ancient China.
原因 Cause	春秋战国时期是社会剧烈动荡的历史阶段，为什么在这样的时期会出现思想文化活跃的局面。 The Spring-autumn and Warring States Period was the historical stage of the violent social unrest. Why would there be an active ideological and cultural phenomenon in such a period?
主张 Claim	在启蒙运动中，众多的启蒙思想家的共性思想主张是什么？他们之间有何继承和发展。 During the Enlightenment, what is the common thought of those enlightening thinkers? What is the inheritance and development between them?
事实 Fact	“农民可能充当一种极端保守的角色，也可能充当一种具有高度革命性的角色。”试结合有关史实评析这一观点。 “Farmers may act as an extremely conservative role, or may be highly revolutionary.” Try to analyze this view in the light of the relevant historical facts.
意义 Influence	用历史唯物主义和辩证唯物主义的观点来分析古代雅典民主政治和罗马法发展的历程，了解它们对后世的作用和影响。 From the perspective of historical and dialectical materialism, analyze the history of Athens’s democratic politics and the development of Rome law in ancient times, and understand their roles and influences on later generations.

Table 4: Examples of Text Fragments of 5 classes.

**Setup** For the sake of computational efficiency, a maximal length of 100 words for each text fragment is specialized and truncating or zero-padding will be applied as necessary. Word embedding is trained by word2Vector (Mikolov et al., 2013) using *Wikipedia* corpus<sup>5</sup>.

The diagonal variant of AdaGrad (Duchi et al., 2011) is used for neural network training. Table 3 shows the hyper-parameters of our models. For other neural models, we fine-tune the hyper-parameters with the following range of values: learning rate  $\in \{1e-3, 1e-2\}$ , dropout probability  $\in \{0.1, 0.2\}$ , CNN filter width  $\in \{2, 3, 4\}$ . The hidden dimensions for all the neural networks are 200.

In the following experiments, we use NTM to annotate salient labels for each question. Then, the question representation and corresponding label vectors are concatenated and fed to CGNN for feature learning.

**Labeling** Examinees are required to understand the history knowledge from different sides. Thus, the solver should also be capable of capturing such kind of features. To specify a key perspective on historical questions, five classes<sup>6</sup> (*background, cause, claim, fact and influence*) are annotated as shown in Table 4. Note that such kind of annotation is even not easy for human annotators or history teachers and our complete set for all answers is very small (less than 2,000). At last, we have 80 answers per class annotated for a total number of 400, in which 50 are selected for training<sup>7</sup> and 350 for test. For our detailed case, our data is too precious to be used too many for training.

One-shot learning is supposed to effectively represent the questions in the feature space with a small set of training data. To evaluate its performance, we specify a fixed number of training data. At each epoch, NTM is randomly provided with a specific number (shot) of samples per class from training set according to the typical training setting of one-shot learning (Santoro et al., 2016). One-shot learning, in previous practice, usually still took a large candidate training set (i.e., hundreds) as input, though for

<sup>5</sup><https://dumps.wikimedia.org/>

<sup>6</sup>These classes are determined according to national education quality control who requires every student should clearly distinguish these specific perspectives about history facts. More details are in Supplementary Materials.

<sup>7</sup>As our labeling is a five-class classification task, 50 samples therefore allow 10-shot learning per class at most.

Method	1-shot	2-shot	5-shot	10-shot
Naive Bayes	22.1	25.7	30.1	34.1
K-Means	12.5	15.7	15.8	16.1
SVM	19.1	21.8	30.2	<b>45.2</b>
Feedforward	20.1	22.7	21.3	21.6
LSTM	23.8	28.5	32.5	38.2
NTMs	<b>38.7</b>	<b>49.1</b>	<b>63.9</b>	<b>76.6</b>

Table 5: Accuracy for labeling task.

each class, only a small fixed number of shots (i.e., less than 10) were randomly selected and fed for learning. However, it is quite different from our data reality and we have to keep a least training set (10 for our case) for one-shot learning.

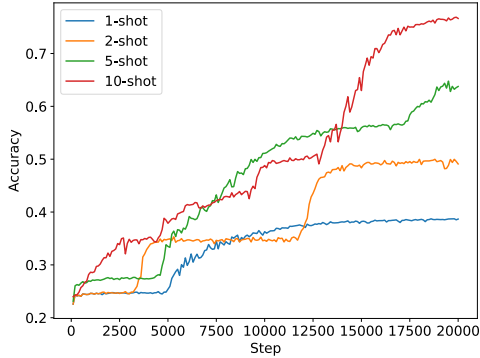


Figure 4: Labeling accuracy curve.

Table 5 compares NTM with traditional classifiers (Naive Bayes, K-Means and SVM) and neural networks (Feedforward network and LSTM). All the baseline methods are fed the training instances as the same shot style of NTM and parameters are tuned for best performance.

Figure 4 illustrates NTM learning curves with different shots. The results show that NTM works effectively on a small amount of samples. With memory cells as well, LSTM is chosen as the baseline. The hidden layer dimension of the LSTM is 200, which is the same as the NTM controller size. The LSTMs are fed with the training instances as the same shot style of NTM. Results shown in Table 5 indicate only NTM provides satisfactory performance with the least input instances. The effectiveness of NTM may attribute to its capability of slowly learning the representations of raw data through weight updates, and rapidly binding relevant information after a single representation via an external memory. To that extend, the NTM could generalize the meta features of each question type and distinguish the perspectives. This indicates one-shot learning is able to lessen over-fitting issues from sparse features of limited data.

**Answer Retrieval** For a discriminative training, we add negative QA pairs by randomly selecting 20 false answers for each positive QA pairs in the original training set. For evaluation, all answers in the corpus are ranked to match each question. NTM labeler is from 10-shot training.

We use baselines including BM25 implemented by a standard search engine *Apache Lucene*<sup>8</sup> and other neural models, Convolutional Neural Network (CNN), LSTM, Gated Recurrent Unit (GRU). Our evaluation is based on the following metrics: Precision, Recall and F1-score, which are widely used for relevance evaluation.

Table 6 presents the experimental results of all the models with and without NTM labels. All of the neural networks greatly outperform BM25 which is purely based on word form matching. In addition, NTM labeler boosts the performance of all the methods. For instance, our CGNN model has obtained 17.7% gains on the F1-score metric with the assistance of extra labeling. Furthermore, our model using CGNN and NTM labeler outperforms all the other baselines. This superior performance indicates that the one-shot learning strategy is competent for our deep QA task with only a small amount of data and

Method	Precision	Recall	F1-score
BM25	8.6 (8.7)	23.2 (23.7)	12.5 (12.7)
CNN	9.8 (27.0)	21.1 (45.0)	13.4 (33.8)
GRU	11.4 (32.9)	24.1 (50.4)	15.5 (39.8)
LSTM	8.4 (30.6)	21.2 (48.0)	12.0 (37.4)
+	<b>22.2</b>	<b>26.8</b>	<b>25.4</b>
CNN+Highway	15.8 (33.2)	30.5 (50.6)	20.8 (40.1)
CNN+GRU	17.4 (32.4)	31.1 (50.8)	22.3 (39.6)
CNN+LSTM	12.7 (27.5)	21.4 (47.0)	16.0 (34.7)
CGNN	<b>18.2 (33.7)</b>	<b>31.6 (51.6)</b>	<b>23.1 (40.8)</b>
+	15.5	20.0	17.7

Table 6: Answer retrieval performance without and with NTM labels (in brackets). ”+” means the performance gains with the help of NTM labels.

<sup>8</sup><http://lucene.apache.org/>



Type	Proportion	Precision	Recall	F1-score
Background	16.0	38.0	56.3	45.4
Cause	8.4	48.6	67.5	56.5
Claim	4.7	47.8	65.1	55.1
Fact	51.0	23.0	42.0	29.7
Influence	19.9	39.0	54.5	45.5

Table 7: Question type distribution and model performance of CGNN.

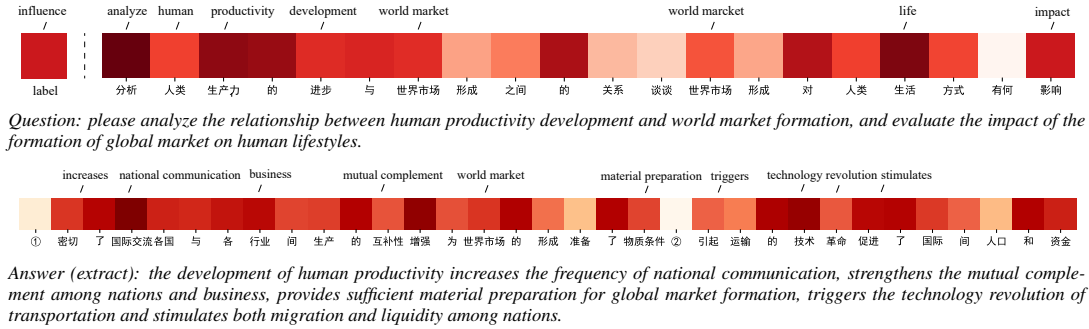


Figure 5: Visualization for question and answer representation weights after gated flow. The darker color means the higher weights.

the adoption of gated mechanism is also well-suited to work with CNN, adaptively transforming and combining local features detected by the individual filters. Table 7 shows question type distribution and the performance of CGNN. Our model performs well on most types but drops on *Fact* since these questions are diverse kinds of facts whose patterns are extremely hard to distinguish, even for human. This also shows our task is challenging.

**Model Analysis** The primary motivation of highway network is to ease gradient-based training of highly deep networks through utilizing gated units. When remaining one gate in our network, our model works similarly to the highway network. To compare different gated mechanisms, we carry out further experiments with the same CNN setting. As shown in Table 6, CGNN achieves the best performance in terms of all evaluation metrics, indicating our gated mechanism is well-suited to work with CNN, adaptively transforming and combining local features detected by the individual filters.

In order to give an insight into the effectiveness of the gated mechanism on information flow across neural network layers, we visualize the final weights for each word from question and answer after gated flow as shown in Figure 5. We can see that the key information (*human, productivity, development, world market, life and influence*) of the lengthy questions and answers will be given higher weights, especially the common words (*world market*) of the question and corresponding answer. Besides, the label in the question is also assigned a high attention, indicating the label works essentially.

## 4 Related Work

### 4.1 Question Answering

Various neural models have been proposed to tackle the tasks of QA task and related knowledge representation (Wang et al., 2017b; Chen et al., 2016b; Todor and Anette, 2018; Kundu and Ng, 2018). Previous work mainly focused on factoid questions, falling into the architecture of knowledge base embedding and question answering from knowledge base (Khashabi et al., 2016; Angeli et al., 2015). Yang et al. (2014) proposed a method that transforms natural questions into their corresponding logical forms and conducted question answering by leveraging semantic associations between lexical representations and knowledge base properties in the latent space. Yin et al. (2016) proposed an end-to-end neural model that can generate answers to simple factoid questions from a knowledge base. Nonetheless, the knowledge base construction requires a lot of human workload, and the related semantic parsing and knowledge representation will become much more complicated as the scale of the knowledge base increases.

For non-factoid QA, most work formulized it as a semantic matching task on sentence pairs by vector modeling. Different from answer retrieval from knowledge bases, this type of studies used vector to represent QA pairs and compare the distance in vector space to match answer text (Tan et al., 2015; Feng et al., 2015). However, the performance of these neural models greatly depends on a large amount of labeled data.

Our concerned task cannot be simply regarded as either factoid or non-factoid. In fact, questions in our task consist of both factoid and non-factoid ones with purely unstructured corresponding answers. Conventional networks might not be sufficient to represent these QA pairs and we need to seek more powerful models. A recent hot comprehensive QA task is the SQuAD challenge (Rajpurkar et al., 2016) which aims to find a text span from given paragraph to answer the question. However, our task is quite different from SQuAD. With sufficient learning data, neural models have shown satisfying performance in the SQuAD challenge. In real world practice, examinees are acquired to learn and comprehend meta-knowledge from textbooks so as to pass the exams through review and reasoning among the whole knowledge space instead of simply finding an answer span from a given paragraph, which shows to be challenging for neural networks.

## 4.2 Semi-supervised Learning

For real-world applications of specific domains, the datasets are always inadequate. Semi-supervised Learning methods have been extensively studied to make use of unlabeled data. A batch of models have been proposed based on representation learning (Yang et al., 2017; Lei et al., 2016). For question answering, current semi-supervised models generally use auto-encoder framework or generative model to obtain the representation of unlabeled data. However, these semi-supervised models can not be applied to generate labels for our raw QA pairs since our task requires diverse types of labels from discourse to sentence which are difficult to annotate, and it is more serious that the entire unlabeled data is also too small to support an effective semi-supervised learning. Thus, we must very carefully choose a proper learning strategy for our task.

Different from the above research line, one-shot learning belongs to a kind of weakly supervised method (Bordes et al., 2014) that was first proposed to learn information about object categories from one or only a few training images (Li et al., 2006). Recently, memory-based neural network models have greatly expanded the ways of storing and accessing information. Two promising neural networks with external memory have been proposed to connect deep neural network with one-shot learning. One is Neural Turing Machine (Graves et al., 2014), which can read or write the facts to an external differentiable memory. Santoro et al. (2016) proved it to be an effective method for image recognition. The other is Memory Network (Vinyals et al., 2016). The crucial difference between them is that the latter does not have a mechanism to modify the content of the external memory, which lets us choose the former as our one-shot learning implementation. Compared with previous methods for question answering, our model is more weakly supervised with a limited amount of training data. As to our best knowledge, this is the first attempt that adopts one-shot learning for a deep QA task by using NTM as automatic labeler.

## 5 Conclusion

This paper presents a neural model with NTM for a challenging deep question answering task from history examinations. Our experimental results show that the adopted CGNN together with other neural models works much better than BM25. With an NTM labeler, all the deep neural models are further enhanced. We also release a Chinese comprehensive deep question answering dataset and have launched a new research line for the Gaokao challenge and solve more complicated questions using deep neural networks to learn semantic representation while the previous work focused on choice questions using simple information retrieval methods (Cheng et al., 2016). In fact, open-domain questions that can often be distinguished into different aspects can always benefit from one-shot learnings over a few labeling samples, which has been verified the effectiveness in this paper by only relying on the least task-dependent assumptions.

## References

- Gabor Angeli, Melvin Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *ACL*, pages 344–354.
- Antoine Bordes, Jason Weston, and Nicolas Usunier. 2014. Open question answering with weakly supervised embedding models. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 165–180.
- Jiaxun Cai, Shexia He, Zuchao Li, and Hai Zhao. 2018. A full end-to-end semantic role labeler, syntactic-agnostic or syntactic-aware? In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*.
- Jifan Chen, Qi Zhang, Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016a. Implicit discourse relation detection via a deep architecture with gated relevance network. In *ACL*, pages 1726–1735.
- Qian Chen, Xiaodan Zhu, Zhenhua Ling, Si Wei, and Hui Jiang. 2016b. Enhancing and combining sequential and tree lstm for natural language inference. *arXiv preprint arXiv:1609.06038*.
- Gong Cheng, Weixi Zhu, Ziwei Wang, Jianghui Chen, and Yuzhong Qu. 2016. Taking up the gaokao challenge: An information retrieval approach. *IJCAI*, pages 2479–2485.
- Peter Clark. 2015. Elementary school science and math tests as a driver for ai: Take the aristo challenge! In *IAAI*, pages 4019–4021.
- Zihang Dai, Lei Li, and Wei Xu. 2016. CFO: Conditional focused neural question answering with large-scale knowledge bases. *arXiv preprint arXiv:1606.01994*.
- John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(39):2121–2159.
- Minwei Feng, Bing Xiang, Michael R. Glass, Lidan Wang, and Bowen Zhou. 2015. Applying deep learning to answer selection: A study and an open task. In *ASRU*, pages 813–820.
- David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. 2010. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79.
- Akira Fujita, Akihiro Kameda, Ai Kawazoe, and Yusuke Miyao. 2014. Overview of today robot project and evaluation framework of its nlp-based problem solving. *ICLR*, pages 2590–2597.
- Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv preprint arXiv:1410.5401*.
- Shangmin Guo, Xiangrong Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2017. Which is the effective way for gaokao: Information retrieval or neural networks? In *EACL*, pages 111–120.
- Shexia He, Zuchao Li, Hai Zhao, Hongxiao Bai, and Gongshen Liu. 2018. Syntax for semantic role labeling, to be, or not to be. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*.
- Mikael Henaff, Jason Weston, Arthur Szlam, Antoine Bordes, and Yann Lecun. 2016. Tracking the world state with recurrent entity networks. *arXiv preprint arXiv:1612.03969*.
- Yafang Huang, Zuchao Li, Zhuosheng Zhang, and Hai Zhao. 2018. Moon IME: neural-based chinese pinyin aided input method with customizable association. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018), System Demonstration*.
- Daniel Khashabi, Tushar Khot, Ashish Sabharwal, Peter Clark, Oren Etzioni, and Dan Roth. 2016. Question answering via integer programming over semi-structured knowledge. In *IJCAI*, pages 1145–1152.
- Tushar Khot, Ashish Sabharwal, and Peter Clark. 2017. Answering complex questions using open information extraction. *ACL*, pages 311–316.
- Souvik Kundu and Hwee Tou Ng. 2018. A question-focused multi-factor attention network for question answering. *arXiv preprint arXiv:1801.08290*.
- Tao Lei, Hrishikesh Joshi, Regina Barzilay, Tommi Jaakkola, Katerina Tymoshenko, Alessandro Moschitti, and Lluís Marquez. 2016. Semi-supervised question retrieval with gated convolutions. In *NAACL*, pages 1279–1289.

- Fei-Fei Li, R. Fergus, and P. Perona. 2006. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611.
- Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. 2018. Seq2seq dependency parsing. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*.
- Hanqing Lu and Ming Kong. 2017. Community-based question answering via contextual ranking metric network learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 4963–4964.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Lianhui Qin, Zhisong Zhang, Hai Zhao, Zhiting Hu, and Eric P. Xing. 2017. Adversarial connective-exploiting networks for implicit discourse relation classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 1006–1017.
- Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *EMNLP 2016*, pages 2383–2392.
- Spyridon Samothrakis, Tom Vodopivec, Michael Fairbank, and Maria Fasli. 2017. Convolutional-match networks for question answering. In *IJCAI*, pages 2686–2692.
- Adam Santoro, Sergey Bartunov, Matthew Botvinick, Daan Wierstra, and Timothy Lillicrap. 2016. One-shot learning with memory-augmented neural networks. In *ICML*, pages 1842–1850.
- Rupesh Kumar Srivastava, Klaus Greff, and Jürgen Schmidhuber. 2015. Highway networks. *arXiv preprint arXiv:1505.00387*.
- Ming Tan, Cicero dos Santos, Bing Xiang, and Bowen Zhou. 2015. LSTM-based deep learning models for non-factoid answer selection. *arXiv preprint arXiv:1511.04108*.
- Mihaylov Todor and Frank Anette. 2018. Enhancing cloze-style reading comprehension with external common knowledge using explicit key-value memory. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (ACL 2018)*.
- Armando Vieira. 2016. Knowledge representation in graphs using convolutional neural networks. *arXiv preprint arXiv:1612.02255*.
- Oriol Vinyals, Charles Blundell, Tim Lillicrap, Daan Wierstra, et al. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, pages 3630–3638.
- Zhenghao Wang, Shengquan Yan, Huaming Wang, and Xuedong Huang. 2014. An overview of microsoft deep qa system on stanford webquestions benchmark. Technical report, Technical report, Microsoft Research.
- Rui Wang, Hai Zhao, Bao Liang Lu, Masao Utiyama, and Eiichiro Sumita. 2015. Bilingual continuous-space language model growing for statistical machine translation. *IEEE/ACM Transactions on Audio Speech & Language Processing*, 23(7):1209–1220.
- Wenhui Wang, Nan Yang, Furu Wei, Baobao Chang, and Ming Zhou. 2017a. Gated self-matching networks for reading comprehension and question answering. In *ACL*, pages 189–198.
- Zhiguo Wang, Wael Hamza, and Radu Florian. 2017b. Bilateral multi-perspective matching for natural language sentences. *arXiv preprint arXiv:1702.03814*.
- Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. *arXiv*, 1603.
- Min Chul Yang, Nan Duan, Ming Zhou, and Hae Chang Rim. 2014. Joint relational embeddings for knowledge-based question answering. In *EMNLP*, pages 645–650.
- Zhilin Yang, Junjie Hu, Ruslan Salakhutdinov, and William W Cohen. 2017. Semi-supervised qa with generative domain-adaptive nets. *arXiv preprint arXiv:1702.02206*.
- Wen Tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *ACL*, pages 643–648.

- Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2016. Neural generative question answering. In *IJCAI*, pages 2972–2978.
- Kai Zhang, Wei Wu, Fang Wang, Ming Zhou, and Zhoujun Li. 2016a. Learning distributed representations of data in community question answering for question retrieval. In *ACM*, pages 533–542.
- Zhisong Zhang, Hai Zhao, and Lianhui Qin. 2016b. Probabilistic graph-based dependency parsing with convolutional neural network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1382–1392.
- Xiaodong Zhang, Sujian Li, Lei Sha, and Houfeng Wang. 2017. Attentive interactive neural networks for answer selection in community question answering. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 3525–3531.
- Zhuosheng Zhang, Yafang Huang, and Hai Zhao. 2018a. Subword-augmented embedding for cloze reading comprehension. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*.
- Zhuosheng Zhang, Jiangtong Li, Hai Zhao, and Bingjie Tang. 2018b. Sjtunlp at semeval-2018 task 9: Neural hypernym discovery with term embeddings. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval 2018), Workshop of NAACL-HLT 2018*.
- Zhuosheng Zhang, Jiangtong Li, Pengfei Zhu, and Hai Zhao. 2018c. Modeling multi-turn conversation with deep utterance aggregation. In *Proceedings of the 27th International Conference on Computational Linguistics (COLING 2018)*.
- Hai Zhao, Chang-Ning Huang, Mu Li, and Taku Kudo. 2006. An improved Chinese word segmentation system with conditional random field. *Proceedings of the Fifth Sighan Workshop on Chinese Language Processing*, pages 162–165.