ALS at IJCNLP-2017 Task 5: Answer Localization System for Multi-Choice Question Answering in Exams

Changliang Li[1], **Cunliang Kong**[1,2]
[1]Institute of Automation, Chinese Academy of Sciences
[2]Beijing Language and Culture University
changliang.li@ia.ac.cn, 201621198311@stu.blcu.edu.cn

Abstract

Multi-choice question answering in exams is a typical QA task. To accomplish this task, we present an answer localization method to locate answers shown in web pages, considering structural information and semantic information both. Using this method as basis, we analyze sentences and paragraphs appeared on web pages to get predictions. With this answer localization system, we get effective results on both validation dataset and test dataset.

1 Introduction

Multi-Choice Question Answering in Examinations is the 5th shared task in IJCNLP-2017, which aims to test how accurately system built by participants could answer the questions in exams. The dataset contains multiple choice questions from science and history curriculum, and is comprised of English part and Chinese part. In this work, we focus on the Chinese part.

We found that there are many web pages containing answers of the questions in dataset. To accomplish the task, we crawled these web pages and analyze answers appeared on them. When analyzing these pages, we need to compare sentences in the original dataset and sentences on web pages. There are many sentence pairs have the same meaning but with different forms. To process these sentences pairs, we use a localization method to locate the positions of sentences in web pages. We use edit distance of sentence pairs to represent structural similarities, and use cosine score of two vectors represented by a convolutional neural network (CNN) to represent semantic similarities. With merging these two scores together, we locate choices appeared on web pages.

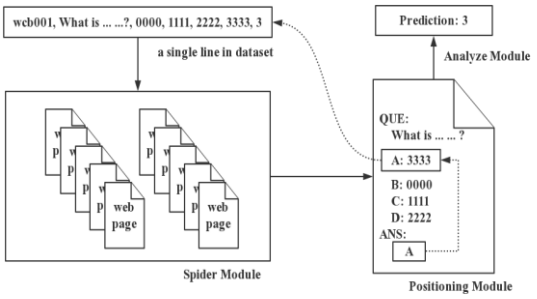

Figure 1: Overview of our system. Communication between modules is indicated by arrows.

Finally, we use the analyzed answer to find out the right choice.

This system can solve multi-choice question answering in exams as long as there are relevant web pages. The answer localization method used in the system can provide high robustness. The system is applicable to a variety of situations, even when choices on the web pages show different orders and different forms with choices in the original dataset.

The final accuracy score of our system on the test dataset is 58%, and on the validation dataset is 60%, while the baseline given by the organizer is 44.63%.

2 System Description

The system is comprised of three modules: spider module, positioning module and analysis module. Full system is as shown in figure 1. The following will describe these three parts separately.

Proceedings of the 8th International Joint Conference on Natural Language Processing, Shared Tasks, pages 199–202,
Taipei, Taiwan, November 27 – December 1, 2017.

2.1 Spider Module

To search related resources on the Internet, for each line in the dataset, we concatenate the question and choices as a query. After that, we use Baidu Search Engine to search relevant web pages with the query. We analyze and save web pages that relevant to each line as a single file with JavaScript Object Notation (JSON) format.

For line l in dataset, we have a question sentence q, choices c_i ($0 \leq i \leq 3$), and an answer a. And there are several web pages related to l, each of them is saved into two fields: *QUE* and *ANS*. Filed *QUE* contains the question and choices, and field *ANS* contains the answer shown on the web page.

2.2 Positioning Module

The positioning module is to select the fittest web page related to l, and position each choice shown on the web page.

Our target is to analyze predictions from field *ANS* of web pages. But the order of choices shown on web pages might be different from the order of choices shown in original dataset (as is shown in figure 1). Moreover, choices shown on web pages might have different forms, while they have the same meaning. There are various cases, we can cite some cases here as examples.

(1) Increasing or reducing words. The phrase *生物克隆* and the phrase *生物克隆技术* have the same meaning of *cloning technology*, but the latter has two more words than the former.

(2) Changing of word order. The phrase *种子的有无* means *have or not have seeds*, while the phrase *有无种子* has the same meaning.

(3) Synonyms. The phrase *产生二氧化碳* and the phrase *产生* CO_2 have the same meaning of *producing carbon dioxide* while *二氧化碳* and CO_2 are synonyms.

As we can see, it is necessary to solve the problem of orders and multiple forms. We use following steps to solve these problems.

Step 1, select the fittest web page. We use two strategies to choose the best one from several web pages.

(1) Field *ANS* should contains at least one character in 'A, B, C, D'.

(2) Field *QUE* should have the minimum edit distance ratio with the original question. The edit distance ratio of two strings a and b is defined as below.

$$r_e(a,b) = 1 - \frac{ED(a,b)}{\text{maxlen}\,(a,b)} \quad (1)$$

where r_e is edit distance ratio, *ED* stands for edit distance.

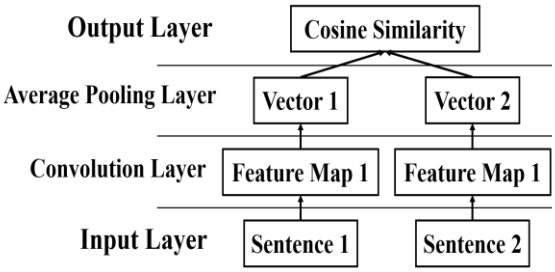

Figure 2: Convolutional architecture we used for computing semantic similarities.

Step 2, positioning each choice in field *QUE*.

(1) Computing structural similarities.

For c_i as the i^{th}choice, we have a window size o_i defined as following:

$$o_i = len(c_i) + size \quad (2)$$

where *size* is an hyperparameter tuned using training dataset.

Getting o_i words from field *QUE* as a string s_t, where the first word of s_t is the t^{th} word of field *QUE*, we use edit distance ratio to compute the structural similarity of c_i and s_t. *i.e.*:

$$sim^f_{i,t} = r_e(c_i, s_t) \quad (3)$$

(2) Computing semantic similarities.

We use CNNs to represent the semantic information of c_i and s_t as two vectors v^c_i and v^s_t respectively. And we compute cosine score $sim^s_{i,t}$ of v^c_i and v^s_t as the semantic similarity of c_i and s_t.

In our implementation, we pad sentences to have the same length z, which is the max length of these sentences. As is shown in figure 2, there are four layers in the CNN model: input layer, convolution layer, average pooling layer and output layer. We now describe each in turn.

Input Layer. In each sentence, each word is represented as a d_0-dimensional precomputed word2vec (Mikolov et al. 2013) embedding. In this work, we set $d_0 = 30$. In this way, each sen-

tence is represented as a matrix of dimension $d_0 \times z$.

Convolution layer. Let ω be the filter width, and $(v_0, v_1, \ldots, v_s)$ be the words of a sentence. We concatenate embeddings of $(v_{k-\omega+1}, \ldots, v_k)$ to be $c_k \in \mathbb{R}^{\omega \cdot d_0}$ $(\omega < k < z)$. Then, we generate the representation $p_k \in \mathbb{R}^{d_1}$ for c_k using convolution weights $\mathrm{W} \in \mathbb{R}^{d_1 \times \omega d_0}$ as follows:

$$p_k = f(\mathrm{W} \cdot c_k + \mathrm{b}) \quad (4)$$

where f is the activation function, $\mathrm{b} \in \mathbb{R}^{d_1}$ is the bias.

Average pooling layer. There are several kinds of pooling commonly used to extract robust features from convolution, such as min pooling, max pooling and average pooling. In this work, we found average pooling showing the best result. This layer generates a pair of representation vectors for each of the two sentences. These two representations are then the basis for similarity computation.

Output layer. The last layer returns the output. In our work, we use cosine similarity of two representation vectors mentioned above as the output. i.e.:

$$sim_{i,t}^{s} = \frac{v_i^c \cdot v_t^s}{\|v_i^c\| \, \|v_t^s\|} \quad (5)$$

(3) Merging.

Merging structural similarity $sim_{i,t}^{f}$ and semantic similarity $sim_{i,t}^{s}$ together, we get the final similarity $sim_{i,t}$ of c_i and s_t.

$$sim_{i,t} = \alpha sim_{i,t}^{f} + (1 - \alpha) sim_{i,t}^{s} \quad (6)$$

where α is an hyperparameter tuned using train dataset.

To get the position of c_i in field *QUE*, we select the s_t that has biggest similarity with c_i. *i.e.*:

$$t^* = \mathrm{argmax}(sim_{i,t}) \quad (7)$$

Finally, we get a tuple (i, t^*) that contains the order of a choice and its index in field *QUE*. Since there are four choices, we get a list L of four tuples.

Step 3, sorting the list and get the right order.

We sort the list L by the second element of tuples, and we assume the order after sorting is the order of choices appeared on the web page. i.e., the first tuple is the choice *A*, the second tuple is the choice *B*, and so on.

2.3 Analysis Module

In this module, we analyze the field *ANS* in saved web pages. In this way, we get the answer given by the web page such as *A*, *B*, *C* or *D*.

For ease of calculation, we use *0, 1, 2, 3* instead of *A, B, C, D*, so we get the result n $(0 \leq n \leq 3)$.

To get the final result, we select the first element of the n^{th} tuple in sorted list L as a^*. And a^* is the predicted answer of q.

3 Related Work

We got a lot of inspiration from others' work, they've given many shoulders on which this paper is standing.

Question answering has attracted lots of attention in recent years. Sukhbaatar et al. (2015) introduced a neural network with a recurrent attention model over a possibly large external memory, which is called end-to-end memory networks. After that, Kumar et al. (2015) introduced the dynamic memory network (DMN) which processes input sequences and questions, forms episodic memories, and generates relevant answers. Based on DMN, Xiong et al. (2016) proposed several improvements for memory and input modules, and introduced a novel input module for images in order to be able to answer visual questions. With rapid growth of knowledge bases (KBs) on the web and the development of neural network based (NN-based) methods, NN-based KB-QA has already achieved impressive results. Zhang et al. (2016) presented a neural attention-based model to represent the question with dynamic attention. Wang et al. (2016) have done a lot of valuable exploration of different attention methods in recurrent neural network (RNN) models.

Convolutional neural networks (CNNs) has been widely used in NLP fields in recent years, and yield effective results. Kalchbrenner et al. (2014) introduced a convolutional architecture dubbed the Dynamic Convolutional Neural Network (DCNN) for the semantic modeling of sentences. Kim (2014) used CNN for sentence classification, and achieved excellent results on multiple benchmarks. Yin et al. (2015) presented a general Attention Based Convolutional Neural Network (ABCNN) for modeling a pair of sentences, which can be applied to a wide variety of tasks. Hu et al. (2015) used CNN architectures for matching natural language sentences.

4 Experiment

The dataset we used is given by the organizer of IJCNLP-2017, shared task 5. The dataset totally contains 9080 lines in Chinese and is randomly divided into train, validation and test datasets, each line has the form like:

id, "q", "c0", "c1", "c2", "c3", a

where *id* is the unique integer id for each question, *q* is the question text, *c0, c1, c2, c3* is four choices respectively, and *a* is the correct answer which is only available for train dataset.

The dataset contains two subjects: biology and history. And detail statistics is showed as table 1.

	Train	Validation	Test	Total
Biology	2266	566	1699	4531
History	2275	568	1706	4549
Total	4541	1134	3405	9080

Table 1: Detail statistics of dataset.

We only use edit distance at first to collect choices on web pages with their similar choices in dataset as sentence pairs marked label '1'. Then we collect the same amount of sentence pairs which are not similar marked label '0'. With these sentence pairs, we set filter width ω in the convolutional architecture to be 3, and uses mean squared error loss function to train the CNN model.

After trained the convolutional architecture, we set hyperparameter *size* in Eq. 2 as 5, α in Eq. 6 as 0.95. And the result of our system on test dataset is as shown in figure 3. As we can see, the answer localization system performed well on both test dataset and validation dataset, while the accuracy on test dataset is 58%, on validation dataset is 60.01%.

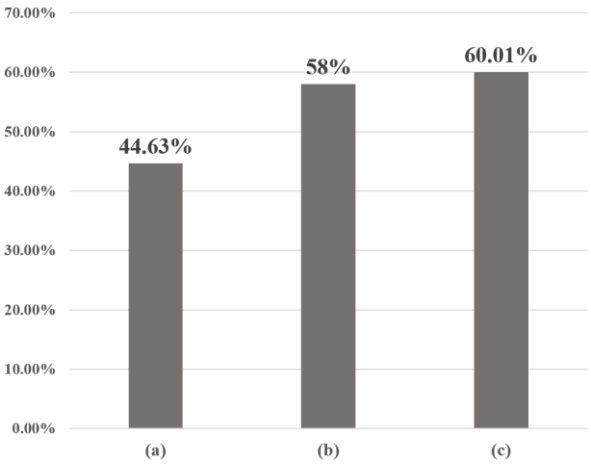

Figure 3: Comparing of system results. (a): The retrieval based method baseline. (b): The answer localization system on test dataset. (c): The answer localization system on validation dataset.

5 Conclusion

This system uses an answer localization method of merging structural information and semantic information together, and uses this information to locate the correct answer appeared on web pages. The final results proved the effectiveness of the proposed method.

References

Yin, W., Schütze, H., Xiang, B., & Zhou, B. (2015). Abcnn: attention-based convolutional neural network for modeling sentence pairs. *Computer Science*.

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A convolutional neural network for modelling sentences. *Eprint Arxiv, 1*.

Hu, B., Lu, Z., Li, H., & Chen, Q. (2015). Convolutional neural network architectures for matching natural language sentences. , *3*, 2042-2050.

Kim, Y. (2014). Convolutional neural networks for sentence classification. *Eprint Arxiv*.

Sukhbaatar, S., Szlam, A., Weston, J., & Fergus, R. (2015). End-to-end memory networks. *Computer Science*.

Kumar, A., Irsoy, O., Ondruska, P., Iyyer, M., Bradbury, J., & Gulrajani, I., et al. (2015). Ask me anything: dynamic memory networks for natural language processing. 1378-1387.

Xiong, C., Merity, S., & Socher, R. (2016). Dynamic memory networks for visual and textual question answering.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems,26*, 3111-3119.

Wang, B., Liu, K., & Zhao, J. (2016). Inner Attention based Recurrent Neural Networks for Answer Selection. *Meeting of the Association for Computational Linguistics* (pp.1288-1297).

Zhang, Y., Liu, K., He, S., Ji, G., Liu, Z., & Wu, H., et al. (2016). Question answering over knowledge base with neural attention combining global knowledge information.