YNUDLG at IJCNLP-2017 Task 5: A CNN-LSTM Model with Attention for Multi-choice Question Answering in Examinations

Min Wang, Qingxun Liu, Peng Ding, Yongbin Li, Xiaobing Zhou*
School of Information Science and Engineering,
Yunnan University, Yunnan, P.R. China
*Corresponding author, zhouxb.cn@gmail.com

Abstract

"Multi-choice Question Answering in Exams" is a typical question answering task, which aims to test how accurately the participants could answer the questions in exams. Most of the existing QA systems typically rely on handcrafted features and rules to conduct question understanding and/or answer ranking. In this paper, we perform convolutional neural networks (CNN) to learn the joint representations of question-answer pairs first, then use the joint representations as the inputs of the long short-term memory (LSTM) with attention to learn the answer sequence of a question for labeling the matching quality of each answer. All questions are restrained within the elementary and middle school level. We also incorporating external knowledge by training Word2Vec on Flashcards data, thus we get more compact embedding. Experimental results show that our method achieves better or comparable performance compared with the baseline system. The proposed approach achieves the accuracy of 0.39, 0.42 in English valid set, test set, respectively.

1 Introduction

Multi-choice question answering systems return the correct answer from four candidates to natural language questions. In recent years, a typical method is to model Question-Answer pairs and classify them (Severyn et al., 2016). The nature of this way is to transform QA problem to classification problem. Some people tackle this directly by computing the cosine distance between question and answer (Feng et al., 2015; Santos et al., 2016). Besides, the development of largescale knowledge bases, such as FREEBASE (Bollacker et al., 2008), provides a rich resource to answer open domain questions. With the generative adversarial nets (Goodfellow, et al., 2014) emerging, it achieves higher performance in NLP or NLU. Neural generative question answering (Yin et al., 2015) is built on the encoder-decoder framework for sequence-to-sequence learning, and the architecture of this system holds the ability to enquire the knowledge-base, and is trained on a corpus of question-answer pairs.

Up to now, there are three mainstream methods for question answering tasks. The first one is based on sentiment contained in question-answer pairs (Zhou et al., 2015). The sentiment method learns to understand natural language questions by converting them into classification problem, 0/1 respectively represents negative or positive sentiment of the Q-A pairs, i.e., wrong answer or correct answer. The second approach uses information extraction techniques for open question answering (Yin et al., 2015; Yao and Van Durme, 2014; Bordes et al., 2014a; Bordes et al., 2014b; Yang et al,. 2015). This method retrieves a set of candidate answers from the knowledge-base, and then extract features from the question and their candidates to rank them. Yin (2015) enquires candidate answers from knowledge-base organized in the form of (subject, predicate, object) and then ranks the similarity between question and candidate answers, finally put out the answer sentence by a generator with

Proceedings of the 8th International Joint Conference on Natural Language Processing, Shared Tasks, pages 194–198,
Taipei, Taiwan, November 27 – December 1, 2017.

attention. And, the method proposed by Yao and Van Durme (2014) relies on rules and dependency parse results to extract handcrafted features for questions. Moreover, some methods (Bordes et al., 2014a; Bordes et al., 2014b) use the summation of question word embeddings to represent questions, which ignores word order information and cannot process complicated questions. The third way is correspondingly easy and directly computes the cosine distance between question and answer. Question and answer vectors are put into a hidden layer and then go through a CNN layer and maxpooling layer, finally computes the cosine distance between question and answer. This way doesn't use pre-trained word embeddings and can't accurately represent the relationship among words. Based on the above analyses, in this paper, we combine the first and second ways to model and train on large question answer datasets. Specifically, we transform this issue to classify question answer pairs negative or positive to judge the answer wrong or correct. Similar to the second way, we train word embeddings from external knowledge-base (KB), which creates a small, more compact embedding. Unlike some work or the baseline described retrieving related text from KB according to question or answer query, our system trains word embeddings on external KB and use CNN and LSTM model with attention mechanism to classify Q-A pairs. The model shares the same word embeddings trained by *word2vec*.

2 Data

We consider the question answer problem in Multi-choice as a sequence labeling task. All questions are restrained within the elementary and middle school level. The subjects of English subset contain biology, chemistry, physics, earth science and life science. We collect many question answer data based on the subjects including these five categories from Allen Institute for Artificial Intelligence (http://allenai.org/data.ht). Summary statistics of the datasets are listed in table 1.

Data	Num of Questions
SciQ	13,679
AI2	1,459
TQA	13,693
Aristo	20k
SS	46k
TrainSet	2,686
ValidSet	669

Table 1: Summary statistics for the datasets: Aristo and SS are used to train word embedding and the other used to train model. Number of questions stands for how many questions the dataset contains, each question has four candidate answers.

- **AI2 Science Questions v2**: 5,059 real science exam questions derived from a variety of regional and state science exams. The AI2 Science Questions dataset consists of questions used in student assessments in the United States across elementary and middle school grade levels. Each question is 4-way multiple choice format and may or may not include a diagram element.

- **Textbook Question Answering**: 1,076 textbook lessons, 26,260 questions, 6,229 images. Each lesson has a set of multiple choice questions that address concepts taught in that lesson. TQA has a total of 26,260 questions, in which 12,567 have accompanying diagrams. We just use the questions without diagrams from this dataset.

- **SciQ dataset**: 13,679 science questions with supporting sentences. The SciQ dataset contains 13,679 crowdsourced science exam questions about Physics, Chemistry, Biology, and other subjects. The questions are in multiple-choice format with 4 answer options each. For the majority of the questions, an additional paragraph with supporting evidence for the correct answer is provided.

- **Aristo MINI Corpus**: The Aristo Mini corpus contains 1,197,377 (very loosely science-relevant sentences drawn from public data). It provides simple science-relevant text that may be useful to help answer elementary science questions.

- **StudyStack Flashcards**: This content source is from StudyStack and manually organized in the form of question answer pairs. This gives us 400k flashcard records, questions followed by the correct answer. We use this and Aristo corpus to train the word embeddings.

3 System

This section explains the architecture of our deep learning model for modeling question-answer pairs and then to classify them. The system

combines CNN with LSTM network, and the attention mechanism is also taken into account. First we put question and answer pairs into CNN network, and get question answer joint representations, then we input them into LSTM network, next we merge question and answer vectors by dot mode. Finally the softmax layer is applied to classify the joint representations.

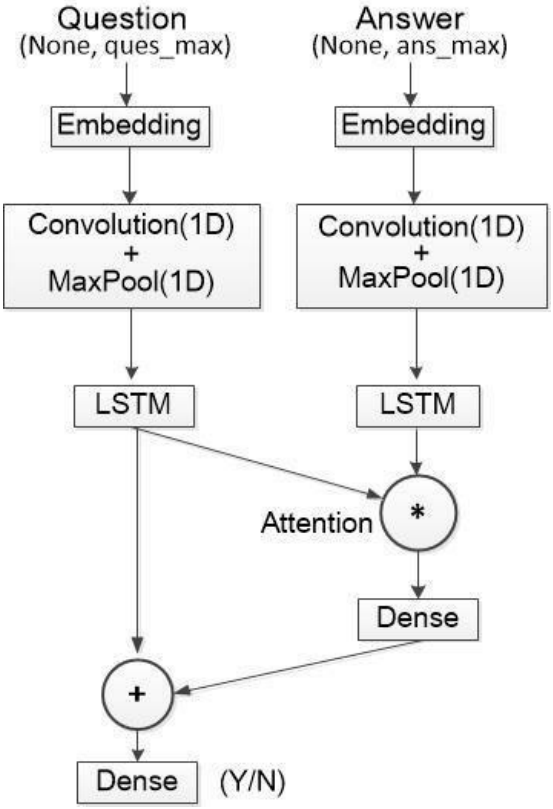

Figure 1: Our deep learning architecture for classifying question-answer pairs. The attention is developed by merging two LSTM output layers with the 'dot' mode.

3.1 CNN for QA Joint learning

The architecture of our CNN network for mapping sentences to feature vectors is shown in Fig.1. It is mainly inspired by the convolutional architectures used in (Blunsom et al., 2014; Kim, 2014; Aliaksei et al., 2016) for performing different sentence classification tasks. Different from the previous work, the goal of our distributional sentence model is to learn intermediate representations of questions and answers used to classify them into negative or positive. We use the following parameters: word embedding dimension is 300, sentence length is 64, kernel size is 5, number of filters is 32. In our model the input shape of the sentence (question or answer) is (None, 64), input this into embedding layer the shape changes to (None, 64, 300), next through convolution computing, we get sentences shape (None, 60, 32). After pooling, we get (None, 30, 30) sentences vectors.

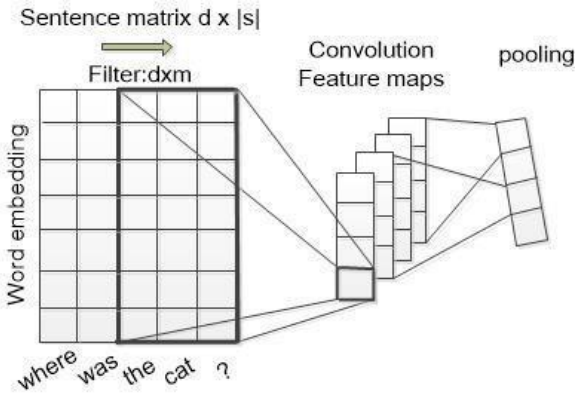

Figure 2: Our sentence model for mapping input sentences to their intermediate representations.

3.2 Combining LSTM and CNN for Ordinary Classification

Based on the joint representation of QA pairs, the LSTM layer of our model performs answer sequence learning to model semantic links between continuous answers. In Fig.1, unlike a conventional LSTM model which directly uses word embeddings as input, the proposed model takes outputs from a single layer CNN with maxpooling.

Due to the gradient vanishing problem, conventional RNNs are found difficult to be trained to exploit long-range dependencies. In order to mitigate this weak point in conventional RNNs, specially designed activation functions have been introduced. LSTM is one of the earliest attempts and still a popular option to tackle this problem. In the LSTM architecture, there are three gates (input *i*, forget *f* and output *o*), and a cell memory activation vector *c*. The vector formulas for recurrent hidden layer function *H* in this version of LSTM network are implemented as follows:

$$
\begin{aligned}
i_t &= \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right) \\
f_t &= \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_f\right) \\
c_t &= f_t c_{t-1} + i_t \tau\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \\
o_t &= \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + b_o\right) \\
h_t &= o_t \theta\left(c_t\right)
\end{aligned}
$$

where τ and θ are the cell input and cell output non-linear activation functions which are stated as *tanh* in this paper.

3.3 Attention mechanism

Problem with RNNs in general is the vanishing gradient problem. While LSTMs address the problem, in QA contexts they can't characterize interaction between question and answer. The solution to this is attention mechanism, where the network is forced to look at certain parts of the context and ignore (in a relative sense) everything else. We just adopt a relatively easy way to accomplish this task.

The Keras merge layer provides a series of layer objects and methods for fusing two or more tensors. The Merge layer supports some predefined merge patterns, including sum, concat, mul, dot et al. Mode mul is to deal with the combined layer output to do a simple multiplication operation. Unlike mul, mode dot is used to tensor multiplication. One can use the dot_axis keyword parameter to specify the axis to be eliminated. For example, if the shapes of two tensors a and b are (batch_size,n), the output of dot is a tensor such as (batch_size, 1). We consider this as the attention vector in question and answer.

4 Experiment and Result analysis

We only participate in the English task, and this challenge employs the accuracy of a method on answering questions in test set as the metric, the accuracy is calculated as:

$$Accuracy = \frac{\text{number of correct questions}}{\text{total number of questions}}$$

As indicated in Table 2, the CNN-LSTM with attention model shows better experiment result than single CNN or LSTM. LSTM performs better than CNN, one reason may be when implementing convolution operation, there are much zeroes in sentences matrix after padding. Obviously, when adding attention mechanism, CNN and LSTM can obtain much improvements. In our experiment, we test two layer LSTMs with attention and CNN-LSTM with attention, this two models perform almost the same on valid set and test set, respectively. Finally, we choose the CNN-LSTM with attention model.

Model	Valid Acc	Test Acc
CNN	0.277	0.260
LSTM	0.286	0.292
CNN+Attention	0.301	0.283
LSTM+Attention	0.343	0.305
CNN-LSTM+Attention	0.396	0.422

Table 2: Overview of our result on the English Multi-choice question answering subset in examinations.

5 Conclusion and Future Work

In this paper, we present a question answer learning model, CNN-LSTM with attention for Multi-choice in examinations. This transformation from QA into classification problem is easy and clear. True question answer is complex, there is a need to take into account other features such similarity overlap words between questions and answers. The experiments provide strong evidence that distributed and joint representations are feasible in tackling QA problem. Experiment results demonstrate that our approach can learn the useful context from answering to improve the performance of Multi-choice question answer in exams, compared to baseline model. One of the reasons why our model performs well may be our training data is large and highly correlate to the valid set and test set.

In the future, we plan to explore the method using KB or other neural networks like generative adversarial networks (GAN), variational autoencoder (VAE) to model sentences and perform other NLP tasks.

Acknowledgments

This work was supported by the Natural Science Foundations of China under Grants No.61463050, No.61702443, No.61762091, the NSF of Yunnan Province under Grant No. 2015FB113, the Project of Innovative Research Team of Yunnan Province.

References

Severyn A, Moschitti A. 2016. Modeling Relational Information in Question-Answer Pairs with Convolutional Neural Networks. arXiv preprint arXiv:1604.01178.

Feng M, Xiang B, Glass M R, et al. 2015. Applying deep learning to answer selection: A study and an open task. In *IEEE Workshop on Automatic Speech Recognition and Understanding*, Pages 813-820.

Santos C D, Tan M, Xiang B, et al. 2016. Attentive Pooling Networks. arXiv preprint arXiv: 1602.03609.

Bollacker, K, Evans, C, Paritosh, P, et al. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *International Conference on Management of Data*, pages 1247-1250.

Goodfellow I, Pouget-Abadie J, Mirza M, et al. 2014.Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, pages 2672-2680.

Yin J, Jiang X, Lu Z, et al. 2016. Neural Generative Question Answering. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pages 2972-2978.

Zhou X, Hu B, Chen Q, et al. 2015. Answer Sequence Learning with Neural Networks for AnswerSelection in Community Question Answering. arXiv preprint arXiv:1506.06490.

Yao X, Van Durme B. 2014. Information extraction over structured data: Question answering with freebase. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 956-966. Association for Computational Linguistics.

Yao X, Berant J, Van Durme B. 2014. QA: Information Extraction or Semantic Parsing? In *Proceedings of the ACL 2014 Workshop on Semantic Parsing*, pages 82-86. Association for Computational Linguistics.

Bordes A, Chopra S, Weston J. 2014a. Question answering with subgraph embeddings. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 615-620. Association for Computational Linguistics.

Bordes A, Weston J, Usunier N. 2014b. Open question answering with weakly supervised embedding models. In *Machine Learning and Knowledge Discovery in Databases—European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part I*, pages 165–180.

Yang Y, Yih W, Meek C. WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2013–2018.

Huang J, Zhou M, Yang D. 2007. Extracting chatbot knowledge from online discussion forums. In *Proceedings of the 20th international joint conference on artificial intelligence*, pages 423-428.

Ding S, Cong G, Lin C Y, et al. 2008. Using Conditional Random Fields to Extract Contexts and Answers of Questions from Online Forums. In *Proceedings of the Meeting of the Association for Computational Linguistics*, pages 710-718.

Wang B, Wang X, Sun C, et al. 2010. Modeling semantic relevance for question-answer pairs in web social communities. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1230-1238.

Yu L, Hermann K M, Blunsom P, et al. 2014. Deep Learning for Answer Sentence Selection. arXiv preprint arXiv:1412.1632.

Echihabi A, Marcu D. 2003. A noisy-channel approach to question answering. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 16-23. Association for Computational Linguistics.

Yang M C, Duan N, Zhou M, et al. 2014. Joint Relational Embeddings for Knowledge-based Question Answering. In *Conference on Empirical Methods in Natural Language Processing*, page 645-650. Association for Computational Linguistics.

Blunsom P, Grefenstette E, Kalchbrenner N. 2014. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 655-665.

Kim Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.