Two layers LSTM with attention for multi-choice question answering in exams

Yongbin Li[1.2]

[1]ZunYi Medical University, No.6 Xuefu West Road, Xinpu new distract, Zunyi, GuiZhou, CHINA
[2]YunNan University, University City, Chenggong distract, Kunming, YunNan, CHINA

*Corresponding author's e-mail: 1957405@qq.com

Abstract. Question Answering in Exams is typical question answering task that aims to test how accurately the model could answer the questions in exams. In this paper, we use general deep learning model to solve the multi-choice question answering task. Our approach is to build distributed word embedding of question and answers instead of manually extracting features or linguistic tools, meanwhile, for improving the accuracy, the external corpus is introduced. The framework uses a two layers LSTM with attention which get a significant result. By contrast, we introduce the simple long short-term memory (QA-LSTM) model and QA-LSTM-CNN model and QA-LSTM with attention model as the reference. Experiment demonstrate superior performance of two layers LSTM with attention compared to other models in question answering task.

1. Introduction

Question Answering in Exams is typical question answering task can be formulated as follows: Given a question q and an answer candidate pool { a1 , a2 , ··· , as } for the question (s is the pool size, in this task is 4), we aim to find the best answer candidate ak , 1 ≤ k ≤s. Question and answer are token sequence with arbitrary length, and a question can correspond to multiple ground-truth answers, and correct answer may only have semantically relation with question instead of directly sharing lexical units. The selected answer ak is inside the ground truth set or outside, from the definition, the Question Answering task can be transform as a binary classification problem. For each question, for each answer candidate, we construct a QA pair, it may be appropriate or not, we need to measure the matching degree, the highest will be chosen.

The above is general definition. The task is come from a NLP competition, we employ multiple choice questions from a typical science and history curriculum, questions are restrained within the elementary and middle school level. For each question, there are four possible answers, where each of them may be a word, a value, a phrase or even a sentence.

We approach is constructing a question answers pairs for each question and his one answers, in train dataset, we need to mark this is a positive sample or negative sample. In validation and test dataset, we employ the same processing mode, determining the matching degree of fit between a question and an answer, the highest will be chosen. The deep learning (DL) models are based on utilizing word distributed representation on both questions and answers, and building two layers long short-term memory (LSTM) respectively, on top of this, we introduce an efficient attention model for the answer embedding generation according to the question context, finally, a dense layer is used to determine the matching degree. For confirming the effect of the model, we also tested the QA-LSTM-CNN model and QA-LSTM with attention model to demonstrate the validity of two layers LSTM with attention. By experiment, we use two layers LSTM with attention model get a better performance.

2. Related work

Previously, feature extraction, semantic analysis was usually used in question answering task. For example word semantic relations were makeup based on WordNet in [1]. In [2] [3] using syntactical matching. [4] tried to fulfill the matching using minimal edit sequences between dependency parse trees. Above methods show effectiveness, but these relies on feature extraction and language tools.

With the development of neural networks, we adopt the method of word distributed representation from [5], LSTM model from [6], and the CNN model refers to paper [7]. State-of-the-art, there are three main directions to solve Question Answering Problems based on deep learning models: Firstly, the question and answer representations are learned and matched by similarity metrics [8] [9]. Secondly, a joint feature vector is constructed based on both the question and the answer, then task was converted into a classification problem [10]. Finally, the proposed models for textual generation can intrinsically be used for question answering task and generation [11] [12].

The approach in this task is based on the second category, attempting to transform this task into a classification task, using LSTM models and attention models. The model framework is a little similar to [9], but has distinct differences: first, our models employ a method by constructing joint feature vector sending to a dense layer, meanwhile, In order to extract features better, we have adopted a two-layer LSTM model with attention model, and achieved better performance.

3. Model description

In the section, we describe the two layers LSTM with attention model address the task. The following, introduce baseline models which use QA-LSTM models on both questions and their answer candidates and QA-LSTM-CNN and QA-LSTM-attention as the reference models. The structure of two layers LSTM with attention framework are shown in figure 1.

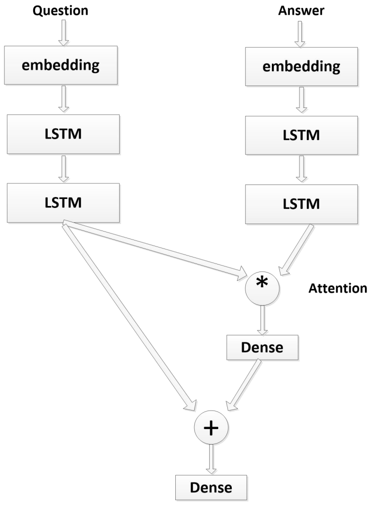

Figure 1. The structure of two layers LSTM with attention.

We train a word embedding by word2vec [2]. The word embedding provides the distributed representation for each token in question and answer respectively. The vectors have dimensionality of 300 and were trained using continuous bag-of-words architecture. Word embedding are parameters that can also be optimized during training. In preliminary experiment, there are two word distributed representation models we used. One is the word embedding from the pre-trained Word2Vec model which were trained by [12] on 100 billion words of Google News and are publicly available, word vector was initialized from an unsupervised neural language model. The other word2vec model is trained by supervised neural language model by a scientific knowledge story corpus which have about 600 thousand story. The format of each story is question and correct answer pairs, e.g. (question, correct answer), the parameter min count is 1, and the window is 10, words not present in the set of pre-trained words are initialized zeros. Through experiments, it is found that the word vector model trained by special scientific knowledge story corpus is more accurate, so the model is chosen in the end.

Each question and answer will be transformed to a word vector matrix and be entered into two layers LSTM models respectively. LSTM is a special type of RNN that can learn to rely on long-distance history and the immediate previous hidden vector, it's a remarkable variations of RNN to alleviate the gradient vanish problem. LSTM was proposed by [6] and has been improved and promoted by [13]. Given an input sentence sequence $x = \{ x_{(1)}, x_{(2)}, \cdots, x_{(n)} \}$, $x_{(t)}$ is the E dimension word vector in t time. The hidden vector $h_{(t)}$ (size is H) at the time step t is updated as follows.

$$i_t = \sigma(W_i [h_{(t-1)}, x_{(t)}]+ b_i) \tag{1}$$

$$f_t = \sigma(W_f [h_{(t-1)}, x_{(t)}] + b_f) \tag{2}$$

$$o_t = \sigma(W_o [h_{(t-1)}, x_{(t)}] + b_o) \tag{3}$$

$$\widetilde{C}_t = \tanh(W_c [h_{(t-1)}, x_{(t)}] + b_c) \tag{4}$$

$$C_t = i_t * \widetilde{C}_t + f_t * C_{t-1} \tag{5}$$

$$h_t = o_t * \tanh(C_t) \tag{6}$$

In the LSTM architecture, there are three gates (input i, forget f and output o), and a cell memory vector C. σ is sigmoid function. The input gate can determine how incoming vectors $x_{(t)}$ later the state of the memory cell. The output gate can allow the memory cell to have an effect on the outputs. Finally, the forget gate allows the cell to remember or forget its previous state. $[h_{(t-1)}, x_{(t)}]$ mean to concatenate vector $h_{(t-1)}$ and vector $x_{(t)}$, $W \in R^{H\times(H+E)}$, $b \in R^{H}$.

Questions and answers were respectively processed through the embedded layer and the two layers LSTM model, generate two sequences of LSTM output vectors. Now, we investigate a state-of-the-art attention model for the answer vector generation based on question, instead of generating QA representation respectively. If the input sentence is longer, semantics are expressed by an intermediate semantic vector, and the information of the word itself has disappeared, which results in the loss of a great deal of detail information. An attention mechanism are used to alleviate weakness by dynamically aligning the more informative parts of answer to the questions. This strategy are widely employed in many natural language processing tasks, such as machine translation and so on. We develop the efficient word-level attention above the two layers LSTM model. Answer output vector will be multiplied by a softmax weight, which is determined by the question outputting from LSTM. Specifically, attention model give more weights on certain words, just like tf-idf for each word, however, the weight is calculated by the question. Given the output vector of two layers LSTM on the answer side at time step t, $h_a(t)$, and the question embedding, o_q, the updated vector $h_a(t)$ for each answer token are formulated below.

$$m_{a,q}(t) = \tanh(W_{am}h_a(t) + W_{qm}o_q) \tag{7}$$

$$s_{a,q}(t) \propto \exp(W_{m,s}^{T} m_{a,q}(t)) \tag{8}$$

$$\tilde{h}_a(t) = ha(t)s_{a,q}(t) \tag{9}$$

Final, we merge the two sequences to one and perform flatten operations, the ultima output is passed through a two-dimensional softmax layer.

As reference, we apply a baseline QA-LSTM model and QA-LSTM-CNN model and QA-LSTM-attention. In QA-LSTM-CNN, we resort a CNN and MaxPooling built on every outputs of LSTM. We find that there is a certain improvement in accuracy compared with basic QA-LSTM model, it is because model can obtain a more composite n-gram representation of questions and answers. Convolutional structure only imposes local interactions between the inputs within a filter size m, size 3 filters are used here. Every windows with size of m order to capture m line information in LSTM output matrix, after the convolution operation, we have added a pooling layer which adopt MaxPooling, pool size is 2. Intuitively, the pooling operation can reduce the amount of computation of the model while retaining most of the information. In preliminary experiment, we do not see a significant difference in accuracy between different pooling methods, so MaxPooling is used only. The QA-LSTM-attention employ a single layer of LSTM with attention which achieves the same accuracy as the QA-LSTM-CNN model, different LSTM units are used here.

We also tried to insert CNN layers before the LSTM with attention framework and found the results were not good, and finally observe it better to replace the CNN with LSTM. Our guess is that the CNN network needs to pad a large number of zeros, resulting in a large amount of invalid data is created, reduces the capability of semantic feature extraction.

4. Related dataset

The task is come from a NLP competition, the train dataset consists of two parts: science and history curriculum dataset, SciQA and 8th Studystack flashcards.

Science and history curriculum: The dataset is a collection of American middle school science and history curriculums selected by competition organizers. A total of seven subsets (e.g. biology, chemistry) are included in this subjects, the trainset contains 3886 questions, and each questions have 4 answer candidates.

SciQA and 8th Studystack flashcards: SciQA and 8th Studystack flashcards is an additional train dataset corpus we introduced, download from https://www.studystack.com/. The dataset is also a multiple-choice question for science and history courses from various regional and national science examinations, there is 13455 sample which have the same for format as above

The validation dataset and test dataset are also scientific and historical questions, with the quantity being 669 and 812 respectively, these question not duplicate from the train dataset.

5. Experimental setup

The models in this task are implemented with keras, we use the accuracy on validation dataset to locate the best epoch and best hyper-parameters for test. The final rate of accuracy is expressed in the correct proportion chosen, formula is as follows:

$$accuracy = \frac{number\ of\ correct\ questions}{total\ number\ of\ questions} \tag{10}$$

We apply word2vec model which is trained by a scientific knowledge story corpus which have about 600 thousand story instead of the pre-trained google news Word2Vec model, and the vector dimensionality is 300. Word vectors are also parameters and can be optimized during training. We use the loss function of categorical cross entropy and the optimizer of adaptive moment estimation. We tried to include l_2 norm in preliminary experiments, but the regularization factors doesn't show any improvement, so we just used dropout layers to prevent overfitting, the parameter is 0.3.

The max length of questions and answers tokens sequence of training datasets is limited to 64, any tokens out of this range will be discarded. If the length of tokens sequence is not enough, then zero is added.

For comparison, we report the performance and analysis of four framework in table 1. The table summarizes the results of our models for the question answering task. From Row (1) to (2), we list the baseline QA-LSTM without either CNN structure or attention mechanism. We use epochs for 20, batch size for 512, and merge mode for sum, the parameter mask_zero means masking for input data with a variable number of time steps. The difference is that the different output dimensionality parameters of LSTM for questions or answers are used, we can observe that increasing the number of units helps to improve accuracy. We take the results of the simple LSTM model as baseline to measure other models.

From Row (3) to (4), CNN and MaxPooling layers are built on the top of the LSTM with different filter numbers with 100 or 200, we set the filter size on 3, we did not see better result if we increase m. The size of pooling operation is 2, and merge mode is concatenate. From the results of the LSTM-CNN model, we can perceive that the addition of CNN structure is beneficial to improve the capability of the model.

Row (5) correspond to QA-LSTM with the attention mechanism, output dimensionality parameters we are still using 64. We observe that the improvement form attention is remarkable, we get improvements from the baseline of simple LSTM model by 6.5% and 6.9% in validation and test dataset, compared to the QA-LSTM-CNN model, it also improved by more than 2%.

Row (6) is the framework proposed in this paper, two layers LSTM with attention. As can be seen from the result, the models get a significant result, achieve a precision of 37.67%. Compared to the single layer LSTM-attention model, it also improved by more than 2%.

Table 1. The results of four models on the question answering task.

Idx	Model	validation	Test
1	QA-LSTM baseline(output_dim=32, mask_zero=True)	28.12	29.14
2	QA-LSTM baseline(output_dim=64, mask_zero=True)	29.99	31.01
3	QA-LSTM-CNN(output_dim=64, filters=100, kernel_size=3 pool_size=2, border_mode="valid")	30.31	32.56
4	QA-LSTM-CNN(output_dim=64, filters=200, kernel_size=3 pool_size=2, border_mode="valid")	31.99	33.22
5	QA-LSTM-attention(output_dim=64)	34.61	36.02
6	two layers LSTM with attention(output_dim=64)	37.11	37.67

6. Conclusion

In this paper, we study the answer selection task by employing a two layers LSTM with attention models based deep learning framework. The proposed approach is to build distributed word embedding of question and answers which is trained by a scientific knowledge story corpus instead of the pre-trained google news Word2Vec model, meanwhile, does not rely on manually extracting features or linguistic tools, and can be applied to many domain. We further introduce attention mechanisms to solve question answering task, the answer vector generation based on question, instead of generating QA representation respectively, the weighted transformation can effectively improve the performance of sequences under Natural Language Processing. On this basis, we find that two layers LSTM with attention can achieve better performance compared to other framework. We conduct experimental using science and history curriculum dataset, SciQA and 8th Studystack flashcards, our experimental results demonstrate that the proposed model outperform a variety of other framework. Therefore, the two layers LSTM with attention model is the framework recommended by this paper for question answering task.

References

[1] Yih, Wen-tau, Chang, Ming-Wei, Meek, Christopher, and Pastusiak, Andrzej. Question answering using enhanced lexical semantic models. Proceedings of the 51st Annual Meeting of the Association for Computational Linguist (ACL), 2013.

[2] Wang,Meng qiu and Manning,Christopher. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering. The Proceedings of the 23rd International Conference on Computational Linguistics (COLING), 2010.

[3] Wang, Mengqiu, Smith, Noah, and Teruko, Mitamura. What is the jeopardy model? a quasisynchronous grammar for qa. The Proceedings of EMNLP-CoNLL, 2007.

[4] Heilman, Michael and Smith, Noah A. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. Annual Conference of the North American Chapter of the Association for Computational Linguistics. Association for Computational Linguistics

(NAACL), 2010.

[5] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean, “Distributed representations of words and phrases and their compositionality,” in Advances in Neural Information Processing Systems 26, pp. 3111–19. Curran Associates, Inc., 2013.

[6] Hochreiter, Sepp and Schmidhuber, Jurgen. Long short-term memory. Neural Computation, 1997.

[7] Yoon Kim, “Convolutionalneural networksfor sentence classification,” in Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, October 2014, pp. 1746–51, Association for Computational Linguistics.

[8] Feng, Minwei, Xiang, Bing, Glass, Michael, Wang, Lidan, and Zhou, Bowen. Applying deep learning to answer selection: A study and an open task. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), 2015.

[9] Ming Tan, Cicero dos Santos, Bing Xiang & Bowen Zhou. Lstm-based deep learning models for non-factoid answer selection. arXiv preprint arXiv:1511.04108, 2015.

[10] Wang, Di and Nyberg, Eric. A long short-term memory model for answer sentence selection in question answering. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015.

[11] Bahdanau, Dzmitry, Cho, KyungHyun, and Bengio, Yoshua. Neural machine translation by jointly learning to align and translate. Proceedings of International conference of learning representations, 2015.

[12] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems. 2013: 3111-19.

[13] Graves, Alex, Mohamed, Abdel-rahman, and Hinton, Geoffrey. Speech recognition with deep recurrent neural networks. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013.