YNU-HPCC at SemEval-2018 Task 11: Using an Attention-based CNN-LSTM for Machine Comprehension Using Commonsense Knowledge

Hang Yuan, Jin Wang and **Xuejie Zhang**
School of Information Science and Engineering
Yunnan University
Kunming, P.R. China
Contact:xjzhang@ynu.edu.cn

Abstract

This shared task is a typical question answering (QA) task. Specially, this task must give the answer to the question based on the text provided. The essence of the problem is actually reading comprehension. For each question, there are two candidate answers, and only one of them is correct. Existing method for this task is to use convolutional neural network (CNN) and recurrent neural network (RNN) or their improved models, such as long short-term memory (LSTM). In this paper, an attention-based CNN-LSTM model is proposed for this task. By adding an attention mechanism and combining the two models, the experimental results have been significantly improved. The accuracy of our final submission is 0.7143.

1 Introduction

Question answering has long been an important research topic in the field of natural language processing. Prior to this, there have been many similar tasks, and many scholars have made very significant contributions to the research in this field. Such as the Allen AI Science Challenge on the Kaggle (Schoenick et al., 2016) and the IJCNLP-2017 shared task 5: Multi-choice Question Answering in Exams (Yuan et al., 2017).

Machine comprehension using commonsense knowledge is required to answer multiple-choice questions based on narrative texts about daily activities of human beings. The answer to many questions does not appear directly in the text, but requires simple reasoning to achieve. In terms of the nature of the problem, this task can be considered as a binary classification. That is, for each question, the candidate answers are divided into two categories: the correct answers and the wrong answers.

In recent years, many achievements have been made in machine comprehension-based question answering. Among the existing methods, the main differences are in the data processing and the application of the model. A dataset for multi-choice question answering was released by Richardson et al. (2013). Clark (2015) described how to obtain more information from the background knowledge base by introducing the use of background knowledge to build the best scene. A large cloze-style dataset using CNN and Daily Mail news articles was created by Hermann et al. (2015). Unlike previous datasets, Rajpurkar et al. (2016) released a machine comprehension-based dataset (SQuAD dataset). It contains over 1M text-question-answer triples crawled from 536 Wikipedia articles, and the questions and answers are structured primarily through crowdsourcing. It also requires people to submit up to five article-based questions and provide the correct answer that has appeared in the original text. For the open-domain QA dataset, it is even more challenging to get answers because it requires simple word matching and some simple reasoning. In SearchQA (Dunn et al., 2017), the question-answer pairs are crawled from the Jeopardy archives and are augmented with text snippets retrieved from Google search. Kundu and Ng (2018) proposed an end-to-end, problem-based, multi-factor attention network that addresses the task of answering document-based questions. This model can collect scattered evidence from multiple sentences for the generation of answers.

In this paper, we mainly propose to use an attention-based CNN-LSTM model for this task. The word-embedding model we choose is Word2Vec. Then, the word vectors are fed into the convolutional neural network (CNN) layer. After that, the results of the CNN layer are fed into the long short-term memory (LSTM) layer. Finally, an attention mechanism is added into the neu-

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 1058–1062
New Orleans, Louisiana, June 5–6, 2018.

ral networks, and the prediction results are output via the softmax activation. All the data is processed into (text-question-answer) form. For each candidate answer, the system will give a correct probability (probability of a correct answer) and a wrong probability (probability of a wrong answer), and the sum of these probabilities is 1. The answer with the larger correct probability of those two candidate answers will be selected by the system as the correct answer. Furthermore, in order to exclude the experimental error caused by chance, nine such models are assembled together for training. The answers are obtained by hard voting. At the same time, we also selected a number of other models (such as the Bi-LSTM, the attention-based Bi-LSTM and the attention-based LSTM) for comparative experiments. The experimental results show that attention-based CNN-LSTM can achieve better results when using Word2Vec as the word embedding technique.

The rest of our paper is structured as follows. Section 2 introduces the CNN, LSTM and attention-based CNN-LSTM. Experiments and evaluation will be described in Section 3. The conclusions are drawn in Section 4.

2 Model

For this task, we select 10 models for comparison. Among these models, the attention-based CNN-LSTM models can get the best results. This model combines the CNN with the LSTM and incorporates the attention mechanism. The most important elements of this model are the CNN, the LSTM and the attention mechanism.

The CNN has been proven to be very effective for local feature extraction. Since the operation of the CNN layer will lose the long-distance dependency, a LSTM layer is added to handle the sequential information of the input vectors. The attention mechanism is a good solution to the information vanish problem in long sequence input situations. When dealing with machine comprehension problems, their combined use is more effective than their use individually.

2.1 Convolutional Neural Network

The convolutional neural network was originally used to process image data. In recent years, the application of the convolution neural network has gradually infiltrated into many fields, such as speech recognition and natural language processing. The convolutional neural network consists of three parts. The first part is the input layer. The second part consists of a combination of n convolution layers and a pooling layer. The third part consists of a fully connected multi-layer perceptron classifier. The difference between the convolutional neural networks and ordinary neural networks is that the convolutional neural networks consist of a feature extractor made up of the convolutional layers and the sub-sampling layers. In the convolutional layers, one neuron is connected to only a few adjacent neurons.

In our experiment, the convolution layer was mainly used to extract features. The convolution matrix has m columns, and m is the maximum length of the sentence. The convolution matrix has n rows, and n refers to the number of sentences. The direct benefit of sharing weights (convolution kernels) is to reduce the number of connections between layers of the network while reducing the risk of overfitting.

2.2 Long Short-Term Memory

Traditional recursive neural networks are ineffective when dealing with very long sentences. The LSTM model is developed to solve the gradient vanishing or exploding problems in the RNN. Currently, the LSTM is mainly used in natural language processing such as speech recognition and machine translation. Compared with the traditional RNN, a LSTM unit is added to the traditional model for judging the usefulness of information. Each unit mainly contains three gates (the forget gate, the input gate, and the output gate) and a memory cell. The system will judge the usefulness of the information after the input information is fed into an LSTM. Only the information that matches the rules of the algorithm will be saved, and the other information will be discarded by the forget gate.

In our experiment, the LSTM layer is designed to ensure that important information in the front part of a long sequence can also have an impact on the processing of the latter part of the long sequence.

2.3 Attention-based CNN-LSTM

Both the CNN and LSTM models have their own advantages and disadvantages. The former performs well in local feature extraction, but easily loses the long-distance dependency of words. The latter can only solve the problem of information

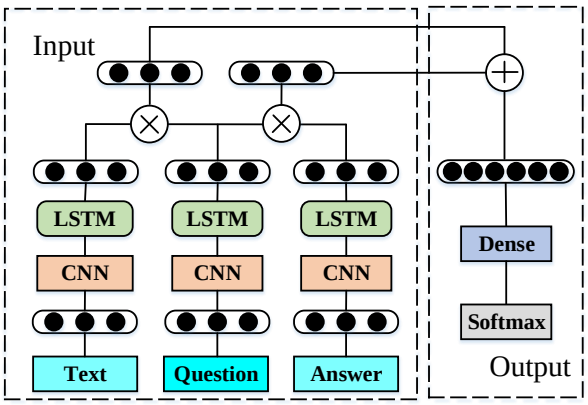

Figure 1: Architecture of a AT-CNN-LSTM.

Word2Vec	Acc
CNN	0.638
LSTM	0.651
BiLSTM	0.654
AT-BiLSTM	0.669
CNN-LSTM	0.687
AT-CNN-LSTM	0.699
AT-CNN-LSTM Ensemble	0.714
GloVe	**Acc**
CNN	0.629
LSTM	0.642
BiLSTM	0.649
AT-BiLSTM	0.658
CNN-LSTM	0.666
AT-CNN-LSTM	0.678
AT-CNN-LSTM Ensemble	0.692

Table 1: Comparative experiment results

vanish in long sequence input situations to a certain extent. Therefore, we combined the two models with additional attention mechanisms to form an attention-based CNN-LSTM model for this task, as shown in Fig 1.

In this model, all the texts, questions, and answers will be converted into word vectors through the word-embedding layer. These word vectors will be first fed into the CNN layer, and then the output feature vectors will be fed into the LSTM layer. Subsequently, the word vectors are expressed as hidden vectors. Next, the attention mechanism assigns a weight to each hidden vector. The attention mechanism produces an attention weight vector and a weighted hidden representation. The attention weight vector is mainly obtained by calculating the similarity. The main operation here is dot. An attention weight vector is generated by computing the question vector matrix and answer vector matrix. Another attention weight vector is generated by computing the question vector matrix and text vector matrix. Next, two attention weight vectors are connected via contact method. Then the attention weight vector is fed into the softmax layer.

The attention mechanism allows the model to retain some important hidden information when the sentences are quite long. In our task, the text, questions and answers are relatively long sentences. The use of a standard CNN or LSTM will result in the loss of hidden information. To address this possible problem, the attention-based CNN-LSTM model is used to design the machine comprehension system.

3 Experiments and Evaluation

3.1 Experiments

Data Pre-processing. The dataset provided by the organizer mainly include three parts: texts, questions, and answers. In the data pre-processing phase, texts and questions-answers pairs are divided into two separate files. The content of each piece of text data mainly includes the text id and the text content. Each question-answer pair data mainly includes the text id, the question id, and the question-answer pair content. In the final experiment, we added validation data to the training set to expand the training data. We also tried sorting the training data randomly to expand the data set, but the result was not satisfactory. All input data is converted into word vectors through the word-embedding layer, and the word-embedding model is Word2Vec. Here, all the punctuation is ignored, and all non-English characters are treated as unknown words. In the word vectors, unknown word vectors are randomly generated from a uniform distribution U(-0.25, 0.25).

Two different methods of word-embedding are used in this experiment: Word2Vec and GloVe (Pennington et al., 2014). They are used to initialize the weights of the embedding layer in building 300-dimension word vectors for all the texts and question-answer pairs. Word2Vec achieved better performance than GloVe in every model we used. Through the list of unknown words, we know that the use of Word2Vec results in fewer unknown words than GloVe.

Implementation Details. All the code involved in this experiment was written in Python 3.5.2. Keras 2.0.4 is used as the framework for the program. The backend used in this experiment is Ten-

Parameters	Optimal
Filter size	250
Kernel size	3
Dropout rate	0.3
Epoch	10
Batch size	64
Word embedding dim	300
Accuracy	0.7143

Table 2: Optimal parameters

sorFlow 1.1.0. We use the attention-based CNN-LSTM to obtain the results for the test dataset.

The first model we use is a standard CNN model. As shown in Table 1, it can achieve an accuracy of 0.638 and 0.629 when respectively using Word2Vec and GloVe as the word-embedding layer. Due to the impact of jagged sentences, the poor result obtained by the CNN model is predictable. After that, a standard LSTM model is used to complete this task. It can achieve an accuracy of 0.651 and 0.642 when respectively using Word2Vec and GloVe as the word-embedding layer. However, the results obtained by the LSTM model have been somewhat improved over the CNN model. Next, we also apply the BiLSTM model and the best result is 0.654, but there are still many points that can be improved. Combining the two models effectively seems to be the perfect choice. In this way, we achieve an accuracy of 0.687. Finally, after adding the attention mechanism, the result is raised to 0.699. Under the same experimental conditions, the attention-based CNN-LSTM model obtained a better result than other models we used in most cases (Wang et al., 2016). To exclude the experimental error caused by chance, nine such models are assembled together for training. The final accuracy can be raised to 0.714. Table 1 presents the results of a comparative experiment for all models we used.

The choice of model parameters has a significant effect on the final accuracy. The main parameters of this model are the word-embedding dimension, the batch size, the epoch, the filter size, the kernel size, the dropout and so on. To get the optimal parameters, the Sklearn grid search function (Liu et al., 2015) is used to determine the best combination of the parameters. Table 2 lists the parameters of the model when the best result is obtained.

3.2 Evaluation

Evaluation Metrics. For this experiment, it measures how well a system is capable of correctly answering questions that may involve commonsense knowledge. This problem is a typical binary classification problem. Therefore, the system is evaluated by calculating the accuracy.

Results. According to the final results provided by the organizers, a total of 199 teams enrolled in the competition. Only 24 teams eventually submitted their results. Our team ranked 13th overall among all teams. As shown in Table 1, the attention-based CNN-LSTM model can achieve the highest accuracy when using Word2Vec as the word-embedding layer. This model combines the advantages of the CNN model, the LSTM model and the attention mechanism. Furthermore, the use of Word2Vec for word-embedding is better than the GloVe word-embedding. The main difference between the two embeddings is in the training sets. The training sets of Word2Vec are practically from the news, while the training sets of GloVe are from Twitter. Therefore, the Word2Vec data source is better suited to this task.

4 Conclusion

This paper mainly focuses on our attention-based CNN-LSTM system for the task of machine comprehension using commonsense knowledge. It gives a brief introduction of the model and gives a detailed description for the experimental process and results. Compared with the attention-based LSTM model, the attention-based CNN-LSTM is better at feature extraction. The experimental results also show that the use of multiple models for ensemble training can also, to some extent, avoid the accidental results and improve the accuracy of the experiment. In the future, we will focus on methods combination and models ensemble. In addition, our team will also continue to propose a new model that can improve the existing results.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (NSFC) under Grants No.61702443 and No.61762091, and in part by Educational Commission of Yunnan Province of China under Grant No.2017ZZX030. The authors would like to thank the anonymous reviewers and the area chairs for their constructive comments.

References

Peter Clark. 2015. Elementary School Science and Math Tests as a Driver for AI: Take the Aristo Challenge! In *Proceedings of the Twenty Ninth AAAI Conference on Artificial Intelligence*, pages 4019–4021.

Matthew Dunn, Levent Sagun, Mike Higgins, V. Ugur Guney, Volkan Cirik, and Kyunghyun Cho. 2017. SearchQA: A New QA Dataset Augmented with Context from a Search Engine. In *arXiv:1704.05179*.

Hermann, Kocisky, Grefenstette, Espeholt, Kay, Suleyman, and Blunsom. 2015. Teaching Machines to Read and Comprehend. In *Advances in NIPS*.

Souvik Kundu and Hwee Tou Ng. 2018. A Question-Focused Multi-Factor Attention Network for Question Answering. In *arXiv:1801.08290v1*.

Pengfei Liu, Shafiq Joty, and Helen Meng. 2015. Fine-grained Opinion Mining with Recurrent Neural Networks and Word Embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1433–1443.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2383–2392.

Matthew Richardson, Christopher, J.C. Burges, and Erin Renshaw. 2013. MCTest: A Challenge Dataset for the Opendomain Machine Comprehension of Text. In *Proceedings of the 2013 Conference on Empirical Methods on Natural Language Processing (EMNLP)*, pages 397–401.

Carissa Schoenick, Peter Clark, Oyvind Tafjord, Peter Turney, and Oren Etzioni. 2016. Moving beyond the Turing Test with the Allen AI Science Challenge. In *arXiv:1604.04315*.

Jin Wang, Liang Chih Yu, K. Robert Lai, and Xuejie Zhang. 2016. Dimensional Sentiment Analysis Using a Regional CNN-LSTM Model. In *Proceedings of the Association for Computational Linguistics(ACL)*, pages 225–230.

Hang Yuan, You Zhang, Jin Wang, and Xuejie Zhang. 2017. YNU-HPCC at IJCNLP-2017 Task 5: Multi-choice Question Answering in Exams Using an Attention-based LSTM Model. In *Proceedings of IJCNLP, Shared Tasks*, pages 208–212.