ZMU at SemEval-2018 Task 11: Machine Comprehension Task using Deep Learning Models

Yongbin Li[1,2], **Xiaobing Zhou**[1,*]
[1]Yunnan University, Kunming, Yunnan, P.R. China
[2]Zunyi Medical University, Zunyi, Guizhou, P.R. China
[*] Corresponding author, zhouxb.cn@gmail.com

Abstract

Machine Comprehension of text is a typical Natural Language Processing task which remains an elusive challenge. This paper is to solve the task 11 of SemEval-2018, Machine Comprehension using Commonsense Knowledge task. We use deep learning model to solve the problem. We build distributed word embedding of text, question and answering respectively instead of manually extracting features by linguistic tools. Meanwhile, we use a series of frameworks such as CNN model, LSTM model, LSTM with attention model and biLSTM with attention model for processing word vector. Experiments demonstrate the superior performance of biLSTM with attention framework compared to other models. We also delete high frequency words and combine word vector and data augmentation methods, achieved a certain effect. The approach we proposed rank 6th in official results, with accuracy rate of 0.7437 in test dataset.

1 Introduction

Machine Comprehension of text is one of the important goals of natural language processing. The traditional approaches to machine reading and comprehension have been based on either hand engineered grammars (Riloff and Thelen, 2000), or information extraction methods of detecting predicate argument triples that can later be queried as a relational database (Poon et al., 2010). These methods show effectiveness, but they rely on feature extraction and language tools. Recently, with the advances of neural networks, there have been great interests in building neural architectures for various NLP task, including several pieces of work on machine comprehension (Hermann et al., 2015; Hill et al., 2015; Yin et al., 2016; Kadlec et al., 2016; Cui et al., 2016), which have gained significant performance in machine comprehension domain. We also adopt deep learning models to solve this task.

The goal of Machine Comprehension using Commonsense Knowledge task is to choose a correct answer in two candidates to the question based on the contents of text. This task relates to how the inclusion of commonsense knowledge in the form of script knowledge would benefit machine comprehension systems, answering the questions requires knowledge beyond the facts mentioned in the text. We do not employ extra commonsense knowledge resources in the proposed approach, we assume that word vectors have contained some commonsense knowledge information, so we only use the deep learning model to solve this problem.

In the train dataset provided by this task, there are 1432 instances, each instance contains a text and several questions, and each question is associated with a set of two answers which are short and limited to a few words. The texts used in this task cover more than 100 everyday scenarios, hence include a wide variety of human activities. Therefore, each example can be summed up as {*text, question, answer_0, answer_1, correct option*}. There are 9731, 1411, 2797 examples in train, validation, test datasets, respectively.

Being a binary classification task, we split an example into two triples, which are {*text, question, answer_0*} and {*text, question, answer_1*}, the label is true or false. In validation and test datasets, we employ the same processing mode to determine the matching degree of fit between triples, the highest will be chosen. We adopt the method of word distributed representation from (Mikolov et al., 2013) and a series of deep learning (DL) models, such as Convolutional Neural Network (CNN) from (Kim, 2014), Long Short-Term Memory (LSTM) model proposed from (Hochreiter and Schmidhuber, 1997) and improved by (Graves et al., 2013), attention mechanism from

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 1073–1077
New Orleans, Louisiana, June 5–6, 2018.

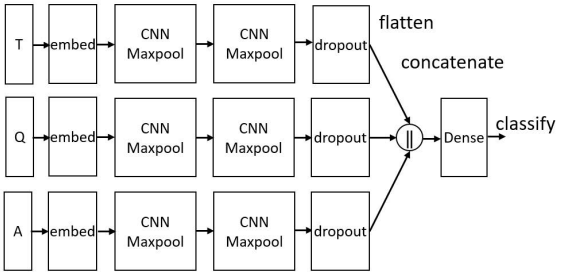

Figure 1: The architecture of CNN framework, T for text, Q for question, A for answer.

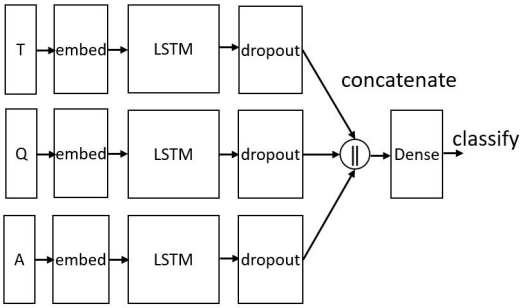

Figure 2: The architecture of LSTM framework.

(Graves and Schmidhuber, 2005). The four main frameworks we applied are as follows:

- CNN framework
- LSTM framework
- LSTM with attention framework
- biLSTM with attention framework

Above the framework, a joint feature vector is constructed, which is used to classify (Tan et al., 2015). In order to increase the accuracy of the model, we also delete high frequency word and combine word vector and data augmentation methods, thus achieve a better effect. Experiments demonstrate the superior performance of biLSTM with attention framework compared to other models, and data preprocessing is also important to improve the model accuracy.

2 Model description

In this section, we describe the four main proposed deep learning frameworks, which are shown in figures 1 to 4. The main idea of these systems is the same: learn a distributed vector representation of given text, question and answer candidates, then use a dense layer which processes the joint feature to measure the matching degree.

2.1 CNN framework

The first framework is based on CNN model. Step one is to obtain word embedding from pre-trained word distributed representation models. In preliminary experiment, there are two distributed representation models used. One is the pre-trained word2vec model which is trained by 100 billion words of Google News and has a dimensionality of 300, the other is pre-trained Glove model which is trained by Wikipedia data and has a dimensionality of 300 too. The two models are all initialized from an unsupervised neural language model. The word embedding provides the distributed representation for each token in sequence.

Text, question and answer will be transformed to a word vector matrix and be entered into CNN layer respectively. In order to get more comprehensive representation of semantic features, we adopt double layer CNN model. The numbers of filters are 64 and 32, respectively, and the filter size is set as 3. After each CNN layer, we resort to a MaxPooling layer of size 2.

Above the CNN layer, the output of text, question and answer is merged to one and performs flatten operations, through a dense layer, the final output is passed through a two-dimensional softmax layer.

2.2 LSTM framework

Its the same way of producing word vector representation in embedding layer. Because LSTM model can process variable length sequence, masking method is introduced. The main principle of masking is to skip time steps which tokens are equal to zero, thus ignoring the meaningless padding in the text.

Above the embedding layer, we introduce the LSTM layer with a unit number of 64. LSTM is a special type of RNN that has three gates (input i, forget f and output o), and a cell memory vector C, and can learn to rely on long-distance history and the immediate previous hidden vector. Its a remarkable variations of RNN to alleviate the gradient vanish problem. Through the LSTM layer, text, question and answer will be transformed to a vector respectively.

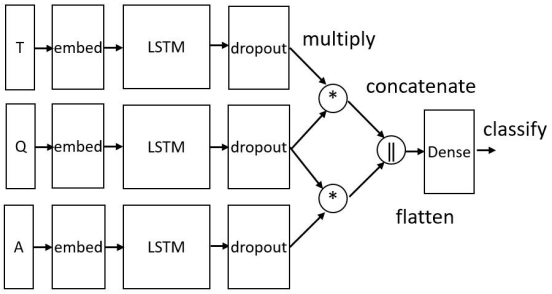

Figure 3: The architecture of LSTM with attention framework.

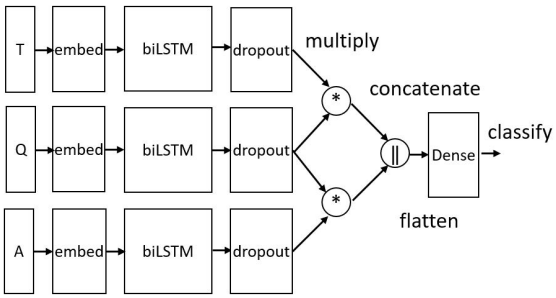

Figure 4: The architecture of biLSTM with attention framework.

2.3 LSTM with attention framework

Like the LSTM framework, the masking method is used in the embedding layer. Text, question and answer are respectively processed through the embedded layer and the LSTM layer, generating three sequences of LSTM output vectors. Unlike the LSTM framework, here returns full output sequences, instead of the final output of model.

Now, we investigate a state-of-the-art attention model for the question vector generated by text, and the answer vector generated by question, instead of generating representation respectively. If the input sentence is long, 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 is used to alleviate the weakness by dynamically aligning the more informative parts. Specifically, attention model gives more weights on certain words, just like tf-idf for each word, while the weight is calculated by another vector. Therefore, the formula of the attention matrix f of text and question vectors is as follows:

$$f(m_t, m_q) = {m_t}^T m_q \tag{1}$$

where m_t and m_q correspond to text and question vectors produced by previous LSTM layers, respectively. The attention matrix of answer and question is constructed in the same way.

2.4 biLSTM with attention framework

The framework is similar to the above, just changing the LSTM model into a biLSTM model. Single direction LSTMs suffer a weakness of not utilizing the contextual information from the future tokens. biLSTM utilizes both the previous and future context by processing the sequence on two directions, and generates two independent sequences of LSTM output vectors.

3 Data Preprocessing

In order to improve the accuracy of the model, we have tried a series of data preprocessing methods, such as deleting high frequency words, combining word vectors and data augmentation methods, which achieve a certain effect.

3.1 Deleting high frequency words

In a large corpus, many common words appear, such as "the", "a" and so on. Although these words have higher word frequency, few useful information can be provided. We try to delete stopwords through the NLTK tools, but the effect was not ideal. So we calculate the word frequency statistics on the words in all dataset, and delete the top 20 words in frequency, that is, the most frequently occurring 20 words. We also tried other numbers in preliminary experiment, but 20 worked best.

3.2 Combining word vectors

Pre-trained word2vec model is trained by Google News, Glove model is trained by Wikipedia data. The effect of the former is slightly better. In order to obtain more comprehensive semantic features, we try to combine two vectors of each token, so that the word vector of each token is transformed into 600 dimensions, which is better than using only one word vector model.

3.3 Data augmentation

Our idea is inspired by data augmentation in the image domain, where one can increase the amount of train data by the geometric transformation of the image. In order to enrich the train dataset of

	Framework	val	test
1	CNN	71.32	70.13
2	CNN(data augmentation)	72.86	70.65
3	LSTM	72.10	69.57
4	LSTM(data augmentation)	72.57	70.36
5	LSTM with attention	73.11	69.97
6	LSTM with attention(data augmentation)	73.99	70.15
7	biLSTM with attention	75.90	71.11
8	biLSTM with attention(data augmentation)	76.61	72.47
9	biLSTM with attention(data augmentation and combine word vector)	77.75	74.37

Table 1: Results of four main framework

images, extract image features better and generalize models (prevent models from over fitting), data augmentation is done on images data. We know that in text understanding, we can still read articles even if we disorder the order of the words. In this task, we've implemented the data augmentation by randomly disordering the word order in the sentence. The preliminary implementation proves that this method is effective.

4 Experimental setup

Our approach in this task use the accuracy on validation dataset to locate the best parameters. The final rate of accuracy is expressed in the correct proportion chosen in test dataset. **All the model parameters were adjusted by preliminary experiment, at the same time, the results are taken three times, and the average value is taken**.

In the experiment, we use the loss function of categorical cross entropy and the optimizer of adaptive moment estimation. The length of text, question and answer tokens sequence all take the maximum length, if the length is not enough, then zero is added. To prevent over fitting, we employ dropout layers which the parameter is 0.3.

For comparison, we report the performance and analysis of four framework in Table 1, which summarizes the results of our system for this task. All the experiments have deleted the high frequency words. The word embedding we employed is word2vec in Rows (1) to (8). Because in preliminary experiment, the accuracy of model using word2vec is generally better than Glove.

In Row (1) to (2), we list the results on validation dataset and test dataset respectively of CNN framework which employ filter size of 3, and filter number of 64. The difference is that Row (2) model uses the data augmentation. Row (3) to (4) correspond to LSTM framework which uses 64 as output dimensionality parameter of LSTM unit. The framework results in similar result with the CNN framework. In Row (5) to (6), we can observe that the framework for using the attention mechanism has been significantly improved in the accuracy rate. In Row (7) to (8), the improvement from biLSTM with attention compared to LSTM with attention is remarkable, increase more than 2%, illustrating that Bi-directional LSTM can achieve more comprehensive features than unidirectional LSTM. Row (9) is the approach proposed in this paper, which combines word2vec vector and Glove vector of each tokens. The model gets a significantly result, achieving a precision of 77.75% in validation dataset and 74.37% in test dataset. Compared to single word2vec, the improvement on the test set is more significant.

5 Conclusion

In this paper, we solve the Machine Comprehension Task by employing four main frameworks and a series of Data Preprocessing methods. Although the commonsense knowledge library is not used, the results are acceptable. The experiment results demonstrate the effectiveness of the biLSTM with attention framework in dealing with this task, the Bi-directional LSTM model is more advanced than the unidirectional LSTM model, and attention mechanism allows a model to focus on the aspects of a text that it will help answering a question. For a deep learning model, the Data Preprocessing is more critical, data augmentation and combining word vectors are beneficial to improve the model ability in some task backgrounds.

Acknowledgments

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

References

Yiming Cui, Ting Liu, Zhipeng Chen, Shijin Wang, and Guoping Hu. 2016. Consensus attention-based neural networks for chinese reading comprehension. *arXiv preprint arXiv:1607.02250*.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *Acoustics, speech and signal processing (icassp), 2013 IEEE international conference on*, pages 6645–6649. IEEE.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional lstm networks. In *Neural Networks, 2005. IJCNN'05. Proceedings. 2005 IEEE International Joint Conference on*, volume 4, pages 2047–2052. IEEE.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *Advances in Neural Information Processing Systems*, pages 1693–1701.

Felix Hill, Antoine Bordes, Sumit Chopra, and Jason Weston. 2015. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv preprint arXiv:1511.02301*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Rudolf Kadlec, Martin Schmid, Ondrej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. *arXiv preprint arXiv:1603.01547*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Hoifung Poon, Janara Christensen, Pedro Domingos, Oren Etzioni, Raphael Hoffmann, Chloe Kiddon, Thomas Lin, Xiao Ling, Alan Ritter, Stefan Schoenmackers, et al. 2010. Machine reading at the university of washington. In *Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading*, pages 87–95. Association for Computational Linguistics.

Ellen Riloff and Michael Thelen. 2000. A rule-based question answering system for reading comprehension tests. In *Proceedings of the 2000 ANLP/NAACL Workshop on Reading comprehension tests as evaluation for computer-based language understanding sytems-Volume 6*, pages 13–19. Association for Computational Linguistics.

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*.

Wenpeng Yin, Sebastian Ebert, and Hinrich Schütze. 2016. Attention-based convolutional neural network for machine comprehension. *arXiv preprint arXiv:1602.04341*.