YNU_Deep at SemEval-2018 Task 11: An Ensemble of Attention-based BiLSTM Models for Machine Comprehension

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

Abstract

This paper reports our submission to task 11 (Machine Comprehension using Commonsense Knowledge) in SemEval 2018. We firstly use GloVe to learn the distributed representations automatically from the instance, question and answer triples. Then an attention-based Bidirectional LSTM (BiLSTM) model is used to encode the triples. We also perform a simple ensemble method to improve the effectiveness of our model. The system we developed obtains an encouraging result on this task. It achieves the accuracy 0.7472 on the test set. We rank 5th according to the official ranking.

1 Introduction

Machine comprehension of text is one of the ultimate goals of natural language processing. The machine comprehension problem can be formulated as follows: Given an instance i, a question q and an answer candidate pool $\{a_1, a_2, \ldots, a_s\}$, the aim is to search for the best answer candidate a_k, where $1 \leq k \leq s$. The major challenge of this task is that the words in the answer do not necessarily appear in the instance.

In recent years, deep learning models are widely used in the field of NLP, such as semantic analysis (Tang et al., 2015), machine translation (Bahdanau et al., 2014) and text summarization (Rush et al., 2015). (Bahdanau et al., 2014) also introduced the attention mechanism into NLP task for the first time. This attention-based model yielded state-of-the-art performance on the machine translation task. (Hermann et al., 2015) built a supervised reading comprehension data set, the CNN/Daily Mail data sets[1]. They also presented Attentive Reader for machine comprehension, which allows a model to focus on the aspects of an instance that can help to answer a question, and also allows us to visualize its inference process (Hermann et al., 2015). The key point of the attention-based models is the design of attention function. Compared to Attentive Reader, Attention Sum Reader (Kadlec et al., 2016) used the dot products instead of a $tanh$ layer to compute the attention between question and contextual embeddings. Stanford Attention Reader (Chen et al., 2016) took a bilinear term as the attention function and obtained state-of-the-art results on the CNN/Daily Mail data sets.

The reasoning process was implemented in some models for machine comprehension. Memory Networks (Sukhbaatar et al., 2015) was the first model to propose reasoning process, which had important influence on other follow-up models. Compared to the traditional attention model, Memory Networks additionally uses a function t that constantly updates the representation of the instance and the question so as to realize the reasoning process. (Tseng et al., 2016) proposed an attention-based multi-hop recurrent neural network which achieved good performance on the machine listening comprehension test of TOEFL. Other reasoning models (Dhingra et al., 2017; Sordoni et al., 2016) shared the same idea as previous models, i.e., the representations of the instance and the question embedding were updated through continuous conversion of attention. Some more complex models (Hu et al., 2017; Liu et al., 2017) were proposed based on SQuAD data set (Rajpurkar et al., 2016). Their performances have been very close to or even exceeded the human performance on this dataset.

In this paper, we introduce a simple ensemble method on multiple identical attention-based BiLSTM models, only changing the dropout parameters in each model. We use each model to generate a soft prediction, and sum each result, then take the sum as the final prediction result. Experiments

[1] http://www.github.com/deepmind/rc-data/

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 1043–1047
New Orleans, Louisiana, June 5–6, 2018.

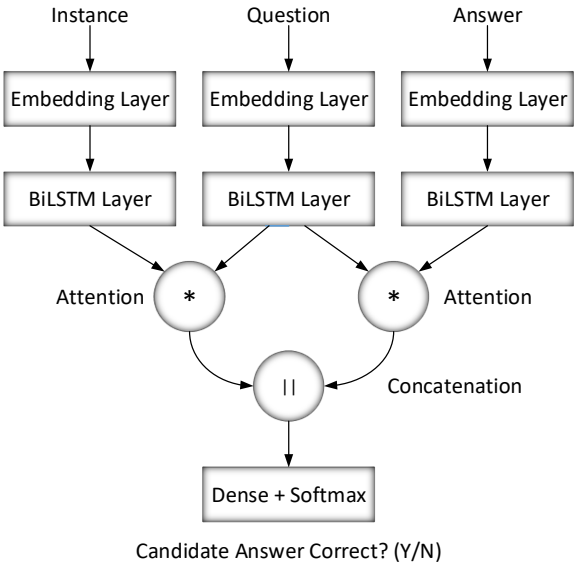

Figure 1: Our attention-based BiLSTM model for machine comprehension.

show that the ensemble model is about 2% higher than the single model in terms of accuracy on both development and test sets. Besides, we also made our code available online[2].

2 System Description

Our model is called an ensemble of attention-based BiLSTM models. Firstly, we use an embedding layer to obtain the distributed representations of the instance, question and answer triples. They are encoded by three different BiLSTM layers. The attention mechanism is implemented by *dot* products via a merge layer. Finally, we assign the same weight to each model when ensembling. The final result is the sum of the soft probabilities yielded by each single model. We keep the structure of each model the same, just fine tune the dropout parameters. The model architecture is shown in Figure 1. The attention mechanism is developed by calculating the *dot* product of the outputs from two BiLSTM layers. Then we use '||' operation to concatenate two matrices from the previous layer in the specified dimension. Finally, a $Dense$ fully connected layer with activation $softmax$ is used to get the predicted probabilities.

[2]https://github.com/Deep1994/An-Ensemble-of-Attention-based-BiLSTM-Model-for-Machine-Comprehension

2.1 BiLSTM

Single direction LSTM (Hochreiter and Schmidhuber, 1997) suffers a weakness of not using the contextual information from the future tokens. Bidirectional LSTM (BiLSTM) exploits both the previous and future context by processing the sequence on two directions and generates two independent sequences of LSTM output vectors. One processes the input sequence in the forward direction, while the other processes the input in the backward direction. The words in the instances, questions and answers are represented by the concatenation of the hidden layer outputs in both directions at each time step.

2.2 Word Embedding

Word embedding is arguably the most widely known technology in the recent history of NLP. It is well-known that using pre-trained embedding helps (Kim, 2014). We try two word embedding tools, GloVe (Pennington et al., 2014) and Word2Vec (Mikolov et al., 2013) on this task.

Tool	Size	Vocab	Dimension
GloVe	5.5G	2.2million	300
Word2Vec	3.5G	3million	300

Table 1: Summary statistics for the embedding tools: Size is the file size after decompression. Both tools have a dimension of 300. The Vocab is the number of word vectors contained in the tool.

2.3 Attention Mechanism

The LSTM model can alleviate the problem of gradient vanishing, but this problem persists in long range reading comprehension contexts. The attention mechanism breaks the constraint on fix-length vector as the context vector, enables the model to focus on those more helpful to outputs. (Luong et al., 2015) presented several attention computation ways, such as dot, $general$, $concat$. In our model, we adopt the dot mode to compute the attention. After BiLSTM layer, we implement a dot product operation on the output vectors produced by previous layer. It is proven effective to improve the performance of our model.

The attention mechanism in our model uses a matching function f to associate the target module with the source module, the function f is implemented as follows:

$$f(m_t, m_s) = {m_t}^T m_s \quad (1)$$

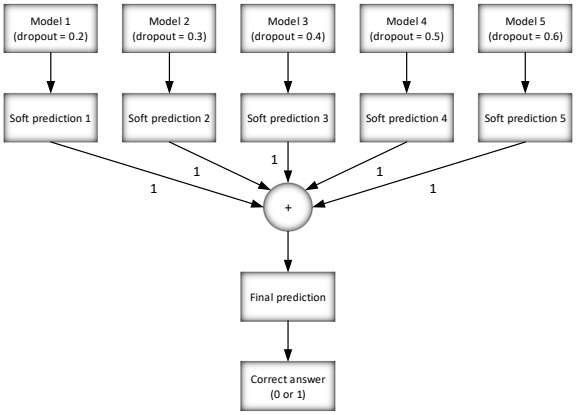

Figure 2: The ensemble method used in our model. The dropout parameters for the five models varied from 0.2 to 0.6 after the BiLSTM layer.

where m_t and m_s correspond to instance and question vectors, or answer and question vectors produced by previous BiLSTM layers, respectively.

2.4 Model Ensemble

Combining multiple models into an ensemble by averaging their predictions is a proven strategy to improve model performance. While predicting with an ensemble is expensive at test time, recent advances in distillation allow us to compress an expensive ensemble into a much smaller model (Hinton et al., 2015; Kuncoro et al., 2016; Kim and Rush, 2016).

In our model, each single model will yield a soft probability to determine if the candidate answer is correct or not. As is shown in figure 2, we train several models and sum the results produced by them. Then we use the sum of the probabilities as the final prediction. We found that the performance of the ensemble model is always better than that of a single model.

3 Experiments

We run each model 10 times, taking the average results as the final experimental results to enhance reliability. In all single models, the dropout parameter is taken as 0.3, and the ensemble model is trained through 5 BiLSTM models. Their dropout parameters are changed from 0.2 to 0.6, respectively, and then the results of the 5 models are summed as the final prediction. We set *epoch* = 6, *batch size* = 512 and *LSTM Units* = 64. Optimization is carried out using Adaptive Moment Estimation (Adam).

3.1 Data Processing

The organizers provided training, development, and test sets, containing 9837, 1417, 2797 questions, respectively. Each question corresponds two answers, only one is correct.

We firstly substitute the abbreviation characters and remove the meaningless characters. Then we combine the instance, question and answer as the supervised training set. Labels are represented by 0 (False) or 1 (True). The *TweetTokenizer*[3] in NLTK is adopted for word segmentation. Furthermore, we find the maximum length of instances is much longer than that of questions and answers, so we remove the stop words in the instances. Our experiments show that doing so not only does not harm the accuracy, but also drastically reduces the training time.

3.2 Experiments and Result Analysis

We compare two word embedding tools, Word2Vec and GloVe, and the experimental results show that GloVe almost always outperforms Word2Vec on this task. Although the vocabularies in GloVe are less than those in Word2Vec, GloVe contains more abbreviations, which are especially useful after tokenizing the instance, question and answer triples, and greatly reduce the number of unknown words in word embedding, making the context semantics better learned by the model. We make random assignments on unknown words, ranging from -0.25 to 0.25.

Tool	Ukw	Time	Dev Acc	Test Acc
GloVe	33	597s	**0.7448**	**0.7276**
Word2Vec	276	40s	0.7415	0.7087

Table 2: Comparison between Word2Vec and GloVe tools on BiLSTM models. Ukw is the number of unknown words. Time is the loading time of two tools. We can see GloVe performs better, but its loading time is much longer than that of Word2Vec.

As seen in Table 3, we compare two network architectures, LSTM and BiLSTM. The results show that the BiLSTM model performs better than the LSTM model on this task.

Based on Glove word embedding and BiLSTM architecture, we train 5 single models for ensemble. The only difference between them is the difference in dropout parameters, which increases from 0.2 to 0.6. In our experiments, we train the

[3]http://www.nltk.org/api/nltk.tokenize.html

Network	Tool	Dev Acc	Test Acc
LSTM	GloVe	0.7448	0.7276
BiLSTM	GloVe	**0.7508**	**0.7301**

Table 3: Comparison between LSTM and BiLSTM. BiLSTM performs better than LSTM on both datasets.

single model with the dropout in order of 0.3, 0.5, 0.4, 0.2, 0.6, then the first ensemble is the result of adding the first two models with the dropout of 0.3, 0.5 as the predictive result, the result of the second ensemble is based on the first ensemble plus the single model with dropout of 0.4, and so on. We perform a total of 4 ensemble experiments, the results show that the accuracy of each ensemble model improved on both datasets. The final ensemble model has an accuracy rate of 0.7699 on the development set and 0.7472 on the test set. However, we find that our model was slightly more accurate on the test set without the ensemble of the model with a dropout of 0.6, but the overall effect is not obvious. Ensemble makes our model perform well on this task, ranking 5th out of 11 submissions.

Dropout	Dev Acc	Test Acc
0.3	0.7476	0.7311
0.5	0.7516	0.7183
Ensemble 1	0.7608	0.7386
0.4	0.7615	0.7294
Ensemble 2	0.7692	0.7408
0.2	0.7354	0.7143
Ensemble 3	0.7664	**0.7479**
0.6	0.7410	0.7308
Ensemble 4	**0.7699**	**0.7472**

Table 4: Results on single and ensemble models. All models adopt GloVe + Attention-based BiLSTM architecture. The dropout layer is behind the BiLSTM layer.

4 Conclusion and Future Work

In this paper, we present an ensemble of attention-based BiLSTM models for machine comprehension task. We find GloVe is superior to Word2Vec on this task, a simple ensemble method can significantly enhance the overall performance.

In the future, we plan to explore more ways to compute the attention, such as a bilinear term. Future work also involves using more external knowledge and deeper network to improve model performance. We will explore the ensemble method in greater depth, trying ensemble on the models with more structural difference.

Acknowledgments

This work was supported by the Natural Science Foundations of China No.61463050, No.617-02443, No.61762091, the NSF of Yunnan Province No. 2015FB113, the Project of Innovative Research Team of Yunnan Province.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Danqi Chen, Jason Bolton, and Christopher D Manning. 2016. A thorough examination of the cnn/daily mail reading comprehension task. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2358–2367.

Bhuwan Dhingra, Hanxiao Liu, Zhilin Yang, William Cohen, and Ruslan Salakhutdinov. 2017. Gated-attention readers for text comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1832–1846.

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.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *stat*, 1050:9.

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

Minghao Hu, Yuxing Peng, and Xipeng Qiu. 2017. Reinforced mnemonic reader for machine comprehension. *CoRR, abs/1705.02798*.

Rudolf Kadlec, Martin Schmid, Ondřej Bajgar, and Jan Kleindienst. 2016. Text understanding with the attention sum reader network. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 908–918.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.

Yoon Kim and Alexander M Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327.

Adhiguna Kuncoro, Miguel Ballesteros, Lingpeng Kong, Chris Dyer, and Noah A Smith. 2016. Distilling an ensemble of greedy dependency parsers into one mst parser. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1744–1753.

Xiaodong Liu, Yelong Shen, Kevin Duh, and Jianfeng Gao. 2017. Stochastic answer networks for machine reading comprehension. *arXiv preprint arXiv:1712.03556*.

Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421.

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.

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*, pages 2383–2392.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 379–389.

Alessandro Sordoni, Philip Bachman, Adam Trischler, and Yoshua Bengio. 2016. Iterative alternating neural attention for machine reading. *arXiv preprint arXiv:1606.02245*.

Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, pages 2440–2448.

Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1422–1432.

Bo-Hsiang Tseng, Sheng-syun Shen, Hung-Yi Lee, and Lin-Shan Lee. 2016. Towards machine comprehension of spoken content: Initial toefl listening comprehension test by machine. *Interspeech 2016*, pages 2731–2735.