Joker at SemEval-2018 Task 12: The Argument Reasoning Comprehension with Neural Attention

[1]Guobin Sui , [1]Wenhan Chao , [2]Zhunchen Luo*
[1]School of Computer Science and Engineering / Beihang University
[2]Information Research Center of Military Science / PLA Academy of Military Science
[1]Beijing 100191, China; [2]Beijing 100142, China
{suigb,chaowenhan}@buaa.edu.cn;zhunchenluo@gmail.com

Abstract

This paper describes a classification system that participated in the SemEval-2018 Task 12: The Argument Reasoning Comprehension Task. Briefly the task can be described as that a natural language "argument" is what we have, with reason, claim, and correct and incorrect warrants, and we need to choose the correct warrant. In order to make fully understand of the semantic information of the sentences, we proposed a neural network architecture with attention mechanism to achieve this goal. Besides we try to introduce keywords into the model to improve accuracy. Finally the proposed system achieved 5th place among 22 participating systems.

1 Introduction

In recent years, as an extremely important part of argument mining, argument reasoning has received considerable research. The argumentation reasoning can be used in many situations such as automatic score, policy decision, stance detection, and many others (Habernal et al., 2018). The task can be described as follows in detail: Given an argument consisting of a claim and a reason, the goal is to select the correct warrant that explains reasoning of this particular argument. There are only two warrants given and only one answer is correct. The correct warrant inferred from the argument has a supported relation with the argument. However the other warrant either opposes the argument or has no correlation with the argument. Actually, this task could be treated as a binary classification(A vs. B).

The task could be regarded as an argumentative relation work. The argumentative relation mining aims at identifying relations of attack and support between natural language arguments in text, by classifying pairs of pieces of text as attack, support or neither attack nor support relations (Cocarascu and Toni, 2017). The corrected warrant means supporting the argument, while the other means attacking the argument. So we refer to some literature methods on the relationship between arguments. Cocarascu and Toni (2017) use Long Short Term Memory (LSTM) to classify the relations between arguments. Rocktäschel et al. (2016) propose neural network with attention mechanism, making neural networks interpretable. We infer to their methods and construct our model in new manner. In this paper, we use the LSTM networks with attention mechanism to construct the classification system.

The following sections are arranged as follows. In section 2, we will give an overview on the task and have an analysis of the datasets. In section 3 we describe the system used in this paper and introduce some interesting attempt in detail. Section 4 introduces the experiment and results. Finally, we get conclusions and have an outlook of the future work.

2 Task Definition

We use the corpus provided by the SemEval-2018 Task 12, which has a training corpus of 2420 samples with gold labels. The organization website also provides a corpus of 316 samples with gold labels as verification set. And finally about 444 samples without gold labels are provided by the organization website as the final test corpus. This data has a variety of contemporary issues across topics in user-generated web comments (Habernal et al., 2018).

For example:

Topic: There She Is, Miss America

Additional info: In 1968, feminists gathered in Atlantic City to protest the Miss America pageant, calling it racist and sexist. Is this beauty contest bad for women?

Argument: Miss America gives honors and education scholarships. And since ..., Miss America is good for women.

Warrant options:

a) scholarships would give women a chance to study

b) scholarships would take women from the home

Only (a) fills the gap in this argument; (b) would in fact lead to the opposite claim (such that Miss America is not good for women).

Observing the data, we find this task has some challenges. We notice that many warrants have high

* Corresponding author.

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 1129–1132
New Orleans, Louisiana, June 5–6, 2018.

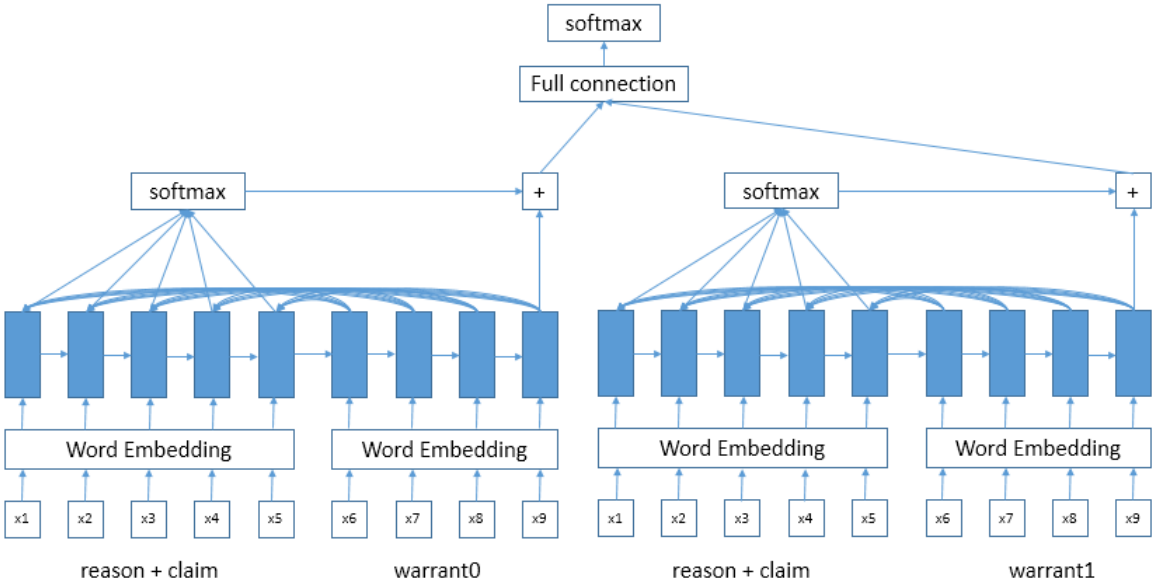

Figure 1: The system architecture - LSTM model with word-by-word attention mechanism.

semantic similarity, which makes it difficult to have a correct choice. In order to distinguish the similar sentences, the model must have the ability to model sentence semantics well. Besides, we find that many pairs of warrants' difference is the negative words. For example, consider the following warrant options from the training corpus. One warrant is "we can't have citizens being loyal to their home country", the other is "we can have citizens being loyal to their home country". The difference of them is the first one has the negative word "n't", which makes the sentences have different semantics. Every sentence has keywords which make a contribution to understand the sentence. We hope that putting the keywords extracted from the sentence into the model could enhance the comprehension of the sentences. We have a try with all these ideas.

3 System description

Our system is based on the LSTM model with word-by-word attention which could been seen as an encoder to decoder model. The encoder encodes the reason and claim and gives the initialization to the decoder. The decoder decodes the warrant and uses its output to compute the weight of tokens from the reason and claim. The higher the weight is, the more important the token is to choose the correct warrant. Besides, we introduce the keywords into the model to improve the accuracy.

3.1 Sequence model

In order to have a full understanding of the sentences, we try to use the neural network to model the sentences. The LSTM model could capture long-term dependencies (Hochreiter and Schmidhuber, 1997) so we use that to construct the sequence model.

LSTM models, a type of RNNs, address the problem of the vanishing gradients problem while trying to capture long-term dependencies by introducing memory cells and gates into networks (Cocarascu and Toni, 2017). Although the LSTM models could solve the problem of the long-term dependencies, it usually capture the words behind the sentence, which causes a problem that the words before the sentence make a little contribution. In order to achieve full comprehension of the sentences, an attention mechanism is introduced into the model. The attention mechanism has been demonstrated success in a wide range of tasks from handwriting synthesis (Graves, 2013), machine translation (Bahdanau et al., 2015) and sentence summarization (Rush et al., 2015). In view of the effectiveness of the attention mechanism, we combine it with the LSTM model expecting a great performance.

Based on the statistics on sentence lengths in training corpus, we set the length of the reason, claim, warrant to 50, 15, 30 respectively. In this paper, we propose three deep learning methods to represent the sentences. The first is that we use BiLSTMs to parse the semantics of the sentences and then merge their output of the BiLSTMs. The concatenated vector is fed into a fully-connected neural network whose output is concatenated with the softmax function to have a prediction. The detailed computation is described in (1-5). The R_i, C_i, $W0_i$ and $W1_i$ are the embedding presentations of the reason, claim, warrant0 and warrant1 respectively and the R_o, C_o, $W0_o$ and $W1_o$ are the outputs of the BiLSTMs. V is the vector connecting

them. This model is the baseline of our paper.

$$R_o = BiLSTMs(R_i) \quad (1)$$
$$C_o = BiLSTMs(C_i) \quad (2)$$
$$W0_o = BiLSTMs(W0_i) \quad (3)$$
$$W1_o = BiLSTMs(W1_i) \quad (4)$$
$$V = [R_o; C_o; W0_o; W1_o] \quad (5)$$

The second and the third method refer to the methods (Rocktäschel et al., 2016) where the author proposed LSTM with attention and word-by-word attention mechanism to solve the problem of reasoning about entailment and achieved the state-of-the-art results. We refer to their model to construct our model. The second and the third method are called LSTM model with attention and LSTM model with word-by-word attention. From the figure 1, there is the LSTM with attention and word-by-word attention mechanism. The two methods see reason and claim as the part of encoder, which gives the initial weight to the warrant and treats the warrant as decoder. The difference between them is that attention is only based on the last output vector of warrant, while the word-by-word attention is based on all output vectors of warrant.

$$Y = BiLSTMs(R_i; C_i) \quad (6)$$
$$H = BiLSTMs(W_i) \quad (7)$$
$$M_t = tanh(W^y Y + (w^h h_t + W^r r_{t-1}) \otimes e_L) \quad (8)$$
$$\alpha_t = softmax(w^T M_t) \quad (9)$$
$$r_t = Y\alpha_t^T + tanh(W^t r_{t-1}) \quad (10)$$
$$h^* = tanh(W^p r_N + W^x h_N) \quad (11)$$

Equation (6-11) provide the details about the computation on word-by-word attention. Y is the output vector of the encoder whose input is embedding representation of reason and claim. H is the output of the decoder with the embedding representation of warrant. h_t is the state of H at time t. We use h_t to compute the weight of the token from reason and claim at every time. Then we can get the weight matrix which means how important the token is for the decoder at every time. The weight matrix of attention is one dimension while the word-by-word attention has n dimensions. So the computation of the attention is simple and the same as the word-by-word attention. The two warrants with the reason and claim are used to construct the word-by-word attention respectively. Then the outputs of them are merged by concatenation and are put into a fully-connected neural network to make a prediction.

3.2 Keywords

We expect that introducing the keywords into the model could improve the accuracy. The sequence model can only express the basic meaning of the sentence and can't grab the main part. So the keywords can semantically enhance the sentence meaning. Based on that, we carry out the keywords extraction using text rank algorithm based on the graph. The specific method is described in the paper (Mihalcea and Tarau, 2004). For getting the keywords, we let G=(V,E) be a undirected graph with the set of vertices V and set of edges E where V is the token corpus and E is the weight between two tokens. The formula could be calculated according to the following equation (12) where the $S(V_i)$ is the score of the vertex V_i, the w is the edges and d is a damping factor that is usually set to 0.85. $Out(V_j)$ and $In(V_i)$ are the adjacent vertices of V_i in undirected graph. Based on the scores we could get the keywords from the sentence. The higher the score is, the more important the word is.

$$S(V_i) = (1-d)+d* \sum_{v_j \in In(V_i)} \frac{w_{ij}}{\sum_{v_k \in Out(V_j)} w_{jk}} S(V_j) \quad (12)$$

According to observing the keywords from the two warrants, we find that some pairs of warrants have the same keywords but different negative word. So we do statistics on negative words. If the number of the negative words is odd, we will add a negative word such as "not" into the keyword corpus. However, if the number of the negative words is even, we won't add any negative word into the keyword corpus. We expect to make a difference between the two warrants with the same keywords. We use the bag-of-word model to model the corpus of the keywords. Every keywords are put into the pre-trained embedding layer to get the word representation. In order to ignore the difference in the number of keywords, we adopt the average operation. The computation is introduced in equation (13-16) where r_i, c_i, $w0_i$ and $w1_i$ are the word representations of reason, claim, warrant0 and warrant1 generated from the Glove vectors (Pennington et al., 2014). The output I_i is then put into a fully-connected neural network to make a classification decision.

$$R_{ave} = \frac{1}{len(R)} \sum_{i=1}^{len(R)} r_i \quad (13)$$
$$C_{ave} = \frac{1}{len(C)} \sum_{i=1}^{len(C)} c_i \quad (14)$$
$$W0_{ave} = \frac{1}{len(W0)} \sum_{i=1}^{len(W0)} w0_i \quad (15)$$
$$W1_{ave} = \frac{1}{len(W1)} \sum_{i=1}^{len(W1)} w1_i \quad (16)$$
$$I_i = [R_{ave}; C_{ave}; W0_{ave}; W1_{ave}] \quad (17)$$

We introduce the keywords into the sequence model and combine the bag-of-word model with the word-by-word attention model. We train the bag-of-word model and incorporate it into LSTM model with word-by-word attention by averaging the predicted probabilities to get the final label to make a correct choice. We call the combination hybrid model. The details of the experiment will be introduced in next section.

Model	dev corpus	test corpus
LSTMs model	0.500	0.504
Attention	0.657	0.571
WBW attention	0.684	0.586
BOW model	0.606	0.524
hybrid model	0.654	0.585

Table 1: The results of the experiments. Attention, WBW attention, BOW model stand for LSTM model with attention, LSTM model with word-by-word attention, bag-of-word model respectively. Training on the training corpus while testing accuracy is computed on the dev corpus and test corpus.

4 Experiment and result

While training the model, the input sentences are separately embedded as 100-dimensional GloVe vectors (Pennington et al., 2014) and the embedding layer is based on the 100-dimensional GloVe vectors. We use ADAM (Kingma and Ba, 2015) for optimization and set the initial learning rate 0.001. We trained for 13 epochs or until the performance on development set stopped improving so as to avoid overfitting. Some Hyper-parameters for model: the dropout is 0.9, the embedding size is 100, the size of LSTM is 64 and the batch size is 256.

We conduct experiment on the training, dev and test corpus downloaded from the SemEval-2018 Task 12. There are five models used to make experiments are LSTM model, LSTM model with attention, LSTM model with word-by-word attention, bag-of-word model and hybrid model. The results of the experiments could be seen in Table 1. We choose the LSTM model without attention as baseline.

According to the experiment, the simple sequence model couldn't complete the semantic understanding task well. The bag-of-word model performs better than the LSTM model, which proves that the keywords could express the semantics of the sentences. As for the keywords can't express all the information of the sentences while the LSTM model with attention can not only express the whole information but also grab the important part, the bag-of-word performs worse than the attention model. LSTM model with word-by-word attention makes a great contribution to the best result. The hybrid model doesn't have an improvement in dev corpus but have a similar results with the word-by-word attention model. We guess that what causes such a result is the small data corpus and simply mechanically combining the models with each other. So we don't get a satisfactory result from the hybrid model. The LSTM model with word-by-word attention gets the accuracy of 0.586 in the final submission, achieving the fifth place in the shared task.

In the future, we will consider more reasonable combinations of the sentence model with keywords to enhance the comprehension of the sentences. Besides, we will introduce the CNN into our model to extract the word character to improve the accuracy.

5 Acknowledgments

We very appreciate the comments from reviewers which will help further improve our work. This work is supported by National Key Research and Development Program of China (Grant No. 2017YFB1402400) and National Science Foundation of China (No. 61602490).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *international conference on learning representations*.

Oana Cocarascu and Francesca Toni. 2017. Identifying attack and support argumentative relations using deep learning. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1385–1390.

Alex Graves. 2013. Generating sequences with recurrent neural networks. *arXiv: Neural and Evolutionary Computing*.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. SemEval-2018 Task 12: The Argument Reasoning Comprehension Task. In *Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018)*, New Orleans, LA, USA. Association for Computational Linguistics.

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

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. *international conference on learning representations*.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into texts. pages 404–411.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomas Ko Isky, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. *international conference on learning representations*.

Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *empirical methods in natural language processing*, pages 379–389.