GIST at SemEval-2018 Task 12: A network transferring inference knowledge to Argument Reasoning Comprehension task

HongSeok Choi, Hyunju Lee
Department of Electrical Engineering and Computer Science,
Gwangju Institute of Science and Technology, Gwangju, Republic of Korea
Data Mining and Computational Biology Laboratory
{hongking9,hyunjulee}@gist.ac.kr

Abstract

This paper describes our GIST team system that participated in SemEval-2018 Argument Reasoning Comprehension task (Task 12). Here, we address two challenging factors: unstated common senses and two lexically close warrants that lead to contradicting claims. A key idea for our system is full use of transfer learning from the Natural Language Inference (NLI) task to this task. We used Enhanced Sequential Inference Model (ESIM) to learn the NLI dataset. We describe how to use ESIM for transfer learning to choose correct warrant through a proposed system. We show comparable results through ablation experiments. Our system ranked 1st among 22 systems, outperforming all the systems more than 10%.

1 Introduction

Argument Reasoning Comprehension is a task that choose correct warrant from two options given a claim and a reason. The Argument Reasoning Comprehension is a very important task because *"argument comprehension requires not only language understanding and logic skills, but it also heavily depends on common sense"*, as mentioned by Habernal et al. (2018). There are two challenging factors. One is a certain part of an argument is left unstated (Habernal et al., 2018). Because of the unstated part, humans or machines need reasoning ability about that part. Human can reconstruct the unstated part depending on common knowledge. However, it has still remained difficult to machines. Another is that *"both options are plausible and lexically very close while leading to contradicting claims"*, as mentioned by Habernal et al. (2018). To address these factors, we have two assumptions. One is that similar and large datasets may help to address the unstated common sense by learning various cases. Another is that an inferrence model to distinguish semantic differences between two sentences may help to choose one of two lexically close warrants that lead to contradicting claims. There are two suitable datasets in the Natural Language Inference (NLI) task, Stanford NLI (SNLI) (Bowman et al., 2015) and Multi NLI (MNLI) (Williams et al., 2017) datasets. NLI is a task choosing one of relationships *(Entailment, Contradiction, Neutral)* between two sentences. Both SNLI and MNLI are very large corpus (each 0.5M sentence pairs). In addition, there is a good performance model for the task, Enhanced Sequential Inference Model (ESIM) (Chen et al., 2017). To make use of other datasets for our task, we use transfer learning. About transfer learning, Conneau et al. (2017) showed a good precedent, using SNLI dataset. By learning the NLI task, the model can obtain inference knowledge. Therefore, we propose a network transferring inference knowledge to argument reasoning comprehension task. We summarize our system with 5 main components.

1. ESIM is trained on SNLI and MNLI datasets. Then, parameters are frozen and used to transfer the inference knowledge.
2. As inputs of the ESIM, we make sentence pairs such as *(claim, warrant)*, *(warrant, reason)* and *(warrant, other warrant)*.
3. To add flexibility, we added biLSTM module encoding *claim*, *reason* and *warrant*.
4. To make a fixed length vector from variable one, we used average and max pooling.
5. Finally, all the fixed length vectors from ESIM and biLSTM are concatenated and fed into a fully-connected neural network to determine whether the warrant is correct or not.

The detail process is described in Section 2.

Proceedings of the 12th International Workshop on Semantic Evaluation (SemEval-2018), pages 773–777
New Orleans, Louisiana, June 5–6, 2018.

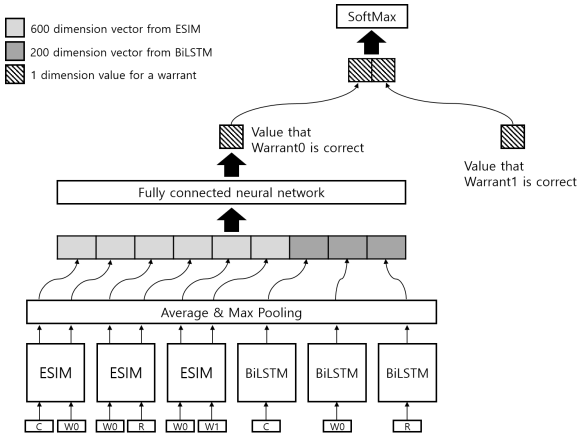

Figure 1: Overview of our system

2 System Description

Figure 1 shows an overview of our system. The preprocessing is described in subsection 2.1. ESIM is described in subsection 2.2. Then, to apply transfer learning, we describe how to compose the inputs of the ESIM in subsection 2.3. In the subsection 2.4, we describe a simple biLSTM module added to our model. Pooling is described in subsection 2.5. Finally, the fully-connected neural network is described in subsection 2.6 to determine whether the warrant is correct or not. We introduce our notations for following sections. A sentence is notated as $S = (w_1^S, ..., w_{len(S)}^S)$. $len(S)$ denotes the length of the sentence S. The $w_i^S \in \mathbb{R}^d$ is a d-dimensional word embeddings. Also C, R, $W0$ and $W1$ denote the sentence of *Claim*, *Reason*, *Warrant0* and *Warrant1* respectively. Our goal is to predict which warrant ($W0$ or $W1$) is more correct given a claim (C) and a reason (R).

2.1 Preprocessing

First, we initialize all words that exist in the vocabulary with pre-trained 300 dimension `word2vec` (Mikolov et al., 2013). When the word does not exist in the vocabulary, we use following several preprocessing rules.

1. All ['s] are removed. (ex. He' s, something's)

2. All words with number are split into number and word. (ex. 17th → 17, th)

3. All abbreviations are replaced with <abbreviation> token.

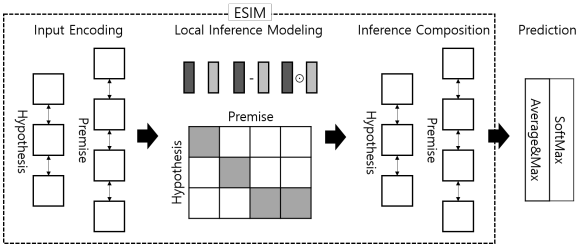

Figure 2: A high-level view of ESIM with prediction part. The prediction part is used when training on the NLI task, and two sentence vectors generated after inference composition are used in our system. This picture is taken from the author (Chen et al., 2017) with a few modifications.

4. All number is replaced with <number> token.

After this preprocessing, if the preprocessed word exists in the vocabulary, we initialized it with the `word2vec` again. Otherwise, we replaced it with <unknown> token. Each token is randomly initialized.

2.2 Pre-trained ESIM on NLI dataset

Because of page limit, we briefly explain this part. Chen et al. (2017) described that ESIM is composed of the following major components: input encoding, local inference modeling, and inference composition. Figure 2 shows a high-level view of the architecture. For more details, refer to the paper (Chen et al., 2017). ESIM generates two sentence vectors after comparing two input sentences with each other. We notate it as follows.

$$\mathbf{sv}_{S_1}^{(S_1,S_2)}, \mathbf{sv}_{S_2}^{(S_1,S_2)} = ESIM(S_1, S_2) \quad (1)$$

The $\mathbf{sv}$ consists of vectors of l dimension, the number of which correspond to the length of each sentence. The $\mathbf{sv}$ is the output of the inference composition part. We implemented it as 300 dimensions. The ESIM was trained on SNLI and MNLI datasets. The training was stopped when the average of development set accuracies was maximum. Then, the parameters were frozen so as to be not updated.

2.3 Input sentence pair for transfer learning

To exploit transfer learning, the sentence pairs are composed of (C, $W0$), ($W0$, R) and ($W0$, $W1$) for Warrant0. In the case of Warrant1, the pairs are composed of (C, $W1$), ($W1$, R) and ($W1$, $W0$). Then, these sentence pairs are fed into the ESIM.

2.4 BiLSTM for Flexibility

LSTM (Hochreiter and Schmidhuber, 1997) is a building block well-suited to learn long and short information in a sequence. We employed bidirectional LSTM, where forward and backward directional LSTMs are concatenated. For more detailes, refer to the paper (Hochreiter and Schmidhuber, 1997).

$$\mathbf{sv}_S = biLSTM(S) \tag{2}$$

We add 100 dimension biLSTMs to our model. Since the ESIM is only trained on NLI dataset, it may be over-fitted to the NLI task. By adding a new module that is not trained on the NLI task, our system may have a chance to learn new knowledge about the target task. We feed *Claim*, *Warrants* and *Reason* into the biLSTM. The biLSTMs for Warrant0 and Warrant1 share the parameters.

2.5 Pooling Layer

To generate a fixed length sentence vector, we use both average and max pooling per one sentence. The equations are as follow:

$$\mathbf{sv}_{S,ave} = \frac{1}{len(S)} \sum_{i=1}^{len(S)} \mathbf{sv}_{S,i} \tag{3}$$

$$\mathbf{sv}_{S,max} = \max_{i=1}^{len(S)} (\mathbf{sv}_{S,i}) \tag{4}$$

After pooling, the vector of average pooling and max pooling are concatenated. We notate it as $\mathbf{sv}_{S,pool} = [\mathbf{sv}_{S,ave}; \mathbf{sv}_{S,max}]$.

2.6 Fully-connected neural network

To determine whether the warrant is correct or not, a fully-connected neural network (FCNN) is used. Finally, all the vectors from ESIM and biLSTM are concatenated. For Warrant0, the vectors are concatenated as follow: $[\mathbf{sv}^{(C,W0)}_{C,pool}; \mathbf{sv}^{(C,W0)}_{W0,pool}; \mathbf{sv}^{(W0,R)}_{W0,pool}; \mathbf{sv}^{(W0,R)}_{R,pool}; \mathbf{sv}^{(W0,W1)}_{W0,pool}; \mathbf{sv}^{(W0,W1)}_{W1,pool}; \mathbf{sv}_{C,pool}; \mathbf{sv}_{W0,pool}; \mathbf{sv}_{R,pool}]$. The Warrant1 is also composed as the same way. The concatenated vector is fed into FCNN. We build two layers of FCNN. The first layer has 600 dimension with the ReLu function. The second layer has only 1 dimension without any activation function. Then, the 1 dimension value for Warrant0 and Warrant1 are concatenated with the softmax function.

3 Experimental setup

Pre-training First, to learn the inference knowledge, we implemented ESIM and trained on NLI training dataset. Our implemented ESIM dimension is 300. Except for the ESIM dimension, we used the same hyperparameter values as those in Chen et al. (2017). The preprocessing process is implemented in the same way with subsection 2.1. The word embeddings are not updated during training. The training was stopped when the average of development set accuracies is maximum. We got development accuracy of 86.58%, 74.09%, 74.67% on SNLI, MNLI match, MNLI mismatch datasets, respectively.

Training We used the ADAM (Kingma and Ba, 2014) optimizer for updating weight parameters. The parameters of ADAM set to be as follow: $\beta 1 = 0.9$, $\beta 2 = 0.999$, $\epsilon = 10^{-8}$. The initial learning rate is 0.0002 and is decayed with 0.9 rate per one epoch. We did not use dropout but added L2 regularization on the first FCNN layer. The regularization parameter λ was set to be 5×10^{-4}. The word embeddings were not updated during training. We randomly shuffled training data during training. The minibatch size was 25. We trained 10 epochs and chose our model when the development set reached the max accuracy. We implemented our system by using lasagne (Dieleman et al., 2015) and theano (Theano Development Team, 2016) library. Our code is available at here[1].

4 Results and Discussion

To get more reliable results, all accuracies were calculated by averaging after repeating ten experiments. In this competition, the official accuracy of our system recorded 0.712 on test set. Table 1 shows accuracies of other approaches and ours on Argument Reasoning Comprehension task. Our approach showed best performance except human, outperforming all the systems more than 10%. Table 2 shows the results of ablation experiments. Model (a) is our proposed system. Model (d) indicates a model that is same as the model (a), except that the inference knowledge is not transferred. This model is directly trained on our task. Models (b) and (e) indicate that the modules including warrant pair inputs $(W0, W1)$ and $(W1, W0)$ are removed from model (a) and (d), respectively.

[1] https://github.com/hongking9/SemEval-2018-task12

Approach	Dev	Test
Human average	-	0.798
Human w/ training in resoning	-	0.909
Our system	**0.716** $\pm$ 0.006	**0.711** $\pm$ 0.007
Random baseline	0.473	0.491
2nd ranked system	-	0.606
Attention†	0.488	0.513
Attention w/ context †	0.502	0.512
Intra-warrant attention†	0.638	0.556
Intra-warrant attent. w/ context†	0.637	0.560

Table 1: Accuracy of each approch. The human and baseline results are taken from Habernal et al. (2018). Our approach ranked 1st among 22 systems, outperforming all the systems more than 10%. † indicates approaches implemented by Habernal et al. (2018). Readers can check all system results at here[2].

Model	Dev	Test
(a) Our system	0.716	**0.711**
(b) − warrants pair input	0.685	0.696
(c) − biLSTM	**0.726**	0.706
(d) No Transferring	0.652	0.599
(e) − warrants pair input	0.653	0.605
(f) − biLSTM	0.656	0.608

Table 2: Ablation experiments.

Models (c) and (f) indicate that the biLSTM module is removed from model (a) and (d), respectively.

As we mentioned above introduction section, our proposed model addresses two challenging factors. The one is about common sense and the other is about the two lexically close warrants that lead to contradicting claims.

Transfer learning First, by comparing models (a) and (d), we can observe that the inference knowledge of the NLI task is very helpful to the argument reasoning comprehension task. We may assume that machine can accommodate common sense by learning similar tasks and large corpus.

Warrants pair input Second, by comparing models (a) and (b), we can observe that the warrants pair input result in improved performance. We may infer that the model can distinguish fine difference of the two warrants well by directly feeding the warrant pair. However, in the case of model (d) and (e), there is no sufficient difference of the performance. We think this is because the model did not learn to infer the relationship between two sentences.

Adding biLSTM Finally, by comparing (a) and (c), we can observe that adding biLSTM results in a little improved performance on the test set. Also, the performance on the test set was nearly similar with those in development set in model (a) whereas there was 2% difference in model (c). We carefully infer that the performance on the development set was more reliable and the model becomes flexible to the target task when adding not-trained module to pre-trained and frozen model. However, since the data is not large enough to prove it, we leave it as future work.

5 Conclusion

To address argument reasoning comprehension task, we proposed a network transferring inference knowledge to the Argument Reasoning Comprehension task. First, we implemented ESIM and trained it on a large NLI task corpus. We took full advantage of the model to transfer inference knowledge of NLI task, appropriately building the network. Our approach showed robustness on this task. Through ablation experiment, we showed the following effects: transfer learning, warrants pair input, and adding biLSTM. Also, we showed our system can address the factors about common sense and two lexically close warrants that lead to contradicting claims.

Acknowledgments

This research was supported by the Bio-Synergy Research Project (NRF-2016M3A9C4939665) of the Ministry of Science, ICT and Future Planning through the National Research Foundation.

References

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Enhanced lstm for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1657–1668.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from

[2]https://github.com/habernal/semeval2018-task12-results

natural language inference data. *arXiv preprint arXiv:1705.02364*.

Sander Dieleman, Jan Schlter, Colin Raffel, Eben Olson, Sren Kaae Snderby, Daniel Nouri, et al. 2015. Lasagne: First release.

Ivan Habernal, Henning Wachsmuth, Iryna Gurevych, and Benno Stein. 2018. The argument reasoning comprehension task: Identification and reconstruction of implicit warrants. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, page (to appear), New Orleans, LA, USA. Association for Computational Linguistics.

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

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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.

Theano Development Team. 2016. Theano: A Python framework for fast computation of mathematical expressions. *arXiv e-prints*, abs/1605.02688.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.