World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:10, No:12, 2016

A Sentence-to-Sentence Relation Network for Recognizing Textual Entailment

Isaac K. E. Ampomah, Seong-Bae Park, Sang-Jo Lee

***Abstract*—**Over the past decade, there have been promising developments in Natural Language Processing (NLP) with several investigations of approaches focusing on Recognizing Textual Entailment (RTE). These models include models based on lexical similarities, models based on formal reasoning, and most recently deep neural models. In this paper, we present a sentence encoding model that exploits the sentence-to-sentence relation information for RTE. In terms of sentence modeling, Convolutional neural network (CNN) and recurrent neural networks (RNNs) adopt different approaches. RNNs are known to be well suited for sequence modeling, whilst CNN is suited for the extraction of n-gram features through the filters and can learn ranges of relations via the pooling mechanism. We combine the strength of RNN and CNN as stated above to present a unified model for the RTE task. Our model basically combines relation vectors computed from the phrasal representation of each sentence and final encoded sentence representations. Firstly, we pass each sentence through a convolutional layer to extract a sequence of higher-level phrase representation for each sentence from which the first relation vector is computed. Secondly, the phrasal representation of each sentence from the convolutional layer is fed into a Bidirectional Long Short Term Memory (Bi-LSTM) to obtain the final sentence representations from which a second relation vector is computed. The relations vectors are combined and then used in then used in the same fashion as attention mechanism over the Bi-LSTM outputs to yield the final sentence representations for the classification. Experiment on the Stanford Natural Language Inference (SNLI) corpus suggests that this is a promising technique for RTE.

***Keywords*—**Deep neural models, natural language inference, recognizing textual entailment, sentence-to-sentence relation.

I. Introduction

UNDERSTANDING the semantic relationship between two sentences is a key for improving performance in several in NLP tasks such as Automatic Text Summarization, Question and Answering, Machine Translation etc. Given a premise *P* and a hypothesis *H*, RTE is the task of determining whether the meaning of *H* can be inferred from the meaning of the *P* (i.e. *P* entails *H*), or if they contradict each other, or no relation exists between them.

Over the past decade, approaches to RTE range from approaches based on lexical similarities and models based on formal reasoning to advanced methods that consider also the syntax information [1]. Also [2] performed explicit sentence alignment on the premise and the hypothesis. The lexical based models utilize machine learning techniques but require extensive human engineering to represent the lexical and syntactic information in the pairs. Under the formal reasoning paradigm, the sentences are converted into their formal logical representations which are then analyzed by interpreters for a proof [3].

Recently, deep neural models have been showed to yield better results in NLP applications such as sentence/ sequence modeling [4] and matching models [5]. The core of the deep neural models applied in RTE is a sentence encoding module which generates the semantic representation of both the premise and the hypothesis. Several forms of matching layers have also been used on top of the sentence encoder to capture the interaction of the two sentence representations.

The sentence encoders employed include LSTM/GRUs-based models [8]-[10], SPINN [6], TBCNN [7]. From [8]-[10], Bidirectional LSTMs/GRUs were showed to be powerful in capturing the contextual information of the input sentences. [8] achieved an accuracy of 77.6% on the SNLI corpus when LSTM was used to obtain the semantic sentence representations. Another work [9] showed that, with the addition of neural attention mechanism, the model performance of NLI systems can be improved. The base of their model was still a LSTM but they added an attention layer which considered the alignment between the premise and the hypothesis.

As a contribution to the task of RTE, we propose a model that draws its motivation from earlier works such as [9]-[10] but also utilizes a sentence-to-sentence relation metric computed in between layers in the network to capture how sentence relate with each other. The base of our model is a convolutional phrase extraction network and a Bi-LSTM sentence encoder. We then extend the base model with the relations information to generate the final representations for both the premise and the hypothesis. This approach seeks to generate a more accurate sentence representations at the same time capturing their relations for the classification task.

Section II introduces our network that models the two sentences in parallel whilst computing their relations at the various stages. Section III evaluates our model on the SNLI task, and finally the conclusion is drawn in Section IV.

Isaac. K. E. Ampomah is with the School of Computer Science and Engineering, Kyungpook National University, Daegu, South Korea (phone: +8210-4677-0820; e-mail: kojo@ sejong.knu.ac.kr).

Seong-Bae Park and Sang-Jo Lee are professors at the School of Computer Science and Engineering, Kyungpook National University, Daegu, South Korea (e-mail: seongbae @knu.ac.kr, sjlee@knu.ac.kr).

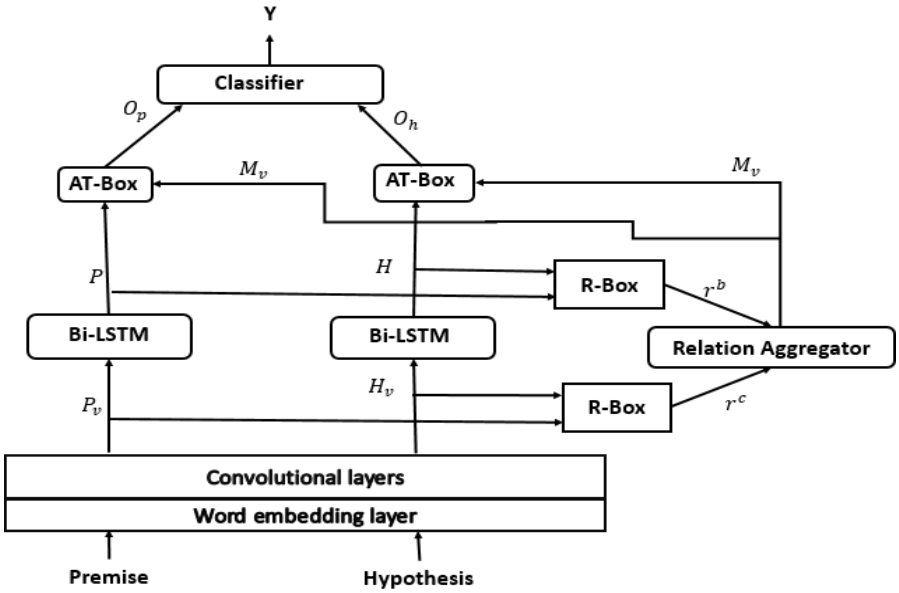

Fig. 1 Illustration of our model

II. Approach

In this section, we describe our architecture shown in Fig. 1. Basically, it is made up of the sentence encoder, a Sentence Interaction/Relation module. The sentence encoder connections are done in a Siamese Network manner where two identical networks share the same weights during training. Following [8]-[10], we also consider the RTE task as a three-way classification problem.

A. Background

1. LSTM

RNNs with LSTM [11] since its inception have been successfully applied to NLP tasks such as Automatic Text Summarization [12], Machine Translation [4]. LSTMs consist of memory cells for information storage at each time step *t* as well as gates (input gates *i*, the forget gates *f*, and the output gate *o*) to control the flow of information within the memory cells. LSTM can be formalized as:

$$i_t = \sigma(W_i x^t + U_i h(t-1) + b_i) \quad (1)$$

$$f_t = \sigma\left(W_f x^t + U_f h(t-1) + b_f\right) \quad (2)$$

$$C^* = tanh\left(W_c x^t + U_c h(t-1) + b_c\right) \quad (3)$$

$$C_t = f_t \odot c_{t-1} + i_t \odot C^* \quad (4)$$

$$o_t = \sigma(W_o x^t + U_o h(t-1) + b_o) \quad (5)$$

$$h_t = o_t \odot tanh(C_t) \quad (6)$$

where $\odot$ is the element-wise multiplication, σ denotes the element-wise application of the sigmoid function. f_t is the forget gate, i_t is the input gate and o_t the output gate. C_t is the memory cell state, C^* is the candidate information to be added to the memory cell state C_t. Also, the representation at time *t* with context information is the *ht*. $W_i, W_f, U_i, U_f, W_c, U_c \in R^{2k \times k}$ are the trained matrices, and $b_i, b_f, b_i, b_f, b_c, b_c \in R^k$ **are** the trained biases for the parameterizations and the transformations of the input.

2. Convolutional Neural Network (CNN)

The convolutional network employed in this work simply computes a 1-D convolution between its input and output. This involves sliding a filter vector over the input sequence and detecting the phrase features at different positions. Therefore, given an input *Zp*, the output of the convolution with filter of size *f* is computed as:

$$Z^* = W^c Z_p + b_c \quad (7)$$

where W^c, b_c are the parameters to be learned. The output of the convolution *Z** contains features extracted in a context window around the individual columns in *Zp*. The number of convolutional filters and the filter sizes are considered hyper-parameters which can be optimized during training. Different forms of pooling are often applied to the output of the convolution to enhance the generated feature maps. But, in this work, because the output of the CNN is feed into to the Bi-LSTM layer, no pooling is applied. This is because applying pooling such max-pooling breaks the sequence organization which affects the performance of any sequence-based encoder such as Bi-LSTM.

B. Sentence Encoder (SE)

The sentence encoder consists of a convolutional neural CNN layer and a Bi-LSTM layer. The CNN layer performs sentence phrasal (N-gram) feature extraction via convolutions. To generate multiple feature maps, we use multiple filters of different length.

As shown in studies [8]-[10], Bi-LSTMs/GRUs are powerful in capturing the semantic meaning of text. We

adopted a Siamese Bi-LSTMs for the sentence encoding to reduce the number of model parameters. The input to this layer is the output of the CNN.

Given a premise representation *Pv* and a hypothesis representation *Hv* from the CNN layer, the Bi-LSTMs encode the *Pv* into its semantic contextual representation *P* and the *Hv* into *H*. Adding an average/mean pooling layer over the outputs the Bi-LSTM further enriches the semantic representation of sentences.

Bi-LSTM is preferred to single directional LSTM as LSTMs suffer a weakness of not utilizing the contextual information from future tokens. Bi-LSTM utilizes both the previous and the future tokens/contexts by processing the sequences in two directions, generating a new sequence for the forward direction and another for the reversed/backward direction. These are then concatenated to produce the distributed representation of the input sequence.

C. Sentence Interaction/Relation Module

This module consists of Relation Boxes (R-Boxes) for relation vector computation, a fully connected layer called the Relation Aggregator, another fully connected layer known as AT-BOX which conditions the outputs of the Bi-LSTM on the output of Relation Aggregator. The R-Boxes take as input the output of the CNN layer (Pv and Hv) and Bi-LSTM layer output (P and H) to computed the relation vectors r^c and r^b respectively. The relation between the premise and the hypothesis during the encoding stages is computed using:

a. Concatenation of the representation of the premise and the hypothesis
b. Element-wise product ($P_V \odot H_V$) or ($P \odot H$)
c. Element-wise difference ($Pv - Hv$) or ($P - H$)

$$\mathbf{r^c} = \begin{vmatrix} P_v \\ P_v \odot H_v \\ H_v \\ P_v - H_v \end{vmatrix} \tag{8}$$

$$\mathbf{r^b} = \begin{vmatrix} P \\ P \odot H \\ H \\ P - H \end{vmatrix} \tag{9}$$

The computation of r^c and r^b was suggested by [13]. But, instead of using the outputs of the final stage of the sentence encoders, we used the sentence representations P_V and H_V obtained from CNN layer and the *P* and *H* from the Bi-LSTM layer of the encoder.

The Relation Aggregator layer yields the matching vector *Mv* which is the weighted sum of the outputs of the R-Boxes output vectors r^c and r^b. *Mv* is computed as:

$$M_v = (W^b r^b + W^{cv} r^c) \tag{10}$$

where W^b, $W^{cv} \in R^{d*d}$ are the trained parameters in the Relation Aggregator layer. Therefore, given the output of the sentence encoder $S^* \in [P, H]$ and the Relation Aggregator matching output Mv, we generated the conditional sentence representation from the AT-Boxes as:

$$h_A = tanh(W^s S^* + W^m Mv) \tag{11}$$

$$\alpha = softmax(W^T h_A) \tag{12}$$

$$O_s = S^* \alpha^T \tag{13}$$

where $O_s \in [O_P, O_h]$ is the final sentence representation of the premise and the hypothesis**.** Also, $W^s, W^m \in R^{k*d}$ are the trained projection matrices, $W \in R^k$ is a trained parameter, and W^T is its transpose. The generated conditional representations Op, Oh and their element-wise product (Op ⊙ Oh) are then concatenated and fed into the classifier which consists of an n-layers of 600D ReLU layers and then finally a Softmax layer which produces a distribution over the class labels. The number of ReLU layers are chosen as hyper-parameter and tuned during training. We use batch normalization [14] to improve the training speed of the network as it reduces the covariate shift. Additionally, for model regularization, we use L2 regularization and dropout [15].

III. Experiments

A. Dataset and Parameters

We evaluated our model using the SNLI corpus [8] which has attracted a lot of attention as it makes it possible to apply deep learning methods to solve RTE problems. The SNLI is a corpus of 570k human labeled pairs of sentences as either Entailment, Contradiction or Neutral or –, where – indicates a lack of consensus from human annotators (unlabeled examples).

Following [8], we use the standard train, validation, and test splits, discarding the unlabeled sentence pairs. In the end, we have 549,367 pairs for training, 9,842 pairs for development and 9,824 pairs for testing. No hand-crafted features are used in our model.

TABLE I
Comparison of Our Model Performance to Others Models

Model	Test Accuracy (%)
Lexicalized Classifier [8]	78.2
LSTM [8]	77.6
LSTM shared [9]	81.4
Word-by-word attention [9]	**83.5**
Our Model	81.13

Our model training objective is cross entropy loss, a batch-size of 512 and Adam optimizer [16] with a learning rate of 1E-3. Dropouts are applied in between layers with dropout rate chosen randomly between the range 0.1 and 0.9. We also use pretrained 300D Glove 840B vectors [17] to initialize the word embeddings. Following [10], out-of-vocabulary words in the training set are randomly initialized by sampling values uniformly from -0.05 and 0.05. We do not update the word embedding during training. For the CNN layer, multiple filters

of length 1, 2, 3, 4 are used. We choose the number of filters for each filter length as 150.

B. Results and Discussion

We evaluated our proposed model on the SNLI dataset as described in Section III *A*. The results are summarized in the Table I. The results are expressed in terms of accuracy. In Table I, our model achieved a test accuracy of 81.13%, which is less than that of some previous works. But, we still find these results promising as we were not able perform exhaustive optimization of the hyper-parameters due to computational resources. We hope that more parameter optimization will lead to higher accuracy.

Contrary to the previous works that utilized usually the last output of the Bi-LSTM/GRUs or RNNs, to compute the attention weighted sentence representations, we adopted an approach that utilizes a Relation Aggregator layer to aggregate the relation vectors computed at different stages during the sentence encoding which is then used for the conditioning.

This approach was adopted to generate the final representation of the premise and hypothesis capturing how the sentences relates with each other. And also, we seek to investigate the possibility of computing attention weighted representation of sentences using relation vectors generated from another related network. The accuracy achieved was surprisingly higher than expected.

IV. Conclusion

In this paper, we introduce a model that exploits the relations between the premise and the hypothesis to solve the RTE problem. We compute the relation vector which is then combined with the output of the sentence encoder to generate the final sentence representations. This method generates sentence representation with information from its relations to the other sentence under consideration.

Experiments on the SNLI dataset showed a promising result (81.13% test accuracy) which can be improved with exhaustive optimization of the hyper-parameters. In the future, we hope to:

1. Experiment with other techniques in the computation of sentence relations.
2. Perform exhaustive parameter optimizations on the model to ascertain the best value of hyper-parameters to enhance performance.
3. Employ this architecture on other task such as question and answering and other NLI tasks.

Acknowledgment

This work was supported by ICT R&D program of MSIP/IITP. [R0126-15-1117, Core technology development of the spontaneous speech dialogue processing for the language learning].

References

[1] Y. Mehdad, A. Moschittil, and F. Massiomo, "Semker: Syntactic/Semantic Kernels for Recognizing Textual entailment," In *Proceedings of the Text Analysis Conference*, pp. 259-265, 2009.

[2] B. MacCartney, M. Galley, and C. D. Manning, "A Phrase-Based Alignment Model for Natural Language Inference," In *Proceedings of the Empirical Methods in Natural Language Processing*, pp. 802-811, 2008.

[3] E. Lien, and M. Kouylekov, "Semantic Parsing for Textual Entailment," In Proceedings of the 14th International Conference on Parsing Technologies, pp. 40–49, 2015.

[4] N. Kalchbrenner, E. Grefenstette, and P. Blunsom, "A Convolutional Neural Network for Modelling Sentences," In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics,* pp. 655–665, 2014.

[5] B. Liu, M. Huang, S. Liu X. Zhu and X. Zhu, "A Sentence Interaction Network for Modeling Dependence between Sentences," In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics,* pp.558–567, 2016.

[6] S. R. Bowman, J. Gauthier, A. Rastogi, R. Gupta, C. D. Manning and C. Potts, "A Fast Unified Model for Parsing and Sentence Understanding," *arXiv:1603.06021v3*, 2016.

[7] L. Mou, H. Peng, G. Li, Y. Xu, L. Zhang, and Z. Jin, "Discriminative neural sentence modeling by tree-based convolution," In *Proceedings of the Empirical Methods in Natural Language Processing,* pp. 2315-2325, 2015.

[8] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A Large Annotated Corpus for Learning Natural Language Inference," In *Proceedings of the Empirical Methods in Natural Language Processing*, pp. 632-642, 2015.

[9] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kocisky, and P. Blunsom, "Reasoning about Entailment with Neural Attention," In *arXiv preprint arXiv: 1509.06664*, 2015.

[10] Y. Liu, C. Sun, L. Lin, and X. Wang, "Learning Natural language Inference using Bidirectional LSTM model and Inner-Attention," *arXiv:1605.09090v1*, 2016.

[11] S. Hochreiter, and J. Schmidhuber, "Long short-term memory," *Neural computation* No. 9, Vol. 8, pp. 1735-1780, 1997.

[12] A. M. Rush, S. Chopra, and J. Weston, "A Neural Attention Model for Sentence Summarization," In *Proceedings of the Empirical Methods in Natural Language Processing*, pp. 379–389, 2015.

[13] L. Mou, M. Rui, G. Li, Y. Xu, L. Zhang, R. Yan, and Z. Jin, "Recognizing Entailment and Contradiction by Tree-Convolution," *arXiv: 1512.08422*, 2015.

[14] S. Ioffe, and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," In *arXiv preprint arXiv: 1502.03167*, 2015.

[15] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," In *Journal of Machine Learning Research No.15 vol. 1,* pp. 1929-1958, 2014.

[16] D. Kingma, J. Ba, "Adam: A Method for Stochastic Optimization," arXiv: 1412.6980, 2014.

[17] J. Pennington, R. Socher and C. D. Manning, "Glove: Global Vector for Word Representation," In *Proceedings of the Empirical Methods in Natural Language Processing*, pp. 1532-1543, 2014.