Syntax-Aware Attention for Natural Language Inference with Phrase-Level Matching

Mingtong Liu, Yasong Wang, Yujie Zhang, Jinan Xu, and Yufeng Chen

School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China
{16112075,18120467,yjzhang,jaxu,chenyf}@bjtu.edu.cn

Abstract. Natural language inference (NLI) aims to predict whether a premise sentence can infer another hypothesis sentence. Models based on tree structures have shown promising results on this task, but the performance still falls below that of sequential models. In this paper, we present a syntax-aware attention model for NLI, by which phrase-level matching between two sentences is allowed. We design tree-structured semantic composition function that builds phrase representations according to syntactic trees. We then introduce cross sentence attention to learn interaction information based on phrase-level representations between two sentences. Moreover, we additionally explore a self-attention mechanism to enhance semantic representations by capturing the context from syntactic tree. Experimental results on SNLI and SciTail datasets demonstrate that our model has the ability to model NLI more precisely and significantly improves the performance.

Keywords: Natural language inference · Syntax-aware attention · Tree-structured semantic composition · Phrase-level matching.

1 Introduction

Natural Language Inference (NLI) is a core challenge for natural language understanding [18]. More specifically, the goal of NLI is to identify the logical relationship (*entailment*, *neutral*, or *contradiction*) between a premise and a corresponding hypothesis. Recently, neural network-based models for NLI have attracted more attention for their powerful ability to learn sentence representation [1, 25]. There are mainly two class of models: sequential models [20, 25, 13, 9, 24, 26, 7] and tree-structured models [2, 3, 27, 19, 14].

For the first class of models, sentences are regarded as sequences, in which word-level representation is usually used to model interaction between the premise and hypothesis with attention mechanism [20, 25, 7]. These models make no consideration of the syntax, but syntax has been proved to important for natural language sentence understanding [4, 5]. Since the compositional nature of sentence, the same words may produce different semantics because of different word orders or syntactic structures, as shown in Fig. 1. The sentences in Fig. 1(a) have same words but different word orders, and express different meaning. The sentences in Fig. 1(b) have same word orders but different syntactic structures.

Show me the fights from **New York** to **Florida**. Show me the fights from **Florida** to **New York**.

(a) Sentences with same words but different word orders.

(b) Sentences with same word orders but different syntactic structures.

Fig. 1. The examples that are difficult for sequential structure to understand.

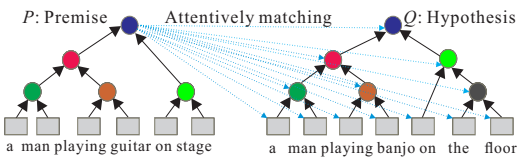
On the left, "with a telescope" is combined with man, and express that "I saw the man who had a telescope". On the right, "with a telescope" provides additional information about the action "saw the man", and express that "I used the telescope to view the man". Thus, for these language expressions with subtle semantic changes, the sequential models can not always work better than tree-structured models, and syntax is still worth of a further exploration.

For the second class of models, tree structures are used to learn semantic composition [2, 3, 27, 19], in which leaf node is word representation and non-leaf node is phrase representation. The final representation of root node is regarded as sentence representation. Recent evidence [23, 8, 3, 19] reveals that tree-structured models with attention can achieve higher accuracy than sequential models on several tasks. However, the potential of the tree-structured network has not been well exploited for NLI, and the performance of tree-structured models still falls below complex sequential models with deeper networks.

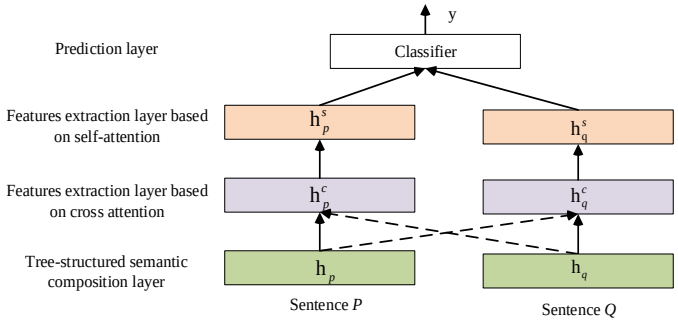
To further explore the potential of tree structure for improving semantic computation, we propose a syntax-aware attention model for NLI, as shown in Fig. 2. It mainly consists of three sub-components: (1) tree-structured composition; (2) cross attention; and (3) self-attention. The tree-structured composition uses syntactic tree to generate phrase representations. Then, we design cross attention to model phrase-level matching that learns interaction between two sentences. A self-attention mechanism is also introduced to enhance semantic representations, which captures the context from syntactic tree within each sentence.

In summary, our contributions are as follows:

- We propose a syntax-aware attention model for NLI. It learns phrase representatiosn by tree-structured composition based on syntactic structure.
- We introduce phrase-level matching with cross attention and self-attention mechanism. The cross attention makes interaction between two sentences, and the self-attention enhances semantic representations by capturing the context from syntactic tree within each sentence.
- We evaluate the proposed model on SNLI and SciTail datasets and the results show that our model has the ability to model NLI more precisely than the previous sequential and tree-structured models.



(a) Tree-structured semantic composition and attention-based phrase-level matching.



(b) An overview architecture of our proposed model.

Fig. 2. (a): It learns tree-structured semantic composition in which non-leaf nodes are composed following syntactic tree and represent phrase representations. Then syntax-aware attention is performed for phrase-level matching between two sentences. (b): Based on phrase representations composed syntactically, cross attention and self-attention mechanism are performed to extract features. Finally, a classifier is used to predict the semantic relation between the two sentences.

2 Related Works

Previous work [22, 17, 23] reveals that models using syntactic trees may be more powerful than sequential models in several tasks, such as sentiment classification [23], neural machine translation [8]. For NLI task, Bowman et al. [2] use constituency parser tree, and explore tree-structured Tree-LSTM to improve sequential LSTM. This method is simple and effective, but ignores the interaction between two sentences. Munkhdalai and Yu [19] use full binary tree, and introduce attention mechanism to model the interaction between two sentences by using node-by-node matching. More recently, Chen et al. [3] design enhanced Tree-LSTM. It shows that incorporating tree-structured information can further improve model performance, and the constituency parser tree is more effective than full binary tree. The latent tree structure [27] is also used to improve semantic computation. However, the existing tree-structured models still fall below complex sequential models [3, 25, 24, 9, 7].

In this paper, we focus on how to use syntactic structure to improve semantic computation for complex language understanding. We propose a syntax-aware attention model for NLI, which explores tree-structured semantic composition

and implements attention-based phrase-level matching between premise and hypothesis. Experimental results demonstrate the effect of the proposed model.

3 Approach

The model takes two sentences P and Q with syntactic trees as input. Let $P = [p_1, \cdots, p_i, \cdots, p_m]$ with m words and $Q = [q_1, \cdots, q_j, \cdots, q_n]$ with n words. The goal is to predict label y that indicates the logic relation between P and Q. In this paper, we focus on learning semantic composition over constituency tree. We give an example of binarized constituency parser tree in Fig. 2(a).

3.1 Tree-Structured Composition

We apply tree-structured composition for P and Q. In our model, each non-leaf node has two children nodes: leaf child l and right child r. We initiate leaf nodes with BiLSTM [12]. For non-leaf nodes, we adopt S-LSTM [28] as composition function. Each S-LSTM unit has two vectors: hidden state h and memory cell c.

Let (h_t^l, c_t^l) and (h_t^r, c_t^r) represent the left child node l and the right child node r, respectively. We compute a parent node hidden state h_{t+1}^p and memory cell c_{t+1}^p as following equations.

$$i_{t+1} = \sigma(W_{hi}^l h_t^l + W_{hi}^r h_t^r + W_{ci}^l c_t^l + W_{ci}^r c_t^r + b_i) \tag{1}$$

$$f_{t+1}^l = \sigma(W_{hf_l}^l h_t^l + W_{hf_l}^r h_t^r + W_{cf_l}^l c_t^l + W_{cf_l}^r c_t^r + b_{f_l}) \tag{2}$$

$$f_{t+1}^r = \sigma(W_{hf_r}^l h_t^l + W_{hf_r}^r h_t^r + W_{cf_r}^l c_t^l + W_{cf_r}^r c_t^r + b_{f_r}) \tag{3}$$

$$u_{t+1} = \tanh(W_{hu}^l h_t^l + W_{hu}^r h_t^r + b_u) \tag{4}$$

$$c_{t+1} = f_{t+1}^l \odot c_t^l + f_{t+1}^r \odot c_t^r + i_{t+1} \odot u_{t+1} \tag{5}$$

$$h_{t+1} = o_{t+1} \odot \tanh(c_{t+1}) \tag{6}$$

where σ denotes the logistic sigmoid function and $\odot$ denotes element-wise multiplication of two vectors; f_l and f_r are the left and right forget gate; i, o are the input gate and output gate; W and b are learnable parameters, respectively.

We use the hidden state of node as phrase representation. Then, two sentences are represented by $h_p = [h_{p_1}, \cdots, h_{p_i}, \cdots, h_{p_{2m-1}}]$ and $h_q = [h_{q_1}, \cdots, h_{q_j}, \cdots, h_{q_{2n-1}}]$. It noted that there are m-1/n-1 non-leaf nodes composed from the tree for phrase representations and m/n leaf nodes for word representations for P/Q,

3.2 Cross Attention

Cross attention is utilized to capture the phrase-level relevance between two sentences. Give two composed representations based on syntactic trees h_p and h_q for P and Q, we first compute unnormalized attention weights A for any pair of nodes between P and Q with biaffine attention function [6] as follows:

$$A_{ij} = {h_{p_i}}^T W h_{q_j} + \langle U_l, h_{p_i} \rangle + \langle U_r, h_{q_j} \rangle \tag{7}$$

where $W \in \mathbb{R}^{h \times h}$, $U_l \in \mathbb{R}^h$, $U_r \in \mathbb{R}^h$ are learnable parameters, and $\langle \cdot, \cdot \rangle$ denotes the inner production operation. p_i and q_j are the i-th and j-th node in P and Q, respectively. Next, the relevant semantic information for nodes p_i and q_j in another sentence is extracted as follows:

$$\widetilde{h}_{p_i} = \sum_{j=1}^{2n-1} \frac{exp(A_{ij})}{\sum_{k=1}^{2n-1} exp(A_{ik})} h_{q_j} \tag{8}$$

$$\widetilde{h}_{q_j} = \sum_{i=1}^{2m-1} \frac{exp(A_{ij})}{\sum_{k=1}^{2m-1} exp(A_{kj})} h_{p_i} \tag{9}$$

Intuitively, the interaction representation $\widetilde{h}_{p_i}$ is a weighted summation of $\{h_{q_j}\}_{j=1}^{2n-1}$ that is softly aligned to h_{p_i}, and the semantics of h_{q_j} is more probably selected if it is more related to h_{p_i}.

To further enrich the interaction, we use a local comparison function ReLU [10].

$$\overline{h}_{p_i} = [h_{p_i}; \widetilde{h}_{p_i}; |h_{p_i} - \widetilde{h}_{p_i}|; h_{p_i} \odot \widetilde{h}_{p_i}] \tag{10}$$

$$h^c_{p_i} = \mathrm{ReLU}(W_p \overline{h}_{p_i} + b_p) \tag{11}$$

$$\overline{h}_{q_j} = [h_{q_j}; \widetilde{h}_{q_j}; |h_{q_j} - \widetilde{h}_{q_j}|; h_{q_j} \odot \widetilde{h}_{q_j}] \tag{12}$$

$$h^c_{q_j} = \mathrm{ReLU}(W_q \overline{h}_{q_j} + b_q) \tag{13}$$

where W, b are learnable parameters. This operation helps the model to further fuse the matching information, and also reduce the dimension of vector representations for less model complexity.

After that, nodes p_i and q_j in P and Q are newly represented by $h^c_{p_i}$ and $h^c_{q_j}$, respectively.

3.3 Self-Attention

We introduce a self-attention layer after cross attention. It captures context from syntactic tree for each sentence and enhances node semantic representations.

For sentence P, we first compute self-attention weights S as equation (7).

$$S_{ij} = \langle h^c_{p_i}, h^c_{p_j} \rangle \tag{14}$$

where, S_{ij} indicates the relevance between the i-th node and j-th node in P. Then, we compute the self-attention vector for each node in P as follows:

$$\widetilde{h}^c_{p_i} = \sum_{j=1}^{2m-1} \frac{exp(S_{ij})}{\sum_{k=1}^{2m-1} exp(S_{ik})} h^c_{p_j} \tag{15}$$

Intuitively, $\widetilde{h}^c_{p_i}$ augments each node representation with global syntactic context from P also from Q.

Similarly, we compute self-attention vector $\widetilde{h}^c_{q_j}$ for each node q_j in Q. Then a comparison function is used to ($h^c_{p_i}$, $\widetilde{h}^c_{p_i}$) and ($h^c_{p_j}$, $\widetilde{h}^c_{p_j}$) to get enhanced representations $h^s_{p_i}$ and $h^s_{q_j}$ as equations (10)-(13).

Table 1. Statistics of datasets: SNLI and SciTail. Avg.L refers to average length of a pair of sentences.

	Train	Dev	Test	Avg.L		Vocab
SNLI	549K	9.8K	9.8K	14	8	36K
SciTail	23K	1.3K	2.1K	17	12	24K

Finally, we further fuse the above cross attention and the self-attention information as follows:

$$\widehat{h}_{p_i} = h^c_{p_i} + h^s_{p_i} \tag{16}$$

$$\widehat{h}_{q_j} = h^c_{q_j} + h^s_{q_j} \tag{17}$$

The representations $\widehat{h}_{p_i}$ and $\widehat{h}_{q_j}$ are learned from cross attention between two syntactically composed trees and then are augmented by self-attention. We then pass them into prediction layer.

3.4 Prediction Layer

We perform mean and max pooling on each sentence as Chen et al. [3], and use two-layer 1024-dimensional MLP with ReLU activation as classifier.

For model training, the object is to minimize the objective function $\mathcal{J}(\Theta)$:

$$\mathcal{J}(\Theta) = -\frac{1}{N}\sum_{i=1}^{N} \log P(y^{(i)}|p^{(i)}, q^{(i)}; \Theta) + \frac{1}{2}\lambda\|\Theta\|_2^2 \tag{18}$$

where Θ denotes all the learnable parameters, N is the number of instances in the training set, $(p^{(i)}, q^{(i)})$ are the sentence pairs, and $y^{(i)}$ denotes the annotated label for the i-th instance.

4 Experiments

4.1 Dataset

We evaluate our model on two datasets: the Stanford Natural Language Inference (SNLI) dataset [1] and the SciTail dataset [14]. The syntactic trees used in this paper are produced by the Stanford PCFG Parser 3.5.3 [16] and they are provided in these datasets.

The detailed statistical information of the two datasets is shown in Table 1.

4.2 Implementation Details

Following Tay et al. [24], we learn word embedding by concatenating pre-trained word vector, learnable word vector and POS vector. Then we use a ReLU layer to

Table 2. Comparison results on SNLI dataset.

Models	Train	Test
SAN [13]	89.6	86.3
BiMPM [25]	90.9	87.5
ESIM [3]	92.6	88.0
DIIN [11]	91.2	88.0
CAFE [24]	89.8	88.5
DR-BiLSTM [9]	94.1	88.5
AF-DMN [7]	94.5	88.6
SPINN [2]	89.2	83.2
NTI [19]	88.5	87.3
syn TreeLSTM [3]	92.9	87.8
Our model (single)	92.0	**88.8**
BiMPM (ensemble) [25]	93.2	88.8
ESIM (ensemble) [3]	93.5	88.6
DIIN (ensemble) [11]	92.3	88.9
CAFE (ensemble) [24]	92.5	89.3
DR-BiLSTM (ensemble) [9]	94.8	89.3
AF-DMN (ensemble) [7]	94.9	89.0
Our model (ensemble)	93.2	**89.5**

the concatenated vector. We set word embeddings, the hidden states of S-LSTM and ReLU to 300 dimensions. The pre-trained word vectors are 300-dimensional *Glove* 840*B* [21] and fixed during training. The learnable word vectors and POS vectors have 30 dimensions. The batch size is set to 64 for SNLI and 32 for SciTail. We use the Adam method [15] for training, and set the initial learning rate to 5e-4 and l_2 regularizer strength to 6e-5. For ensemble model, we average the probability distributions from three single models as in Duan et al. [7].

4.3 Comparison Results on SNLI

The comparison results on SNLI dataset is shown in Table 2.

The first group are sequential models that adopt attention to model word-level matching. SAN [13] is a distance-based self-attention network. BiMPM [25] design a bilateral multi-perspective matching model from both directions. ESIM [3] incorporate the chain LSTM and tree LSTM. CAFE [24] use novel factorization layers compress alignment vectors into scalar valued features. DR-BiLSTM [9] process the hypothesis conditioned on the premise results. DIIN [11] hierarchically extract semantic features using CNN. AF-DMN [7] adopt attention-fused deep matching network.

The second group are tree-structured models, of which SPINN [2] use Tree-LSTM with constituency parser tree, without attention. NTI [19] and syn Tree-LSTM [3] adopt attention for node matching. NTI use full binary tree while syn Tree-LSTM use constituency parser tree. Compared to Chen et al. [3], we use same parser tree but different tree composition function and attention mechanism.

Table 3. Comparison results on SciTail dataset.

Models	Dev	Test
Majority class	63.3	60.3
Ngram	65.0	70.6
DecompAtt	75.4	72.3
ESIM	70.5	70.6
DGEM	79.6	77.3
DEISTE	82.4	82.1
CAFE	-	83.3
Our model	**88.1**	**85.8**

Table 4. Ablation study on SNLI dev and test sets.

Models	Dev	Test
Only root node	86.2	85.9
+ Cross attention	88.9	88.2
+ Self-attention	89.4	88.8

In Table 2, our single and ensemble models achieve 88.8% and 89.5% test accuracy. The comparison results show that our model outperforms not only the existing tree-structured models, but also state-of-the-art achieved by sequential models on SNLI dataset.

4.4 Comparison Results on SciTail

The comparison results on SciTail dataset is shown in Table 3. SciTail is known to be a more difficult dataset for NLI. The first five models in Table 2 are all implemented in Knot et al. [14]. DGEM is a graph based attention model using syntactic structures. CAFE [24] adopt LSTM and attention for word-level matching. DEISTE [26] propose deep explorations of inter-sentence interaction.

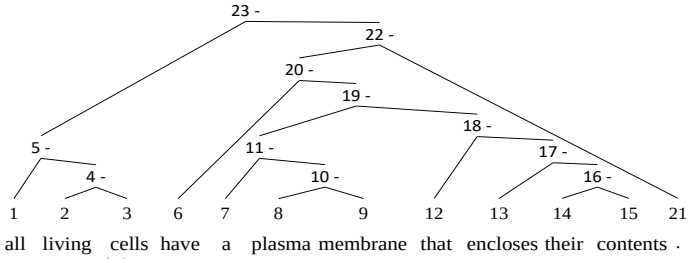
On this dataset, our single model significantly outperforms these previous models, and achieves 85.8% test accuracy.

4.5 Ablation Study

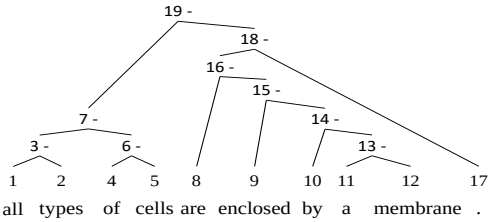
We conduct an ablation study to examine the effect of each key component of our model. As illustrated in Table 4, the first row is the model that uses the representation of root node to represent sentence, without attention. By adding cross attention and self-attention, the model performance is further improved. This proves the effect of our tree-structured composition and matching model.

4.6 Investigation on Attention

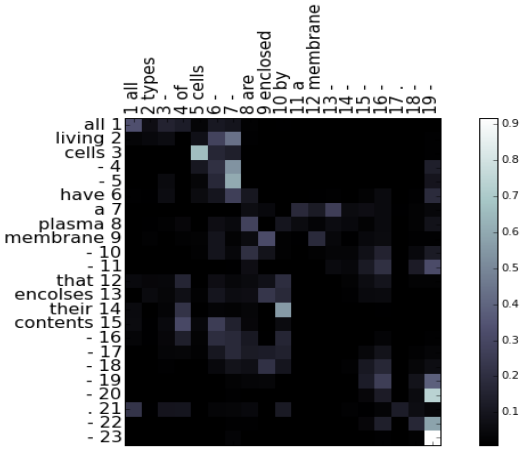
In this section, we investigate what information is captured by the attention, and visualize the cross attention results, as shown in Fig. 3. This is an instance from



(a) The syntactic tree of premise sentence P.



(b) The syntactic tree of hypothesis sentence Q.



(c) The attention results of two syntactic trees of P and Q.

Fig. 3. The syntactic trees and attention result for sentences P and Q.

the test set of SciTail dataset: {P: all living cells have a plasma membrane that encloses their contents. Q: all types of cells are enclosed by a membrane. The label y: entailment.}. From the results, it shows that our syntax-based model can semantically aligns word-level expressions (node 13 "encloses" and node 9 "enclosed") and phrase-level expressions (node 5 "all living cells" and node 7 "all types of cells") in the P and Q, respectively. We also observe that attention degree for the phrase expressions is more obvious than the single word that composes the phrase, such as node 17 in the P and node 16 in the Q. An intuitive explantation is that the syntax-based model can capture more rich semantics by

Table 5. Some complex examples and the classification. The E indicates entailment and the N indicates neutral between P and Q.

ID	Sentence1(P)	Sentence2(Q)	Bleu	Gold	Ours	CAFE
A	early morning sprinkling reduces water loss from evaporation.	watering plants and grass in the early morning is a way to conserve water because smaller amounts of water evaporate in the cool morning.	0.05	E	E	N
B	slow, deep watering allows plant roots to grow deep, prevents blow-over of your trees, and also minimizes salt buildup.	deep roots will best prevent a tree from being blown over by high winds during a storm.	0.10	E	E	N
C	they are among the most primitive of dicotyledonous angiosperm plants.	angiosperms are the most successful phylum of plants.	0.50	N	N	E
D	as the wheel turns, the arc causes the body to lift up.	the turning driveshaft causes the wheels of the car to turn.	0.42	N	N	E
E	multiple tissue types compose organs and body structures.	a(n) organ is a structure that is composed of one or more types of tissues.	0.05	E	N	N
F	both take approximately one year to orbit the sun.	it takes about one year for earth to orbit the sun.	0.53	N	E	E

using tree-structured composition. Finally, the syntax-based model attends over higher level tree nodes with rich semantics when considering longer phrase or full sentence, such as, the larger sub-trees 20, 22 and 23 in the P is aligned to the root node 19 that represents the whole semantics of the Q. It also indicates the composition function can effectively compute phrase representations.

4.7 Case Study

We show some examples from SciTail test dataset, as shown in Table 5. We compare the proposed syntax-based model with sequential model. For sequential model, we use the representative CAFE model [24]. We compute the Bleu score of the P with referenced to the Q and use 1-gram length. The Bleu score assumes the more overlapped words between two sentences, the closer the semantics are.

Examples A-B are entailment cases, but each of which has low Bleu score. Thus, it is more difficult to recognize the entailment relation between them. Our syntax-based approach correctly favors entailment in these cases. It indicates that the low lexical similarity challenges the sequential model to extract the related semantics, but it maybe solved by introducing syntactic structures.

The second set of examples C-D are neutral cases. Each of them has high Bleu score, where sequential model trends to misidentify the semantic relation to entailment, but our syntax-based model have the ability to correctly recognize the neutral relation. It indicates syntactic structure is more superior to solve semantic understanding involving structurally complex expressions.

Finally, examples E-F are cases that sequential and syntactic models get wrong. Examples E are entailment relation, but it have low Bleu score. Meanwhile, the word orders and structures ("compose" and "is composed of") of two sentences are also quite different. It causes models to failure recognizing the entailment relation between them. Example F is neutral relation where two sentences have high lexical overlap and also the similar word orders, which confuses

models to misclassify a entailment class. For the difficult cases, sentence semantics suffer more the issues, such as polysemy, ambiguity, as well as fuzziness, in which the model may need more inference information to distinguish these relations and make the correct decision, such as incorporating external knowledge to help model better understanding the lexical and phrasal semantics.

5 Conclusions and Future Work

In this paper, we explore the potential of syntactic tree for semantic computation and present a syntax-aware attention model for NLI. It leverages tree-structured composition and phrase-level matching. Experimental results on SNLI and SciTail datasets show that our model significantly improves the performance, and that the syntactic structure is important for modeling complex semantic relationship. In the future, we will explore the combination of syntax and pre-trained language model technology, to further improve the performance.

Acknowledgments. The authors are supported by the National Nature Science Foundation of China (Nos. 61876198, 61370130 and 61473294), the Fundamental Research Funds for the Central Universities (Nos. 2018YJS025 and 2015JBM033), and the International Science and Technology Cooperation Program of China (No. K11F100010).

References

1. Bowman, S.R., Angeli, G., Potts, C., Manning, C.D.: A large annotated corpus for learning natural language inference. arXiv preprint arXiv:1508.05326 (2015)
2. Bowman, S.R., Gauthier, J., Rastogi, A., Gupta, R., Manning, C.D., Potts, C.: A fast unified model for parsing and sentence understanding. arXiv preprint arXiv:1603.06021 (2016)
3. Chen, Q., Zhu, X., Ling, Z.H., Wei, S., Jiang, H., Inkpen, D.: Enhanced lstm for natural language inference. In: Proc. ACL (2017)
4. Chomsky, N.: Syntactic structures. the hague: Mouton.. 1965. aspects of the theory of syntax. Cambridge, Mass.: MIT Press.(1981) Lectures on Government and Binding, Dordrecht: Foris.(1982) Some Concepts and Consequences of the Theory of Government and Binding. LI Monographs **6**, 1–52 (1957)
5. Dowty, D.: Compositionality as an empirical problem. Direct compositionality (14), 23–101 (2007)
6. Dozat, T., Manning, C.D.: Deep biaffine attention for neural dependency parsing (2016)
7. Duan, C., Cui, L., Chen, X., Wei, F., Zhu, C., Zhao, T.: Attention-fused deep matching network for natural language inference. In: IJCAI. pp. 4033–4040 (2018)
8. Eriguchi, A., Hashimoto, K., Tsuruoka, Y.: Tree-to-sequence attentional neural machine translation. arXiv preprint arXiv:1603.06075 (2016)
9. Ghaeini, R., Hasan, S.A., Datla, V., Liu, J., Lee, K., Qadir, A., Ling, Y., Prakash, A., Fern, X.Z., Farri, O.: Dr-bilstm: Dependent reading bidirectional lstm for natural language inference (2018)

10. Glorot, X., Bordes, A., Bengio, Y.: Deep sparse rectifier neural networks. In: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics. pp. 315–323 (2011)
11. Gong, Y., Luo, H., Zhang, J.: Natural language inference over interaction space. arXiv preprint arXiv:1709.04348 (2017)
12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8), 1735–1780 (1997)
13. Im, J., Cho, S.: Distance-based self-attention network for natural language inference (2017)
14. Khot, T., Sabharwal, A., Clark, P.: Scitail: A textual entailment dataset from science question answering. In: Proceedings of AAAI (2018)
15. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
16. Klein, D., Manning, C.D.: Accurate unlexicalized parsing. In: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1. pp. 423–430. Association for Computational Linguistics (2003)
17. Li, J., Dan, J., Hovy, E.: When are tree structures necessary for deep learning of representations? Computer Science (2015)
18. MacCartney, B., Manning, C.D.: Modeling semantic containment and exclusion in natural language inference. In: Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1. pp. 521–528. Association for Computational Linguistics (2008)
19. Munkhdalai, T., Yu, H.: Neural tree indexers for text understanding. In: Proceedings of the conference. Association for Computational Linguistics. Meeting. vol. 1, p. 11. NIH Public Access (2017)
20. Parikh, A.P., Täckström, O., Das, D., Uszkoreit, J.: A decomposable attention model for natural language inference. arXiv preprint arXiv:1606.01933 (2016)
21. Pennington, J., Socher, R., Manning, C.: Glove: Global vectors for word representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 1532–1543 (2014)
22. Socher, R., Huval, B., Manning, C.D., Ng, A.Y.: Semantic compositionality through recursive matrix-vector spaces. In: Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning. pp. 1201–1211. Association for Computational Linguistics (2012)
23. Tai, K.S., Socher, R., Manning, C.D.: Improved semantic representations from tree-structured long short-term memory networks. arXiv preprint arXiv:1503.00075 (2015)
24. Tay, Y., Tuan, L.A., Hui, S.C.: A compare-propagate architecture with alignment factorization for natural language inference. arXiv preprint arXiv:1801.00102 (2017)
25. Wang, Z., Hamza, W., Florian, R.: Bilateral multi-perspective matching for natural language sentences. corr abs/1702.03814 (2017)
26. Yin, W., Dan, R., Schütze, H.: End-task oriented textual entailment via deep exploring inter-sentence interactions (2018)
27. Yogatama, D., Blunsom, P., Dyer, C., Grefenstette, E., Ling, W.: Learning to compose words into sentences with reinforcement learning. arXiv preprint arXiv:1611.09100 (2016)
28. Zhu, X., Sobhani, P., Guo, H.: Long short-term memory over recursive structures. In: International Conference on International Conference on Machine Learning (2015)