Better Modeling the Hierarchical Structure of Language for Sentence Semantic Matching

Zhongmin Pei, Changqun Li*, Wei Xiong and Zhangkai Luo

1Science and Technology on Complex Electronic System Simulation Laboratory, Space Engineering University, Beijing, 100014, China

Abstract. Sentence semantic matching requires understanding the semantic relationship between two sentences, which is widely used in various natural language tasks such as paraphrase identification, natural language inference and question answering. Usually, the underlying structure of sentence is not strictly sequential, this structure is hierarchical. This paper studies how to better model the hierarchical structure of sentence for semantic matching. Recent research suggests that BERT implicitly captures classical, tree-like structures. Based on this, this paper proposes to further enhance the strength of modeling hierarchical structure of language with an advanced variant of LSTM – Ordered Neurons LSTM (ON-LSTM), which introduces a syntax-oriented inductive bias to perform the composition of classical tree structure. Experimental results demonstrate that the proposed approach, which enhances hierarchical structure of language, significantly improves the performance of sentence semantic matching.

1. Introduction

Sentence semantic matching is the basic technology of natural language processing. It is necessary to estimate the semantic relationship between two sentences. In Paraphrase Identification (PI), sentence semantic matching is used to identify whether two sentences are semantically equivalent[1]. In Natural Language Inference (NLI), it determines whether a hypothetical sentence can be inferred from a given premise sentence[2].

The oral and written forms of natural language are continuous, but the basic structure of language is not strictly continuous. This structure is usually hierarchical. Bidirectional Encoder Representations from Transformers (BERT)[3] has shown significant improvements across various NLP tasks. Recent studies have found that BERT implicitly captures the hierarchical structure information of language by imitating the composition of classical tree structure[4].

To explicitly enhance BERT’s ability of modeling hierarchical structure information of language, this paper proposes to a hybrid model which combines the strengths of BERT and a LSTM variant – Ordered Neurons LSTM (ON-LSTM)[5]. ON-LSTM is better at modeling hierarchical structure of language by introducing a syntax-oriented inductive bias, which enables RNNs to perform composition of classical tree structure by changing neuron’s update frequency. Specifically, this paper stacks ON-LSTM encoder on top of BERT encoder. BERT encoder is able to implicitly capture and exploit hierarchical information and ON-LSTM encoder can explicitly reinforce structured representation. This paper proposes to simultaneously incorporate two representations by combining outputs of BERT’s intermediate layers and ON-LSTM encoders.
2. Approach
BERT’s intermediate layers implicitly encode hierarchy of linguistic information, this paper stacks a RNNs encoder on top of a BERT encoder to form a stacked encoder. In the stacked encoder, BERT encoder is able to extract better representations and hierarchical structure modeling is enhanced by RNNs encoder. Let $X = \{x_1, x_2, ..., x_n\}$ mean the input sequence, the representation of the stacked encoder is calculated by

$$H_{BERT} = ENC_{BERT}(X)$$

$$H_{RNNs} = ENC_{RNNs}(H_{BERT})$$

where $ENC_{BERT}(\cdot)$ is a encoder that reads the input sequence and extracts better representations, $ENC_{RNNs}(\cdot)$ is a L-layer RNNs encoder that receives the output of BERT encoder as input.

In this work, we replace the standard LSTM with the latest ON-LSTM to better model the language hierarchy, and concat BERT different layers’ outputs to build a richer representation, as described below.

2.1. Enhancing the Hierarchical Structure of Language with Ordered Neurons
ON-LSTM introduces a new grammar-oriented inductive bias by sorting the neurons, which enables LSTM models to perform composition of classical tree structure. Ordered neurons enable to store different long/short-term information by changing neuron’s update frequency. In ordered neurons, different neurons have different frequency of update, and the update frequency is predetermined as part of the model architecture. Therefore, ON-LSTM introduces new update function to change cell state:

$$\omega_t = \widetilde{f}_t \circ \tilde{\omega}_t(3)$$

$$\widetilde{f}_t = f_t \circ \omega_t + (\overline{f}_t - \omega_t)(4)$$

$$\tilde{\omega}_t = i_t \circ \omega_t + (\overline{\omega}_t - \omega_t)(5)$$

$$c_t = \overline{f}_t \circ c_{t-1} + \overline{i}_t \circ \overline{c}_t(6)$$

where forget gate $f_t$, input gate $i_t$ and cell state $c_t$ are same as that in LSTM[6]. The master forget gate $\overline{f}_t$ and the master input gate $\overline{i}_t$ are introduced to adjust the writing and erasing operation on cell states $c_t$. $\omega_t$ represents the overlap of $\widetilde{f}_t$ and $\tilde{\omega}_t$, and when the overlap exists ($\exists k$, $\omega_{tk} > 0$), the corresponding neurons are further adjusted by the $f_t$ and $i_t$ in LSTM[6] model.

The ideal master gate divides the cell state into two segments: the 0-segment and the 1-segment, and the format is binary, for example (0,0,1,1,1). The neurons corresponding to 0-segment and 1-segment are updated with different frequencies, specifically, the information in 0-segment neurons will be preserved for less time steps, while the information in 1-segment neurons will be retained for more time steps. Because this binary gates are not differentiable, it is necessary to find the splitting point $d$, a categorical random variable, indicates the position subscript of the first 1 in the ideal master gate. Finally, ON-LSTM[5] introduces a brand new activation function:

$$cu(\cdot) = \text{cumsum}(	ext{softmax}(\cdot))$$

where softmax(\cdot) generates the probability distribution of each position becoming the split point $d$ (e.g.(0.2,0.2,0.3,0.2,0.1)). cumsum denotes the cumulative sum, used to calculate continuous relaxation. The output for the above example is (0.2,0.4,0.7,0.9,1.0), in which the k-th probability means that d falls within the first k probabilities and different values denotes different update frequencies. $cu(\cdot)$ is seen as the expectation of a binary gate. So the master gates are defined as

$$\widetilde{f}_t = cu_f(x_t, h_{t-1})(8)$$

$$\overline{i}_t = 1 - cu_i(x_t, h_{t-1})(9)$$

where $x_t$ is the present input and $h_{t-1}$ is the hidden state of previous time step. $cu_f$ and $cu_i$ are activation functions of trainable parameters.

2.2. Residual Connection and Concatenation
Similar to positional encoding injection in Transformer[7], this paper proposes to simultaneously incorporate two representations by a simple residual connection[8]. Specifically, this paper adds the output of the BERT to the output of ON-LSTM encoder.
To form a more comprehensive representation, this paper concatenates output of BERT’s higher layers:

\[ H = H_{BERT} + H_{ON-LSTM} \] (10)

where \( H_{ON-LSTM} \in \mathbb{R}^{N \times d} \) is the output of ON-LSTM encoder, and \( H_{BERT} \in \mathbb{R}^{N \times d} \) is the output of BERT encoder.

3. Experiments

This paper first evaluates the model performance on three benchmark datasets for two challenging sentence semantic matching tasks: 1) the Large-scale Chinese Question Matching Corpus (LCQMC)[9] and Quora Question Pair[10] for paraphrase identification; 2) SNLI[11] for natural language inference. Then, we analyze the model and experimental results in detail.

3.1. Data Description

This paper evaluates model on three well-studied datasets: LCQMC, Quora Question Pairs (QQP), and the Stanford Natural Language Inference (SNLI).

LCQMC. The LCQMC contains 260,068 Chinese question pairs. Each pair has a binary value that indicates whether two questions are semantically equivalent.

Quora. The Quora Question Pair dataset consists of over 400,000 potential question duplicate pairs. Each pair has a binary value that indicates whether the pair is a duplicate pair.

SNLI. The SNLI contains 570,152 human annotated sentence pairs. Each sentence pair is marked with one of three relationships: Entailment, Contradiction, or Neutral.

3.2. Hyperparameters and Training Details

This paper evaluates ESIM[13], the ON-LSTM[5], BERT[3] and proposed model on semantic matching dataset. Given a pair of sentences \((s_1, s_2)\), for ON-LSTM encoder, the last hidden state \((h_1, h_2)\) is used as the sentence embedding. The trilinear function[12] \( W_0[s_1, s_2, s_1 \odot s_2] \) is used as the input to a multi-layer classifier. For proposed model, similar to BERT’s input, packing two sentences together and separates them with a special token ([SEP]). And a `sequence` always begins with a special classification token ([CLS]) and ends with a special token ([SEP]). This paper will use two hidden layers together with Residual Connection. In order to make a fair comparison, for each dataset, this paper keeps the same hyper-parameters (such maximum length, batch size, etc) and only tune the initial learning rate from 1e-4 to 1e-5 for Pre-trained Language model.

3.3. Experiment Results

We evaluate models on the PI task over LCQMC and QQP datasets. Table 1 shows the results of our model compared with other published sentence encoding-based models. We utilize the accuracy to evaluate their performances.

| Model                               | LCQMC test | QQP test |
|-------------------------------------|------------|----------|
| ESIM                                | 83.9%      | 86.0%    |
| 3L + LSTM + trilinear function      | 69.9%      | 78.5%    |
| 3L + ON-LSTM + trilinear function   | 72.5%      | 79.4%    |
| BERT                                | 86.9%      | 89.7%    |
| Hybrid Model + Residual Connection  | 87.3%      | 90.2%    |

Table 1. Performance (accuracy) of models on different LCQMC and QQP, “nL LSTM, mL ON-LSTM” denotes stacking n LSTM layers and m ON-LSTM layers, “Hybrid Model” denotes “BERT + 3L ON-LSTM”. 
From Table 1, we can figure out that Hybrid Model achieves the best performance on all test sets. As described before, Hybrid Model is able to model sentence semantic comprehensively by enhancing the strength of modeling hierarchical structure. Therefore, the sentence semantic can be fully explored.

Among these sentence encoding-based baselines, pre-trained language model BERT is the current state-of-the-art models. However, BERT cannot model the hierarchy of the input sentence well, which is essential for understanding the language. Hybrid Model which combines the strengths of BERT and LSTM has outperformed single model on sentence semantic matching task.

In order to further verify the effect of modeling language hierarchical structure on the semantic matching task, this paper compares two modes of LSTM and ON-LSTM. From the table 1, it can be seen that modeling language hierarchical structure is beneficial to the semantic matching task.

Table 2. Performance (accuracy) of models on SNLI.

| Model                                    | Accuracy |
|------------------------------------------|----------|
| Distance-based SAN[14]                   | 86.3%    |
| DRCN[2]                                  | 86.5%    |
| DRr-Net[15]                              | 87.7%    |
| BERT[3]                                  | 89.4%    |
| Hybrid Model + Residual Connection       | 90.1%    |

Table 2 shows performance of all models on the SNLI dataset. Distance-based SAN considers the word distance by using a simple distance mask to model the local dependency, which can effectively explore the sentence semantic from multiple aspects. DRr-Net using Attention Stack-GRU (ASG) unit to repeatedly read the sentence for better sentence semantic understanding. We obtain accuracy of 90.1% on this dataset, surpassing the previous state-of-the-art model of BERT and DRr-Net.

Table 3. Performance (accuracy) of individual model and hybrid models on different datasets.

| Encoder Architecture | LCQMC | QQP   | SNLI  |
|----------------------|-------|-------|-------|
| BERT                 | 86.9% | 89.7% | 89.4% |
| BERT + 1L ON-LSTM    | 85.0% | 89.9% | 89.6% |
| BERT + 2L ON-LSTM    | 87.0% | 90.0% | 89.8% |
| BERT + 3L LSTM + Residual Connection | 87.1% | 90.1% | 89.9% |
| BERT + 3L ON-LSTM + Residual Connection | 87.3% | 90.2% | 90.1% |

Table 3 shows performances of the single model and the hybrid models on the three datasets. Hybrid models consistently outperform the single models. And the ON-LSTM model is significantly better than its LSTM counterpart. We think ON-LSTM has advantages in modeling hierarchical structure, and we believe the hierarchy of the input sentence is important for understanding the language. In addition, the residual connection strategy improves performances by providing richer representations.

4. Related Work

There has been prior work modeling hierarchical structures for natural language sentences in the literature[16]. With the rise of deep learning, there are many works that incorporate syntactic tree structure into Recursive Neural Networks[18], LSTMs[19], CNNs[20]. [21] demonstrates that tree-structured models are more effective for downstream tasks whose data was generated by recursive programs. ON-LSTM[5] proposed to model hierarchical structure by introducing a syntax-oriented inductive bias. Recent research finds BERT implicitly captures classical, tree-like structures and substantial syntactic information is captured in its attention[4].
5. Conclusion
In this paper, we proposed a Hybrid Model for sentence semantic matching, a novel architecture that was able to enhance the strength of modeling hierarchical structure of language for better sentence semantic matching. To be specific, we employed ON-LSTM to improve the BERT model. We also recommend modifying the stacked encoder by combining the outputs of each part to form a more comprehensive representation. Experimental results on paraphrase identification and natural language inference tasks demonstrated the superiority of the proposed model.

Acknowledgments
This work was supported by the Fundamental Research Program of Science and Technology on Complex Electronic System Simulation Laboratory.

References
[1] Dolan W B, Brockett C. Automatically constructing a corpus of sentential paraphrases[C]//Proceedings of the Third International Workshop on Paraphrasing (IWP2005). 2005.
[2] Kim S, Kang I, Kwak N. Semantic sentence matching with densely-connected recurrent and co-attentive information[C]//Proceedings of the AAAI conference on artificial intelligence. 2019, 33: 6586-6593.
[3] Devlin J, Chang M W, Lee K, et al. Bert: Pre-training of deep bidirectional transformers for language understanding[J]. arXiv preprint arXiv:1810.04805, 2018.
[4] Jawahar G, Sagot B, Seddah D. What does BERT learn about the structure of language?[C]. 2019.
[5] Shen Y, Tan S, Sordoni A, et al. Ordered neurons: Integrating tree structures into recurrent neural networks[J]. arXiv preprint arXiv:1810.09536, 2018.
[6] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
[7] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C]//Advances in neural information processing systems. 2017: 5998-6008.
[8] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 770-778.
[9] Liu X, Chen Q, Deng C, et al. Lcqmc: A large-scale chinese question matching corpus[C]//Proceedings of the 27th International Conference on Computational Linguistics. 2018: 1952-1962.
[10] Iyer S, Dandekar N, Csernai K. First quora dataset release: Question pairs[J]. data. quora. com, 2017.
[11] Bowman S R, Angeli G, Potts C, et al. A large annotated corpus for learning natural language inference[J]. arXiv preprint arXiv:1508.05326, 2015.
[12] Seo M, Kembhavi A, Farhadi A, et al. Bidirectional attention flow for machine comprehension[J]. arXiv preprint arXiv:1611.01603, 2016.
[13] Chen Q, Zhu X, Ling Z, et al. Enhanced lstm for natural language inference[J]. arXiv preprint arXiv:1609.06038, 2016.
[14] Im J, Cho S. Distance-based self-attention network for natural language inference[J]. arXiv preprint arXiv:1712.02047, 2017.
[15] Zhang K, Lv G, Wang L, et al. Drr-net: Dynamic re-read network for sentence semantic matching[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2019, 33: 7442-7449.
[16] Chomsky N. Aspects of the Theory of Syntax[M]. MIT press, 2014.
[17] Bever T G. The cognitive basis for linguistic structures[J]. Cognition and the development of language, 1970.
[18] Socher R, Perelygin A, Wu J, et al. Recursive deep models for semantic compositionality over a sentiment treebank[C]//Proceedings of the 2013 conference on empirical methods in natural
language processing. 2013: 1631-1642.

[19] Tai K S, Socher R, Manning C D. Improved semantic representations from tree-structured long short-term memory networks[J]. arXiv preprint arXiv:1503.00075, 2015.

[20] Mou L, Li G, Zhang L, et al. Convolutional neural networks over tree structures for programming language processing[C]//Thirtieth AAAI Conference on Artificial Intelligence. 2016.

[21] Bowman S, Potts C, Manning C D. Recursive neural networks can learn logical semantics[C]//Proceedings of the 3rd workshop on continuous vector space models and their compositionality. 2015: 12-21.

[22] Bowman S R, Manning C D, Potts C. Tree-structured composition in neural networks without tree-structured architectures[J]. arXiv preprint arXiv:1506.04834, 2015.