Hybrid BiLSTM-Siamese Network for Relation Extraction

Extended Abstract

Zeyuan Cui
Shandong University
Jinan, China

Li Pan
Shandong University
Jinan, China

Shijun Liu*
Shandong University
Jinan, China

ABSTRACT

Relation extraction is an important processing task in knowledge graph completion. In previous approaches, it is considered to be a multi-class classification problem. In this paper, we propose a novel approach called hybrid BiLSTM-Siamese network which combines two word-level bidirectional LSTMs by a Siamese model architecture. It learns a similarity metric between two sentences and predicts the relation of a new sentence by k-nearest neighbors algorithm. In experiments, we use the SemEval-2010 Task8 dataset and achieve an F1-score of 81.8%.

KEYWORDS

Siamese Network; Relation Extraction; Knowledge Graph

ACM Reference Format:
Zeyuan Cui, Li Pan, and Shijun Liu. 2019. Hybrid BiLSTM-Siamese Network for Relation Extraction. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019,* IFAAMAS, 3 pages.

1 INTRODUCTION

Relation extraction, aiming to find the semantic relation between a pair of entities given a sentence, is an important processing task in knowledge graph completion. For example, in the sentence "*The <e1> fire </e1> inside WTC was caused by exploding <e2> fuel </e2>*", <e1>, </e1>, <e2>, </e2> are four position indicators which specify the starting and ending of the entities. So, "fire" is the first entity and "fuel" is the second entity. The sentence expresses the relation *Cause-Effect (e2, e1)* between the two entities.

In previous approaches, relation extraction is considered to be a multi-class classification problem. They use neural networks to extract a feature vector of a sentence, and then feed it into a softmax classifier. For example, $O \in R^n$ is the final output of the model, where n is equal to the number of possible relation types. Then O_i can be interpreted as the confidence score of the i-th relation type. Finally, select the most confident relation type as the final result. Actually, a relation type is usually expressed with certain patterns in various sentences. For example, many sentences expressing the relation *Cause-Effect* usually contain the phrase "be caused by", like "an accident was caused by the fog" and "the spiral light was caused by the missile". This clue provides us a new way for relation extraction that we can learn a similarity metric between two sentences and predict the relation of a new sentence by KNN.

*Corresponding author: lsj@sdu.edu.cn, panli@sdu.edu.cn. This work was supported by the National Key Research and Development Program of China (2017YFA0700601).

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada.

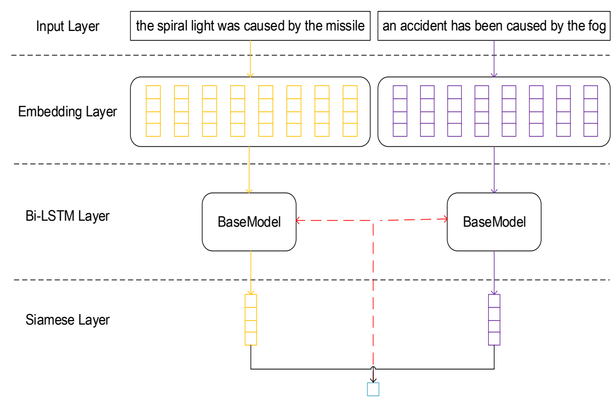

Figure 1: The architecture of the proposed model.

In this paper, we propose a novel approach called hybrid BiLSTM-Siamese network for relation extraction. The model combines two word-level bidirectional long short-term memory neural networks by a Siamese model architecture. The model learns a similarity metric to distinguish pairs of sentences with the same relation from those with different relations. We conduct experiments on a widely used relation extraction dataset. The experimental results show that our approach is competitive with some existing approaches, with lower complexity.

2 METHODOLOGY

Figure 1 shows the architecture of the proposed model for relation extraction. The model consists of four layers: input layer, embedding layer, Bi-LSTM layer and Siamese layer. The input of the model is a pair of sentences. The embedding layer transforms an input sentence into a sequence of vectors. In Bi-LSTM layer, the pair of inputs are passed through the twin networks respectively and transformed into two feature vectors. Then we can calculate the distance of these two vectors in the Siamese layer. During training of the model, we try to project similar relation pairs nearer to each other and dissimilar relation pairs farther away from each other. Then, we can predict the relation of a new sentence by KNN. We describe our proposed model in detail below.

Embedding Layer In this layer, we get a low-dimensional vector V_w for each word in the sentence. To capture syntactic and semantic meanings of words, we use word embeddings learned from a large amount of corpus. For each word, we look up the pre-trained embedding matrix and get a vector V_e. There are relative distances to entity1 and entity2 for each word. In our model, the relative distances are mapped to vectors V_{p1} and V_{p2} by looking up position embeddings matrix which is randomly initialized and optimized through back propagation. Finally, for each word, we concatenate the lexical and position feature vectors and get a vector $V_w = [V_e, V_{p1}, V_{p2}]$. The input sentence is transformed into a vector sequence which is fed into the next layer.

Bi-LSTM Layer The Long Short Term Memory(LSTM) network is suitable for modeling sequential data. However, standard LSTM network processes sequences in temporal order. By introducing another layer flow in opposite temporal order, bidirectional LSTM network is able to exploit information both from left and right. So, for the base model, we use bidirectional LSTM. In this way, we get the output vector O_s of each input sentence S.

Siamese Model Different from ordinary neural networks which take one input, the hybrid BiLSTM-Siamese network takes a pair of inputs. The input sentences are passed through two base models respectively and transformed into two feature vectors. Assuming O_i and O_j are the outputs of the input sentences S_i and S_j, we define that $d(O_i, O_j)$ is the euclidean distance between the O_i and O_j. Then, we use $d(O_i, O_j)$ to measure whether the two input sentences belong to same relation or not.

During training of the model, when two input sentences belong to the same relation, we minimize $d(O_i, O_j)$ to reduce the distance between the outputs. Otherwise, we try to project O_i and O_j far away from each other. The loss function is $loss(O_i, O_j) = (1 - l_{ij})max(0, d(O_i, O_j)) + l_{ij}max(0, m - d(O_i, O_j))$. Here, $l_{ij} = 0$ when two input sentences belong to the same relation. Otherwise, $l_{ij} = 1$. m is the margin which decides by how much distance dissimilar pairs should be moved away from each other. During training of the model, the loss function is minimized and the parameters of the model is updated via back-propagation.

To predict the relation of a new sentence, we need to calculate the distance between the new sentence and some sentences which relation types are already known. For each relation r, we get a sample set $set(r)$ which is a subset of training set. When given a new sentence, it can make pair with each sentence in sample sets. By using the trained model, we can get the distance between the new sentence and every sentence in sample sets. At the end, we use k-nearest neighbors algorithm (KNN) for classification. In this way, the new sentence is classified by a majority vote of its k similar sentences in the sample set.

3 EXPERIMENTS

In order to evaluate the performance of our model, we use the SemEval-2010 Task 8 dataset and adopt the official evaluation metric which is based on macro-averaged F1-score. Our model achieves an F1-score of 81.8%. As shown in Figure 2, we compare our model with previous models including SVM [4], FCM [8], CNN [9], depLCNN [6], depLCNN+NS [6], RNN [5], MV-RNN [5], RNN [2], SDP-LSTM [7], Att-BLSTM [10], LSTM-RNN [3] and BRCNN [1]. For some models, they utilize rich features to achieve higher score. So, there are two results which are called "base model" and "add features". However, this is not a fair comparison because those models usually use elaborately designed features, complicated natural language processing tools and lexical resources. Compared to these models, our model is much simple and effective.

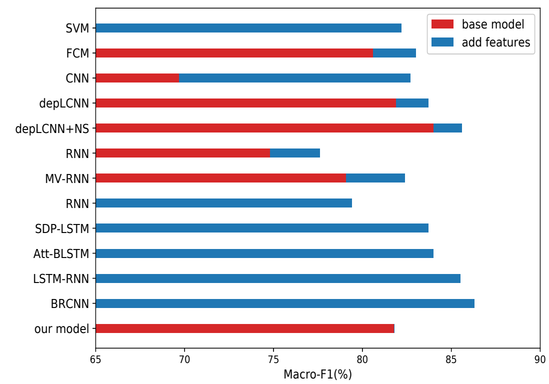

Figure 2: Comparison of state-of-the-art models.

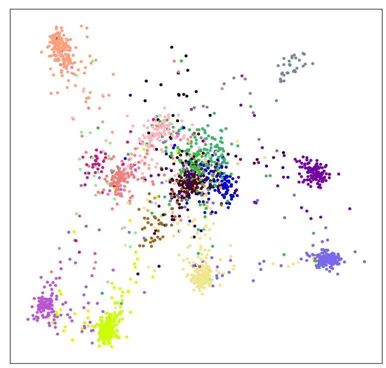

Figure 3: The overview of the output in testing dataset (after PCA dimension reduction).

To demonstrate the result of our model, we use principal components analysis (PCA) for visualization of the outputs in testing dataset, as shown in Figure 3. All the sentences are passed through the base model to get outputs in the relational space. Then, we can project the outputs onto two principal components and color all the sentences based on the relation which they belong to. We can find that same color points are nearer to each other.

4 CONCLUSION

In this paper, we propose a novel neural network model, named hybrid BiLSTM-Siamese network, for relation extraction. Our model is a deep architecture for learning a similarity metric on a pair of variable-length sequences, which combines two word-level bidirectional LSTM models with a Siamese architecture. The model projects similar relation pairs nearer to each other and dissimilar relation pairs farther away from each other. We demonstrate the effectiveness of our model by evaluating the model on SemEval-2010 relation classification task.

REFERENCES

[1] Rui Cai, Xiaodong Zhang, and Houfeng Wang. 2016. Bidirectional Recurrent Convolutional Neural Network for Relation Classification. In *Meeting of the Association for Computational Linguistics*. 756–765.

[2] Kazuma Hashimoto, Makoto Miwa, Yoshimasa Tsuruoka, and Takashi Chikayama. 2013. Simple Customization of Recursive Neural Networks for Semantic Relation Classification. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013, 18-21 October 2013, Grand Hyatt Seattle, Seattle, Washington, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*. 1372–1376.

[3] Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770* (2016).

[4] Bryan Rink and Sanda Harabagiu. 2010. UTD: Classifying semantic relations by combining lexical and semantic resources. In *International Workshop on Semantic Evaluation*. 256–259.

[5] Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. 1201–1211.

[6] Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015. Semantic Relation Classification via Convolutional Neural Networks with Simple Negative Sampling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. 536–540.

[7] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying Relations via Long Short Term Memory Networks along Shortest Dependency Paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*. 1785–1794.

[8] Mo Yu, Matthew Gormley, and Mark Dredze. 2014. Factor-based compositional embedding models. In *NIPS Workshop on Learning Semantics*. 95–101.

[9] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation Classification via Convolutional Deep Neural Network. In *COLING 2014, 25th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers, August 23-29, 2014, Dublin, Ireland*. 2335–2344.

[10] Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. In *Meeting of the Association for Computational Linguistics*. 207–212.