Supplementary Material for "Dependency-Guided LSTM-CRF for Named Entity Recognition"

Zhanming Jie and **Wei Lu**
StatNLP Research Group
Singapore University of Technology and Design
zhanming_jie@mymail.sutd.edu.sg, luwei@sutd.edu.sg

Abstract

We present the details of the baseline as well as further experiment details (including how to obtain predicted dependencies) in our main paper (Jie and Lu, 2019).

A Baseline Systems

We implemented the BiLSTM-CRF (Lample et al., 2016) and BiLSTM-GCN-CRF models based on the contextualized GCN implementation by Zhang et al. (2018). The implementation of BiLSTM-CRF is exactly same as Lample et al. (2016). We presents the neural architecture for the BiLSTM-GCN-CRF model.

A.1 BiLSTM-GCN-CRF

Figure 1 shows the neural architecture for the BiLSTM-GCN-CRF model. Following Zhang et al. (2018), the input representation at each position $\mathbf{w}_i$ is the word representation which consists of the pre-trained word embeddings and its character representation. To capture contextual information, we stack a BiLSTM layer before the GCN. Secondly, the GCN captures the dependency tree structure as shown in Figure 1. Following Zhang et al. (2018), we treat the dependency trees as undirected and build a symmetric adjacency matrix during the GCN update:

$$\mathbf{h}_i^{(l)} = \text{ReLU}\Big(\sum_{j=1}^{n} A_{ij}\mathbf{W}^{(l)}\mathbf{h}_j^{(l-1)} + \mathbf{b}^{(l)}\Big) \quad (1)$$

where $\mathbf{A}$ is the adjacency matrix. $A_{ij} = 1$ indicates there is a dependency edge between the i-th word and the j-th word[1]. $\mathbf{h}_i^{(l)}$ is the hidden state at the i-th position in the l-th layer. We can stack J layers of GCN in the model. In our experiments, we set the number of GCN layers $J = 1$ as we did not observe significant improvements by increasing J. In fact, we might obtain harmful performance for a larger J as deeper GCN layers will diminish the effect of the contextual information, which is important for the task of NER.

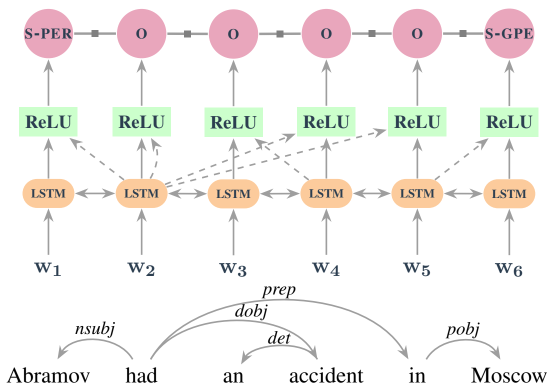

Figure 1: BiLSTM-GCN-CRF. Dashed connections mimic the dependency edges.

However, Equation 1 does not include the dependency relation information. As mentioned in the main paper, such relations have strong correlations with the entity types. We modify the Equation 1 and include the dependency relation parameter[2]:

$$\mathbf{h}_i^{(l)} = \sigma\Big(\sum_{j=1}^{n} A_{ij}\big(\mathbf{W}_1^{(l)}\mathbf{h}_j^{(l-1)} + \mathbf{W}_2^{(l)}\mathbf{h}_j^{(l-1)} w_{r_{ij}}\big)\Big)$$

where $w_{r_{ij}}$ is the dependency relation weight that parameterize the dependency relation r between the i-th word and the j-th word. Such formulation uses the relation to weight the adjacent hidden states in the dependencies.

B Implementation Details

We implemented all the models with PyTorch (**?**). For both BiLSTM-CRF and DGLSTM-CRF

[1] $A_{ij} = A_{ji}$ for symmetric matrix.

[2] The bias vector is ignore for brevity.

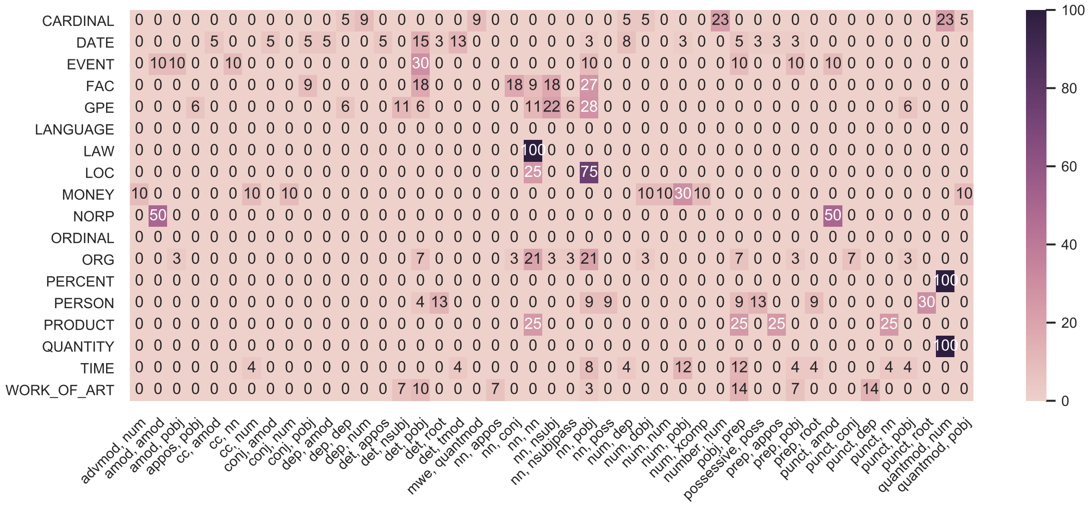

Figure 2: Correlations between the entity types and the dependency relation pairs on the grandchild dependencies.

model, we train them on all datasets with 100 epochs and take the model that perform the best on the development set. For BiLSTM-GCN-CRF, we train for 300 epochs with a clipping rate of 3.

C Relation Pairs on Grandchild Dependencies

Figure 2 visualized the correlations between the entities and the grandchild dependency relation pairs on the OntoNotes English dataset. As mentioned in the paper, such entities are correctly predicted by our models but not the BiLSTM-CRF baseline. As we can see from the figure, most of these entities correlate to the "(*nn*, *nn*)" and "(*nn*, *pobj*)" relation pairs on the grandchild dependencies. Such correlations also show that the relation pair information on the grandchild dependencies can be helpful for detecting certain entities.

D Using Predicted Dependency

We train a BERT-based (Devlin et al., 2019) dependency parser (Dozat and Manning, 2017) using the training set for each of four languages. Specifically, we adopt the `bert-base-uncased` model for English, `bert-base-multilingual-cased` for Catalan and Spanish and `bert-base-chinese` for Chinese. Because the Chinese BERT model is based on characters but not Chinese words which are segmented. We further incorporate a span extractor layer right after BERT encoder for Chinese. We following Lee et al. (2017) to design the span extractor layer. Our code for dependency parser is available at `https://github.com/allanj/bidaf_dependency_parsing`

References

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL*.

Timothy Dozat and Christopher D Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of ICLR*.

Zhanming Jie and Wei Lu. 2019. Dependency-guided lstm-crf for named entity recognition. In *Proceedings of EMNLP-IJCNLP*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT*.

Kenton Lee, Luheng He, Mike Lewis, and Luke Zettlemoyer. 2017. End-to-end neural coreference resolution. In *Proceedings of EMNLP*.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of EMNLP*.