



Enhanced Named Entity Recognition with Semantic Dependency

Peng Wang^{1,2}, Zhe Wang³, Xiaowang Zhang^{1,2(✉)}, Kewen Wang³,
and Zhiyong Feng²

¹ State Key Laboratory of Communication Content Cognition, Beijing, China

² College of Intelligence and Computing, Tianjin University, Tianjin, China
xiaowangzhang@tju.edu.cn

³ School of Information and Communication Technology, Griffith University,
Brisbane, Australia

Abstract. Dependency-based models for the named entity recognition (NER) task have shown promising results by capturing long-distance relationships between words in a sentence. However, while existing models focus on the syntactic dependency between entities, we are unaware of any work that considers semantic dependency. In this work, we study the usefulness of semantic dependency information for NER. We propose a NER model that is guided by semantic dependency graphs instead of syntactic dependency trees. The extensive experiments illustrate the effectiveness of the proposed model and the advantages of semantic dependency over syntactic dependency for NER. Also, it shows correlations between the NER performance and the semantic dependency annotations qualities.

Keywords: Named entity recognition · Syntactic dependency · Semantic dependency · Graph neural network

1 Introduction

Named Entity Recognition is one of NLP tasks to recognize named entities from texts belonging to pre-defined semantic types such as person, date, events, location, etc. [21, 23]. NER has attracted wide interest not only as a standalone task of information extraction, but also as an essential semantic information extraction step for downstream Natural language processing(NLP) tasks such as entity linking [25], entity relationship extraction [16], and semantic parsing [4].

Meanwhile, research in linguistic dependency theory shows that there exists a subject-subordinate relationship between words, and such a dependency structure could also capture useful semantic information within sentences. Based on such insight, there have been quite some research efforts in enhancing NER models through grammar dependency features, with several valuable features proposed based on syntactic dependency structures [9, 10, 24]. As highlighted in [9], there is a clear correlation between the entity types and the dependency relations, which can enhance the prediction of named entities with various dependency types.

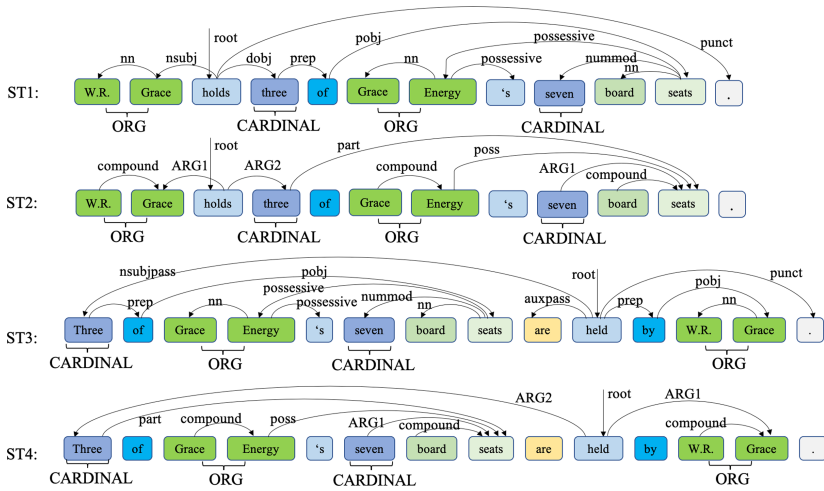


Fig. 1. Examples annotated with linguistic dependencies and named entities.

Figure 1 contains two sentences adapted from the SemEval-2015 task 18 English dataset (DM) [18], and it illustrates the relationship between language dependency structures and named entity types. Some words or phrases in the sentences are annotated with named entity types, such as **ORG** for organization and **CARDINAL** for numerals that do not fall under another type [21]. Also, the dependency relationship between words is expressed as labeled arcs. In particular, arcs in sentences ST1 and ST3 describe the syntactic dependency between words, with tags such as **nn** for noun compound modifier and **nsubj** for a nominal subject. On the other hand, the arcs in sentences ST2 and ST4 describe the semantic dependency between words, with tags such as **poss** for possession relations and **part** for measuring partitives(vague part-whole) relations.

There are several differences between syntactic and semantic dependency. First, it is obvious that the arcs and the tags in these two types of dependency convey different information. Secondly, as shown in the above example, syntactic dependency (in ST1 and ST3) always forms a dependency tree, where each word has only one head parent node. On the other hand, semantic dependency (in Fig. 1 ST2 and ST4) is a directed acyclic graph (DAG). For instance, the word **seats** in ST2 and ST4 has multiple head words **three**, **Energy**, **seven**, and **board**. Thirdly, semantic dependency structure is often preserved under simple rephrasing, whereas it is not the case for syntactic dependency. Note that ST3 and ST4 are rephrasing of ST1 and ST2, and hence the semantic dependency graph is preserved from ST2 and ST4, but the syntactic dependency tree changes from ST1 to ST3. This is an advantage of semantic dependency. Finally, each word in a syntactic dependency tree (e.g., ST1 and ST3) has an arc, but it is not the case for semantic dependency graphs (e.g., ST2 and ST4).

The long-distance dependency has been found valuable for capturing non-local structural information [5], and distributed hybrid representation deep learning models have been deployed to capture both syntactic and semantic features of words. As discussed before, syntactic dependency has been applied to increase the performance of NER, whereas we are unaware of any work on using semantic dependency for NER. Hence, the usefulness of semantic dependency and the complex long-distance interactions conveyed in such structures are unexplored, and how to use such information to enhance the word embedding in NER remains an open question.

In this work, we present the first study on leveraging semantic dependency for NER to the best of our knowledge. The significant contributions are as follows. We propose a BiLSTM-GCN-CRF model to capture the contextual information and the long-distance semantic relationship between words for enhancing the representation of the words for the NER task. Nevertheless, there is no existing NER dataset that contains semantic dependency annotations. Hence, we apply existing semantic parsing models to predict semantic dependency relations for OntoNotes 5.0 Chinese and English datasets [21], the CoNLL-2003 English dataset [23]. Finally, our extensive experiments result on these corpora shows the effectiveness of the proposed model and the advantage of semantic dependency features over syntactic dependency for NER. Also, it shows correlations between the NER performance and the semantic dependency annotations qualities.

2 Related Work

Existing works focus on learning distributed representations that capture semantic and syntactic properties of words. Besides word-level (e.g., GloVe [19], FastText [26], ELMo [20]) and character-level [2] representations, additional information is often incorporated into the representations before feeding them into context encoding layers. For example, the BiLSTM-CRF model [8] uses four types of features: spelling, context, and gazetteer features, as well as word embeddings. Some recent works make use of linguistic dependency information as an additional feature [10, 13]. Jie et al. [9] incorporate syntactic dependency structures to capture long-distance syntactic interactions between words. Aguilar et al. [1] also consider syntactic tree structures with relative and global attentions, and Nie et al. [17] incorporate syntactic information into neural models. These approaches all make use of the syntactic dependency information, but have not considered semantic dependency.

Syntactic and semantic dependency can be extracted by dependency parsing, using bi-lexicalized dependency grammar [27]. Syntactic dependency parsing reveals shallow semantic information in sentences [7]. In contrast, we could regard semantic dependency parsing (SDP), based on dependency graph parsing, as an extension of syntactic dependency parsing that characterizes more semantic relations [18]. Hence, in this paper, we study NER models with semantic dependency information.

As we are unaware of any dataset with both human annotated named entities and their semantic dependency, we need to obtain semantic dependency using existing SDP models. Through comparing the performance of existing models on SDP corpora, including the task 9 of SemEval 2016 [3], and the task 18 of SemEval 2015 [18], we selected two SDP models provided by NLP toolkits HanLP¹ and SuPar².

3 Model

This section first briefly introduces the BiLSTM-CRF model [12], which is the base for our model. Then we introduce our NER model Sem-BiLSTM-GCN-CRF, which builds a GCN on top of the linear-chain structure in BiLSTM-CRF to process complex semantic dependency graphs.

3.1 BiLSTM-CRF

The BiLSTM-CRF model turns the NER problem into a sequence labeling problem. For an input sequence $\mathbf{x} = x_1, x_2, \dots, x_i, \dots, x_n$ with n tokens, we need to predict the corresponding label sequence $\mathbf{y} = y_1, y_2, \dots, y_i, \dots, y_n$, defined according to the BIO, IOBES or IOB tagging schemes [22]. The CRF [11] tags the entity types, i.e., given \mathbf{x} , scoring the label sequence \mathbf{y} :

$$P(\mathbf{y} | \mathbf{x}) = \frac{\exp(\text{score}(\mathbf{x}, \mathbf{y}))}{\sum_{\mathbf{y}'} \exp(\text{score}(\mathbf{x}, \mathbf{y}'))}$$

The label prediction sequence has the highest output score [12], which means the final prediction is the sequence \mathbf{y} with the highest score in all output label sequences. We can get the output score by summing the transitions score and emissions score from the Bi-LSTM:

$$\text{score}(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n-1} T_{y_i, y_{i+1}} + \sum_{i=1}^n E_{i, y_i},$$

where \mathbf{T} is the transitions matrix with $T_{y_i, y_{i+1}}$ being the transition parameter from y_i to y_{i+1} , and \mathbf{E} is the emissions matrix obtained by the hidden layer of the BiLSTM with E_{i, y_i} being the score of the label y_i in the sentence's i -th position.

3.2 Sem-BiLSTM-GCN-CRF

To guide the BiLSTM-CRF model with semantic dependency information, we use GCN to process such dependency graphs. Unlike [28], which uses only adjacency matrices to capture dependency edges between words, our model also processes

¹ HanLP: <https://github.com/hankcs/HanLP>.

² SuPar: <https://github.com/yzhangcs/parser>.

dependency tag information. GCN has also been considered in [9] to incorporate syntactic dependency information. Processing semantic dependency graphs are more involved than syntactic ones, as the latter are tree-shaped, whereas the former is not necessarily so. This is why using an MLP layer instead of GCN in the model [9] improves its performance, as MLP is sufficient to capture dependency trees, but it cannot handle multi-head relationships in semantic dependency graphs. On the other hand, the dependency graphs need to be cleaned before being input to the GCN. This is because some of the edges are often erroneous or irrelevant, which is common in automatically constructed dependency graphs. To address this issue, we employ the edge-wise gating parameters for specific dependency relations. Hence, we use GCN with edge-wise gating for encoding semantic dependency, and our model combines BiLSTM with directed GCN, using CRF as the final layer. The architecture of our model Sem-BiLSTM-GCN-CRF is shown in Fig. 2.³ To represent the input, each word is represented by the concatenation \mathbf{u} of the word embedding \mathbf{w} , its context-based word vector \mathbf{v} from ELMO [20], and its character-based representation \mathbf{t} from GloVe [19] for English and FastText [6] for Chinese. That is, $\mathbf{u} = \mathbf{w} \oplus \mathbf{t} \oplus \mathbf{v}$. And then, the BiLSTM layer captures the contextual information of in \mathbf{u} .

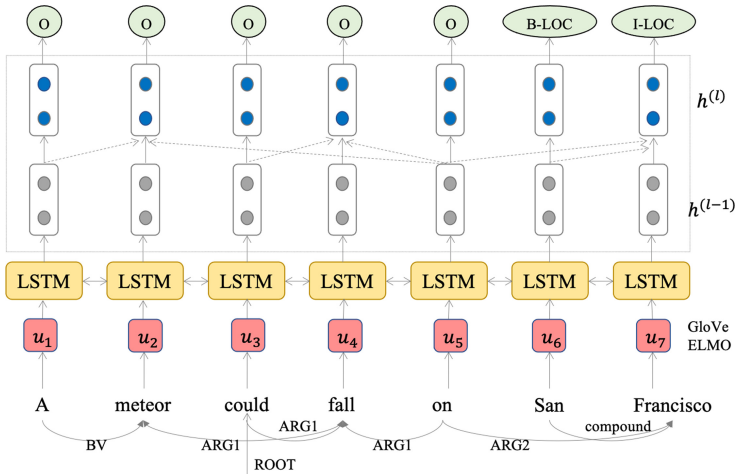


Fig. 2. BiLSTM-GCN-CRF. Dashed connections mimic the dependency edges.

Following most of the implementation for context-based GCN [9, 14, 28], we stack the GCN layer on top of LSTM to capture the semantic dependency relationship between the words to enrich the representation of words. As discussed before, some semantic-dependency prediction models use directed acyclic graphs

³ The named entity tags use the BIO labeling scheme: B-LOC labels the beginning of a location entity, I-LOC represents the inside word of the named entity, and O-LOC means outside a named entity.

(DAG) for dependency parsing. Thus in a dependency graph, each node (word) may have more than one head node (word) (as shown in Fig. 1). Using GCN allows our model to effectively capture global information and gives substantial speedup as it does not involve recursive operations that are difficult to parallelize. We treat the dependency graph as undirected and build a symmetric adjacency matrix during the GCN update. The final GCN computation is formulated as:

$$\mathbf{h}_i^{(l)} = \text{ReLU}\left(\sum_{j=1}^n A_{ij}(\mathbf{W}_1^{(l)}\mathbf{h}_j^{(l-1)} + \mathbf{W}_2^{(l)}\mathbf{h}_j^{(l-1)}w_{r_{ij}} + \mathbf{b}_{r_{ij}}^{(l-1)})\right) \quad (1)$$

where $\mathbf{h}_i^{(l)}$ is the output vector at the i -th position in the l -th layer, $A_{i,j}$ is a value in the adjacency matrix A , and $w_{r_{ij}}$ is the weight of the dependency relation $r_{i,j}$. We use parameter matrix \mathbf{W}_1 for self connections and matrix \mathbf{W}_2 for dependency. For L layers of GCN in the model, $\mathbf{h}_1^{(L)}, \dots, \mathbf{h}_n^{(L)}$ are the output word representations. Finally, the last layer is CRF.

4 Experiment

We evaluate our model’s performance on commonly used datasets by comparing it with the state-of-the-art NER models based on syntactic dependency information and analyzing the behavior of our model in different configurations.

4.1 Datasets

There are datasets with human annotated named entities and their syntactic dependency, including the Chinese and English OntoNotes 5.0 datasets [21]. We chose these datasets because they have syntactic dependency annotation, so that we can compare our model with those using such information. Yet, we are unaware of any open datasets of this type with annotated semantic dependency. Hence, in our experiments, we had to use existing prediction models to generate semantic dependency annotations. Besides OntoNotes 5.0, we also adopted the CoNLL 2003 English dataset [23].

All of these datasets contain part-of-speech tags that can be used to generate semantic dependency annotations. For example, they are used as the input feature of HanLP. Another toolkit SuPar is also used to generate the semantic dependency tags for evaluating the effect of different semantic dependency information (predicted by different models) on our performance. The English SDP models of SuPar are trained on the DM, PAS, and PSD datasets from SemEval-2015 task 18 [18], while Chinese models are trained on TEXT domain data of corpora from SemEval-2016 Task 9 [3].

4.2 Experimental Setup

We used BiLSTM-CRF [12] as the baseline model, which incorporates either syntactic or semantic dependency information. At the same time, we also feed

syntactic dependency to our BiLSTM-GCN-CRF model, denoted Syn-BiLSTM-GCN-CRF model, as another baseline for comparing the benefits of syntactic and semantic dependency. In addition, we also compared our model to the DGLSTM-CRF model [9], the state-of-the-art syntactic dependency NER model.

The system configurations are based on [9] and our parameter tunings. The hidden layer size is set to 200 in the LSTM and GCN models. We use the GloVe [19] with 100-d word embeddings for English text, and FastText [6] word embeddings for Chinese text. ELMo [20] is used for both English and Chinese texts in our experiments for deep contextualized word representations. Our models are optimized by mini-batch stochastic gradient descent, which learning rate is 0.01. The L2 regularization parameter is $1e-8$. We train for 300 epochs with a clipping rate of 3.

4.3 Main Results

Our model are compared with existing models on the three datasets, OntoNotes 5.0 Chinese (OntoNotes CN), English (OntoNotes EN), and CoNLL-2003 English (CoNLL). For each compared model, we used the numbers of LSTM/GCN layers that gave the best performance; for instance, BiLSTM(2)-CRF has a 2 LSTM layers and BiLSTM(1)-GCN(1)-CRF has 1 LSTM lay and 1 GCN layer. All the inputs are concatenated with the ELMo representations. We used SuPar to generate the semantic dependency tags. The Dependency column shows whether dependency information is not included (-), or it is provided with the datasets (gold), or it is generated. If the dependency is generated, we record the F1 score of the generating models and the text corpus they are trained on⁴. Table 1 shows the results, where those for BiLSTM-CRF and DGLSTM-CRF are from [9, 12].

On all the three datasets, Sem-BiLSTM-GCN-CRF outperforms the baseline BiLSTM-CRF and Syn-BiLSTM-GCN-CRF in most of the metrics. Note that Sem-BiLSTM-GCN-CRF and Syn-BiLSTM-GCN-CRF have similar model architecture, and the only difference is the type of dependency used. Also, on OntoNotes CN and EN, Syn-BiLSTM-GCN-CRF uses dependency information that comes from the datasets, where Sem-BiLSTM-GCN-CRF uses dependency generates. Furthermore, on OntoNotes CN and CoNLL, the performance of Syn-BiLSTM-GCN-CRF is not as good as BiLSTM-CRF, which shows the GCN encoding of syntactic dependency may not always benefit the NER task. Hence, overall it suggests the advantages of semantic dependency compared to syntactic dependency in NER.

Compared to DGLSTM-CRF, Sem-BiLSTM-GCN-CRF achieves the state-of-the-art recall performance on OntoNotes CN. Furthermore, while its performance is closely after DGLSTM-CRF with “gold” dependency, it consistently outperforms DGLSTM-CRF with generated dependency in all the other cases.

⁴ TEXT is the textbook corpus from SemEval-2016 Task 9, DM is the DELPH-IN corpus from SemEval-2015 Task 18, PAS is the Enju corpus from SemEval-2015 Task 18, PSD is the Prague corpus from SemEval-2015 Task 18, and LAS is the English Penn Treebank (PTB) corpus [15].

Table 1. Comparison on OntoNotes 5.0 Chinese/English and CoNLL-2003 English.

Dataset	Model (+ELMo)	Dependency	Prec.	Rec.	F1
OntoNotes CN	BiLSTM(1)-CRF	-	79.20	79.21	79.20
	Syn-BiLSTM(1)-GCN(1)-CRF	Gold	78.71	79.29	79.00
	Sem-BiLSTM(1)-GCN(1)-CRF	80.41 (TEXT)	78.30	81.05	79.65
	Sem-BiLSTM(1)-GCN(2)-CRF		<u>79.10</u>	80.60	<u>79.84</u>
	DGLSTM(2)-CRF	89.28	-	-	79.59
	DGLSTM(2)-CRF	Gold	78.86	<u>81.00</u>	79.92
OntoNotes EN	BiLSTM(2)-CRF	-	88.25	89.71	88.98
	Syn-BiLSTM(1)-GCN(2)-CRF	Gold	<u>89.40</u>	89.71	89.55
	Sem-BiLSTM(1)-GCN(1)-CRF	92.32 (DM)	89.22	90.10	<u>89.65</u>
	Sem-BiLSTM(1)-GCN(2)-CRF		88.78	89.90	89.34
	Sem-BiLSTM(1)-GCN(1)-CRF	93.43 (PAS)	89.18	90.04	89.61
	Sem-BiLSTM(1)-GCN(2)-CRF		88.98	89.77	89.37
	Sem-BiLSTM(1)-GCN(1)-CRF	82.64 (PSD)	88.73	90.25	89.49
	Sem-BiLSTM(1)-GCN(2)-CRF		88.00	89.10	88.55
	DGLSTM(2)-CRF	94.89	-	-	89.64
	DGLSTM(2)-CRF	Gold	89.59	<u>90.17</u>	89.88
CoNLL	BiLSTM(2)-CRF	-	92.10	92.30	92.20
	Syn-BiLSTM(1)-GCN(1)-CRF	95.86 (LAS)	91.93	92.26	92.09
	Sem-BiLSTM(1)-GCN(1)-CRF	92.32 (DM)	92.21	<u>92.49</u>	<u>92.35</u>
	DGLSTM(2)-CRF	94.00	<u>92.20</u>	92.50	92.35

This shows the competitiveness of our model compared to DGLSTM-CRF on generated dependency.

For the configurations of GCN layers, when it is increased from 1 to 2, in most of the cases, the NER performance of our model decreases. Hence, it seems GCN with a single layer is sufficient to capture the semantic dependency. We have also evaluated our model jointly with syntactic and semantic dependency features in a naive manner, which gave a suboptimal performance as compared to the semantic based NER model. It is potentially due to the inequality of the two types of information, as semantic dependency edges are often orders of magnitude more than those syntactic ones. Hence, the syntactic dependency information may not be effectively utilized. We leave the study of a joint model as future work.

4.4 Effect of Dependency Quality

The previous set of experiments shows the difference between gold-standard and predicted syntactic dependency in NER performance. To evaluate the impact of the quality of semantic dependency on the NER performance, we used the SuPar and Hanlp toolkits for comparison. As a result, semantic dependency tags

with different accuracy, measured by their F1 scores, are generated for OntoNote 5.0 and ConLL-2003 datasets. Also, SuPar and Hanlp have different data preprocessing methods, and their dataset segmentation sizes are different. Figure 3 shows the NER accuracy (NER F1 scores) of our model using semantic dependency of various quality (dependency parsing F1 scores). A strong correlation between the NER accuracy and dependency accuracy, which shows the potential of our model with high-quality dependency annotations.

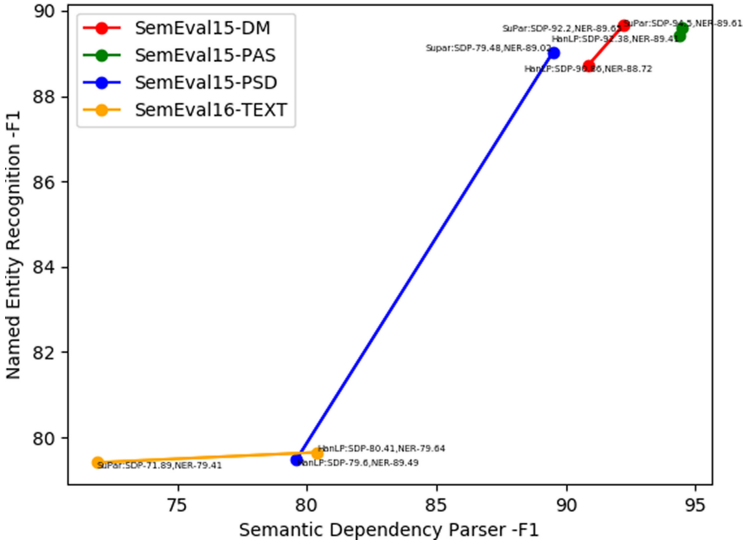
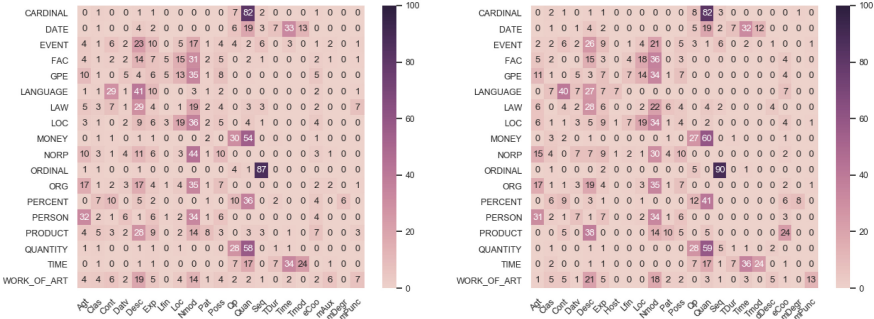


Fig. 3. Correlations between NER performance and semantic dependency quality.

5 Analysis

To further analyze why a NER model could benefit from semantic dependency information, we show the heat maps in Fig. 4) on the named entity types and the corresponding semantic dependency edges in the OntoNotes Chinese dataset. The x-axis lists various semantic dependency annotations, the y-axis is the named entity annotations, and each value shows the percentage (%) of semantic dependency edges with annotation x associated with the named entity type y .

Figure 4(a) shows the correlation between the entity types and the prediction of dependency relations on the OntoNotes Chinese test dataset. Specifically, each entry denotes the percentage of the entities with a parent dependency with a specific dependency relation. We can see that most of the entities relate to the **Desc**, **Nmod**, **Quan** dependencies. Especially the dependency relationship **Quan** (i.e., Quantity) have more than 80% of the entity type **CARDINAL** and 58% of the entity type **QUANTITY** associated to it, which suggests the semantic correlations.



(a) Heap map on training data.

(b) Heap map on NER prediction.

Fig. 4. Correlations and Percentage between the entity types (y axis) and the of semantic dependency relations (x axis) in the OntoNotes Chinese dataset. Columns with percentage less than 5% are ignored for brevity.

We can see that Fig. 4(a) and Fig. 4(b) are similar in terms of density. Moreover, both of them show consistent relationships between the entity types and the dependency relations. The comparison further illustrates that our model effectively captures the relations between the named entities and the semantic dependency.

6 Conclusion

Motivated by the relationships between semantic dependency graph and name entities, we propose a BiLSTM-GCN-CRF model to encode semantic information from the semantic dependency toolkits effectively and then enhanced the word representations. Through extensive experiments on multiple corpora, the proposed model effectively uses and captures the long-distance semantic dependency relationships between the words for improving NER performance. Our experiment analysis shows that NER benefits more from semantic dependency relations than syntactic dependency based on the same model. In addition, we find the high-quality dependency parsing will positively affect the improvement of NER. We leave studying a multi-feature fusion mechanism of syntactic and semantic of full dependencies for NER and other information extraction domains as future work.

Acknowledgements. This work is supported by the funding of State Key Laboratory of Communication Content Cognition (A32002) and Key Research and Development Program of Hubei Province (No. 2020BAB026).

References

1. Aguilar, G., Solorio, T.: Dependency-aware named entity recognition with relative and global attentions (2019)

2. Akbik, A., Blythe, D., Vollgraf, R.: Contextual string embeddings for sequence labeling. In: Proceedings of COLING-18, pp. 1638–1649 (2018)
3. Che, W., Shao, Y., Liu, T., Ding, Y.: SemEval-2016 task 9: Chinese semantic dependency parsing. In: Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, 16–17 June 2016, pp. 1074–1080. ACL (2016)
4. Dong, L., Lapata, M.: Coarse-to-fine decoding for neural semantic parsing. In: Proceedings of ACL-18, pp. 731–742 (2018)
5. Finkel, J.R., Grenager, T., Manning, C.D.: Incorporating non-local information into information extraction systems by Gibbs sampling. In: Proceedings of ACL-05, pp. 363–370 (2005)
6. Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T.: Learning word vectors for 157 languages. In: Proceedings of LREC-18 (2018)
7. Hajic, J., et al.: The CoNLL-2009 shared task: syntactic and semantic dependencies in multiple languages. In: Proceedings of CoNLL-09, pp. 1–18 (2009)
8. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging (2015)
9. Jie, Z., Lu, W.: Dependency-guided LSTM-CRF for named entity recognition. In: Proceedings of EMNLP-19, pp. 3860–3870 (2019)
10. Jie, Z., Muis, A.O., Lu, W.: Efficient dependency-guided named entity recognition. In: Proceedings of AAAI-17, pp. 3457–3465 (2017)
11. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: Proceedings of ICML-01, pp. 282–289 (2001)
12. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proceedings of NAACL-HLT-16, pp. 260–270 (2016)
13. Ling, X., Weld, D.S.: Fine-grained entity recognition. In: Proceedings of AAAI-12 (2012)
14. Marcheggiani, D., Titov, I.: Encoding sentences with graph convolutional networks for semantic role labeling. In: Proceedings of EMNLP-17, pp. 1506–1515 (2017)
15. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of English: the penn treebank. *Comput. Linguistics* **19**(2), 313–330 (1993)
16. Miwa, M., Bansal, M.: End-to-end relation extraction using LSTMs on sequences and tree structures. CoRR abs/1601.00770 (2016)
17. Nie, Y., Tian, Y., Song, Y., Ao, X., Wan, X.: Improving named entity recognition with attentive ensemble of syntactic information (2020)
18. Oepen, S., et al.: SemEval 2015 task 18: broad-coverage semantic dependency parsing. In: Proceedings of SemEval@NAACL-HLT-15, pp. 915–926 (2015)
19. Pennington, J., Socher, R., Manning, C.D.: Glove: global vectors for word representation. In: Proceedings of EMNLP-14, pp. 1532–1543 (2014)
20. Peters, M.E., et al.: Deep contextualized word representations. In: Proceedings of NAACL-HLT-18, pp. 2227–2237 (2018)
21. Pradhan, S., et al.: Towards robust linguistic analysis using OntoNotes. In: Proceedings of CoNLL-13, pp. 143–152 (2013)
22. Ratinov, L., Roth, D.: Design challenges and misconceptions in named entity recognition. In: Proceedings of CoNLL-09, pp. 147–155 (2009)
23. Sang, E.F.T.K., Meulder, F.D.: Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: Proceedings of CoNLL-03, pp. 142–147 (2003)

24. Sasano, R., Kurohashi, S.: Japanese named entity recognition using structural natural language processing. In: Proceedings of IJCNLP-08, pp. 607–612 (2008)
25. Shen, W., Han, J., Wang, J., Yuan, X., Yang, Z.: SHINE+: a general framework for domain-specific entity linking with heterogeneous information networks. *IEEE Trans. Knowl. Data Eng.* **30**(2), 353–366 (2018)
26. Wang, C., Cho, K., Kiela, D.: Code-switched named entity recognition with embedding attention. In: Proceedings of Workshop on Computational Approaches to Linguistic Code-Switching, pp. 154–158 (2018)
27. Zhang, M.S.: A survey of syntactic-semantic parsing based on constituent and dependency structures. *SCIENCE CHINA Technol. Sci.* **63**(10), 1898–1920 (2020). <https://doi.org/10.1007/s11431-020-1666-4>
28. Zhang, Y., Qi, P., Manning, C.D.: Graph convolution over pruned dependency trees improves relation extraction. In: Proceedings of EMNLP-18, pp. 2205–2215 (2018)