Intelligent Automation & Soft Computing
DOI:10.32604/iasc.2021.016320
Article

Tech Science Press

Leverage External Knowledge and Self-attention for Chinese Semantic Dependency Graph Parsing

Dianqing Liu[1,2], Lanqiu Zhang[1,2], Yanqiu Shao[1,2,*] and Junzhao Sun[3]

[1]School of Information Science, Beijing Language and Culture University, Beijing, 100083, China
[2]National Language Resources Monitoring and Research Center (CNLR) Print Media Language Branch, Beijing, 100083, China
[3]Faculty of Applied Science and Technology, Sheridan College, Oakville, ON L6H 2L1, Canada
*Corresponding Author: Yanqiu Shao. Email: yqshao163@163.com

Received: 26 December 2020; Accepted: 29 January 2021

Abstract: Chinese semantic dependency graph (CSDG) parsing aims to analyze the semantic relationship between words in a sentence. Since it is a deep semantic analysis task, the parser needs a lot of prior knowledge about the real world to distinguish different semantic roles and determine the range of the head nodes of each word. Existing CSDG parsers usually use part-of-speech (POS) and lexical features, which can only provide linguistic knowledge, but not semantic knowledge about the word. To solve this problem, we propose an entity recognition method based on distant supervision and entity classification to recognize entities in sentences, and then we integrate the category information of entities as external knowledge feature into our CSDG parser. Furthermore, there are many long sentences in some domains, which makes it difficult for the parser to deal with long-distance dependence. In this paper, we combine self-attention mechanism with Bi-LSTM, which significantly improved the performance of the parser on long texts. We also adopt Bert model to generate more powerful sentence representation and alleviate the problem of unknown words. Experiment results show that both external knowledge and self-attention are beneficial for improving the accuracy of CSDG parser and our parser achieves state-of-the-art performance in the datasets of SemEval-2016 Task 9: Chinese Semantic Dependency Parsing.

Keywords: Chinese semantic dependency graph paring; external knowledge; self-attention; Bert

1 Introduction

Chinese semantic dependency graph parsing is a deep semantic analysis task, aiming to completely analyze the modifying relationship between words in a sentence. Unlike semantic role labeling [1], which only tags the main predicates of a sentence and the arguments corresponding to each of them, each word in CSDG parsing will be given at least one modifier. CSDG is extended from the dependency tree [2]. As Chinese is a paratactic language with flexible word order and diversified functions of part-of-speech, one word sometimes depends on multiple words (non-local), and there may be a non-projection phenomenon where intersection occurs between dependent arcs (non-projection). Therefore, using directed acyclic graph instead of tree structure can illustrate the semantic information of a sentence more comprehensively. Some examples of CSDG are presented in Fig. 1.

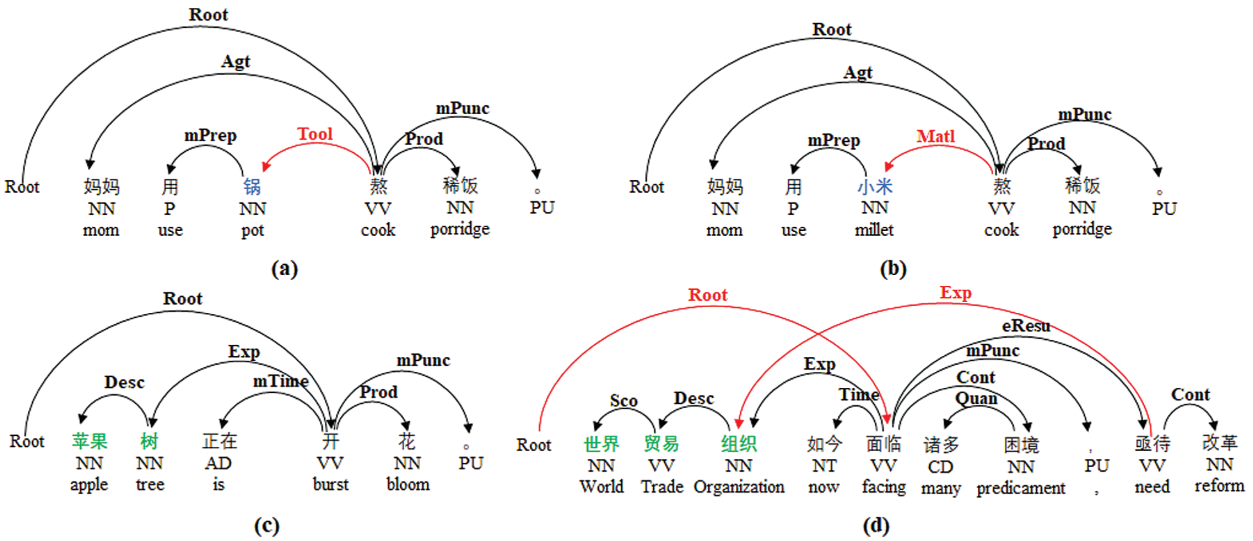

Figure 1: Examples of CSDG. (d) shows non-local ("组织") and non-projection (red lines)

Because of the similarity between semantic and syntactic dependency, many methods of syntactic dependency parsing are used as reference for semantic dependency graph parsing, which can be classified into transition-based approach [3–6] and graph-based approach [7–9]. Due to the complexity of dependency parsing, deep learning models are still indispensable to improve the performance of parsers, even though they have strong ability to automatically extract features from data, lexical features and POS tags [10,11] are still indispensable to improve the performance of parsers. However, these features can only provide linguistic knowledge.

In this paper, we focus on improving CSDG parsing with external knowledge. As CSDG is a deep semantic analysis, it actually needs a lot of background knowledge. For example, comparing the two sentences in Figs. 1a and 1b, we can find that only one word is different and their POS sequences are the same. Although both "锅" (pot) and "小米" (millet) are nouns, they have different relations with "熬" (cook) because of their different natures in reality: in these two sentences, the pot is a tool for cooking while millet is the material for cooking. Therefore, if the model lacks sufficient knowledge of these nouns in advance, it is likely to make mistakes. On the other hand, semantic dependency graph is annotated on word-level, so a named entity may be divided into several words: as is shown in Figs. 1c and 1d, "苹果树" (apple tree) and "世界贸易组织" (WTO) are both entities, and there are semantic relations (Description, Scope, etc.) among the words that make up these entities. Because there is no obvious boundary of entities in a sentence, when the model predicts the semantic relationship of several words within an entity, it needs to judge whether these words form an entity, which undoubtedly increases the difficulty of learning.

To address these challenges, we propose to incorporate entity information from knowledge base in CSDG parsing. Firstly, entity mentions in sentences are identified by distant supervision according to the knowledge base, then the label of each entity mention is acquired by entity-typing, which is equivalent to giving clear entity boundaries in sentences and adding knowledge information at the same time. After that, we add knowledge information as a feature to the model for training and prediction.

Furthermore, motivated by the success of self-attention in dealing with long-distance dependence [12] and the improvement in multiple NLP tasks with Bert [13], we adopt Bert to generate word representations instead of word2vec [14], and we combine Bi-LSTM with self-attention to further extract features. We tested our parser on the dataset of SemEval-2016 Task 9 [15]. Experiment results show that both external knowledge and self-attention are beneficial to CSDG parser and our parser outperforms the existing state-of-the-art models significantly.

2 Related Work

2.1 Chinese Semantic Dependency Graph Parsing

Recently, some progress has been made in CSDG parsing. Ding et al. [3] proposed a two-stage approach: First produce a semantic dependency tree by a transition-based model [4], and then recover the non-local dependencies using a classifier to determine whether other candidate arcs are part of the dependency graph. Wang et al. [5] designed a transition-based dependency graph parser, which parses dependency graphs directly by simply modifying the list-based arc-eager algorithm [4]. Wang et al. [6] proposed a neural transition-based parser, using Bi-LSTM and Tree-LSTM to get better representation of parsing states, based on which better transition actions can be predicted. Dozat et al. [7,8], proposed a neural graph-based model with biaffine networks, which significantly improves the accuracy on multiple dependency datasets and has become a mainstream module. Shen et al. [9] adopted biaffine networks and designed a dependency-gated cascade mechanism to parse CSDG, which achieves state-of-the-art performance in SemEval-2016 Task 9 dataset [15].

2.2 Knowledge Application

External knowledge has been applied to many natural language processing tasks, and has shown its effectiveness [16]. Niu et al. [17] constructed a binary relationship knowledge graph and applied it to opinion discrimination. Li et al. [18] proposed a recommendation system with high recommendation accuracy and good interpretability by combining the knowledge graph and deep learning method. Lu et al. [19] utilized Chinese and English knowledge resources simultaneously by sense definition alignment to disambiguation words. Kim et al. [20] collected scientific data in the field of biotechnology, analyzed the relationship between scientific terms to enhance knowledge representation, and then combined with word embedding to analyze scientific terms.

2.3 Named Entity Recognition

Named entity recognition (NER) aims to identify text spans of named entities, such as person, location and organization, which is the basis for a variety of NLP applications. NER can be regarded as a sequence labeling task. Lafferty et al. [21] proposed a conditional random field (CRF) for sequence labeling, enabling the model to learn constraints among tags to be predicted of a sentence. Lample et al. [22] combined LSTM and CRF for NER, thus improved the accuracy significantly. Recently, combining pre-trained language model (such as Bert) with CRF has become the mainstream method of NER. These methods are suitable for closed domain data with few entity categories. However due to the huge number of entity categories we defined and the small amount of available labeled data, these methods are not suitable for us. In this work, we apply a pipe-line method: first, we use distant supervision method to identify the entity mentions according to the knowledge base, and then use an entity-typing [23] model to assign categories to these entity mentions.

3 Approach

Our CSDG parsing framework incorporated with external knowledge is shown in Fig. 2. It is a pipeline method: first, the entities in a sentence are identified, and then the entity knowledge features are combined with other features for semantic parsing. The entity knowledge acquisition module and the semantic parser are trained separately.

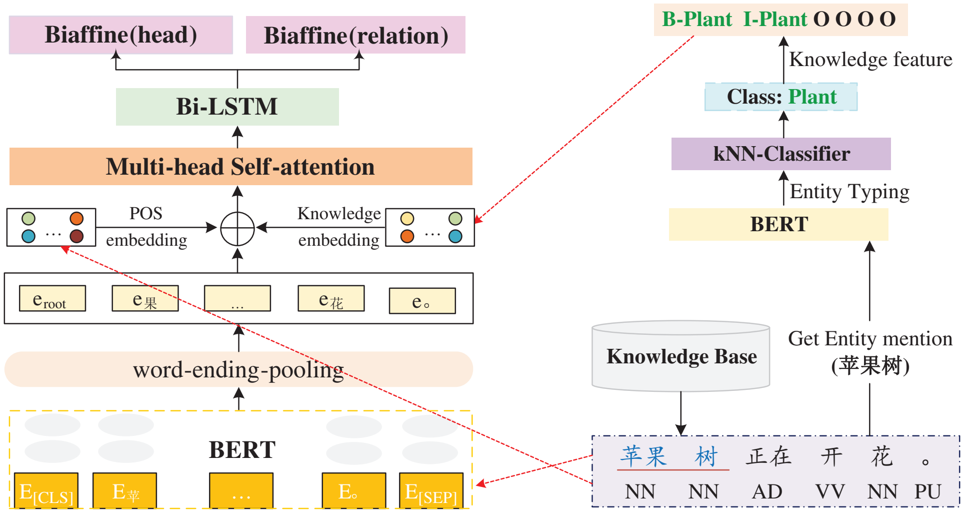

Figure 2: CSDG parsing model integrating external knowledge

3.1 Knowledge Acquisition

3.1.1 Knowledge Base

The knowledge base we use comes from OwnThink[1], which is a large general Chinese knowledge graph with 140 million triples. It contains about 11 million entities, each of which has corresponding classification labels. However, the number of entity categories is too large to use as the knowledge base is open domain. Therefore, we only use the knowledge base to identify entity mentions. Types of entity mentions are determined by entity typing module. We only keep 18 categories related to the semantic relationship defined by the Chinese semantic dependency graph, and merge categories with the similar meaning, for instance, institution and agency are both merged into the organization category.

3.1.2 Entity Recognition and Typing

We first identify the entity mentions in sentences through distant supervision technology in according with the knowledge base. Then we classify entities via an entity-typing model. Since we only keep 18 entity categories, many entity mentions are bound to be classified into 'others', but the number of entities in the 'others' categories is so large that it is impossible for us to have them all included in the dataset. To solve this problem, inspired by Perera et al. [24], we propose a two-stage entity-typing method. At the training stage, we train a multi-class text-classification model with more than 18 categories by fine-tuning Bert (Bert will be described in detail in Section 3.2.1). For entity mention $entity_i$, we use the output of Bert in position of '[CLS]' as its representation e_i^{entity}, and we feed it into a multi-layer perceptron (MLP) to determine the probability of each category:

$$e_i^{entity} = Bert(entity_i)_{CLS} \tag{1}$$

$$p_i^{*entity} = MLP\left(e_i^{entity}\right) \tag{2}$$

The loss is cross-entropy:

$$Loss_i^{entity} = -\sum\nolimits_{classes} \log p_i^{*entity} \tag{3}$$

In this way, features extracted by Bert will produce distinct representations for each entity mention of different categories. Ideally, each category will have a distinct representation from each other.

[1] https://www.ownthink.com/knowledge.html

At the prediction stage, we use the fine-tuned Bert networks to produce representation of an entity mention, and then we use a kNN classifier to determine the class of entity *ent*:

$$class(ent) = class_i, \text{ if } N_{ent,i} \geq \delta \ (0 \leq i \leq 18) \tag{4}$$

For *ent*, if the number of its similar neighbors of $class_i$ $N_{ent,i}$ exceed the threshold value δ, it will be classified into $class_i$, otherwise, it will be classified into ‘others’. We use the Euclidean distance to calculate the similarity between entity mentions and templates of $class_i$. Meanwhile, we randomly selected 20 entities from each class of development set correctly classified by the model, whose extracted features are used as templates.

In order to integrate the boundary and type of an entity into the model as external knowledge, the words that make up an entity will be tagged with the entity label by BIO tagging strategy. For example, in Fig. 2, “苹果树 (apple tree)” is an entity which is mark up with words of “苹果 (apple)” and “树 (tree)”, “苹果” is tagged as “B-Plant”, while “树” is tagged as “I-Plant”. As for words that are not part of an entity are marked as “O”.

3.2 CSDG Parser

3.2.1 Encoding

In this work, we adopt Bert to generate contextualized word representations instead of embeddings generated by word2vec. Bert is not simply used to generate static word embeddings, but rather to fine tune the whole network so as to obtain powerful contextualized word representations. Given a sentence of words with length n: $\{w_1, w_2, \ \ldots, w_n\}$, the input of Bert is a subword (English words will be cut into subwords, while Chinese words will be cut into Chinese characters), and its output is also at subword level, while CDSG parsing is of word level. To solve this problem, considering that outputs of Bert are contextualized and representations of adjacent subwords ar sequence e similar, we propose word-end-pooling: Using the representation of Chinese character at the end of a word to represent the whole word. For example, a word w_i is divided into s subwords, whose representation generated by BERT is $\{e_1, \mathrm{e}, \ \ldots, e_s\}$, the representation of w_i is expressed as:

$$ew_i \ = \ e_s \tag{5}$$

Since the input of Bert is subword sequences, another advantage of using Bert is that it can reduce the negative impact of unknown words on the parser.

Then we concatenate ew_i with POS embedding and knowledge embedding, word w_i is represented as:

$$x_i \ = \ ew_i \oplus e_i^{pos} \oplus e_i^{knowledge} \tag{6}$$

3.2.2 Multi-Head Self-attention

Word representations $\{x_1, x_2, \ \ldots, x_n\}$ are fed into a multi-head self-attention module. In attention mechanism, a query is compared with a key in a set of key-value pairs, the attention weight of query and each key-value pair is calculated, and then the weighted sum of the value using the attention weight is output. In self-attention mechanism, all of the keys (K), values (V) and queries (Q) come from the same place: the output of the previous layer in the model, each position can attend to all positions to get a more effective representation.

$$SelfAtt(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{7}$$

where d_k is the dimension of keys, and $\sqrt{d_k}$ is a scaling factor.

Multi-head attention linearly projects Q, K and V m times by different projection matrices, and then perform m self-attentions in parallel, yielding m independent representations, which are then concatenated as the representation of w_i:

$$SingleAtt_i = SelfAtt\left(QW_i^Q, KW_i^K, VW_i^V\right) \tag{8}$$

$$MultiAtt = Concate(SingleAtt_i)W^C,\ 1 \le i \le m \tag{9}$$

where W_i^Q, W_i^K and W_i^V are $d^{att} \times d^k$ matrices, $W^C \in \mathbb{R}^{md^{att} \times d^k}$, d^{att} is the hidden size of self-attention layer. The output of multi-head attention represents each position from different representation subspaces, so it is more effective.

After getting the representation of each word, we feed them to a Bi-LSTM layer to further extract features and get better representations:

$$h^{lstm} = BiLSTM(h^{att}) \tag{10}$$

where h^{att} is the self-attention encoded representation.

3.2.3 Biaffine Scorer

We use the biaffine networks [7] as classifiers to predict whether there is a modifying relationship between words (whether they form an edge) and the type of relationship respectively. Here, the process of prediction edge is described in detail. We first feed the Bi-LSTM encoded representation h_i^{lstm} into two MLPs to generate the word as a representation of the head node and the tail node respectively:

$$h_i^H = MLP^{edge-head}\left(h_i^{lstm}\right) \tag{11}$$

$$h_i^T = MLP^{edge-tail}\left(h_i^{lstm}\right) \tag{12}$$

Then we use biaffine networks and a sigmoid layer to get the probability of the edge between two words in a sentence:

$$Biaffine(x_1, x_2) = x_1^T U x_2 + W(x_1 \oplus x_2) + b \tag{13}$$

$$score_{i,j}^{edge} = Biaffine^{edge}\left(h_i^T, h_j^H\right) \tag{14}$$

$$p_{i,j}^{*edge} = Sigmoid\left(score_{i,j}^{edge}\right) \tag{15}$$

The process of relation prediction is similar to that of edge prediction. Since relation prediction is a multi-classification task, we use a softmax layer to get the probability of each relation:

$$p_{i,j}^{*rel} = Softmax\left(score_{i,j}^{rel}\right) \tag{16}$$

We use cross-entropy to calculate the losses of two classifiers $Loss_{i,j}^{edge}$ and $Loss_{i,j}^{rel}$. The loss of the entire model is the sum of them:

$$Loss = Loss_{i,j}^{edge} + Loss_{i,j}^{rel} \tag{17}$$

When predicting, the edge with a probability greater than the threshold τ will become a part of the final semantic dependency graph G:

$$edge_{i,j} \in G,\ if\ p_{i,j}^{*edge} > \tau \tag{18}$$

$$rel_{i,j} = argMax\left(p_{i,j}^{*rel}\right) \tag{19}$$

4 Experiments

4.1 Dataset and Evaluation Metrics

For entity recognition and typing, we constructed a dataset with 50 categories and more than 280,000 entities, mainly come from Sogou Thesaurus[2], to train our entity-typing model. We refer to the sememe system of HowNet[3] and retain 18 entity categories according to the property of relationships in CSDG parsing. The 18 entity categories are *human, animal, plant, place, organization, tool, vehicle, food, clothing, material, building, medicine, body part, disease, literary work, theoretical idea, psychology*, and *attribute*. For entity typing, we use macro-averages accuracy, precision, recall and F1-score as evaluation metrics.

For CSDG parsing experiments, we use the dataset provided by SemEval-2016 Task 9: Chinese semantic dependency parsing [15], which includes two distinguished corpora of NEWS and TEXTBOOKS. Statistics about the dataset are shown in Tab. 1. We follow the official evaluation setup and metrics. Evaluation metrics are labeled F-score (LF) and unlabeled F-score (UF) at the dependency arcs level, and LF is the primary evaluation metric. As for non-local (a word has multiple head nodes) dependencies, NLF and NUF are evaluated separately.

Table 1: Statistics of the dataset. #comp is the number of words that make up these entities

Domain	Dataset	#sent	#word	#entity	#comp
NEWS	Train	8301	250249	27315	33735
	Dev	534	15325	1828	2291
	Test	1233	34305	3913	5002
TEXT	Train	10754	128095	11367	14639
	Dev	1535	18257	1630	2134
	Test	3073	36097	3178	4096

4.2 Hyperparameters

The Bert model we used is the Chinese Bert-Base[4] released by Google. We linearly transformed its output to 512-dimensional and then concatenate it with other features. The POS embedding dim and knowledge embedding dim are 256, 128 respectively. We use a two-layer self-attention networks with 8 head. The Bi-LSTM we use has 3 layers with a hidden size of 600. The hidden size of biaffine networks is 600. The dropout probability of the inputs of biaffine, self-attention networks and Bi-LSTM is 0.3. Threshold τ is 0.95. In order to compare the performance of models, each experiment is trained 30 epochs with a batch size of 12.

[2] https://pinyin.sogou.com/dict/

[3] http://www.keenage.com/

[4] https://github.com/google-research/bert

5 Results and Discussion

5.1 Result of Entity Typing

Tab. 2 shows the result of entity-typing in development set. We can see that the accuracy of micro-average is good, but the results of macro average are poor due to the large number of categories and the imbalance in various training data (some classes have less training data).

Table 2: Result of entity typing

Accuracy	Macro-averages		
	Precision	Recall	F1-Score
0.9312	0.7158	0.7800	0.7242

5.2 Result of CSDG Parsing

In order to explore the influence of external knowledge and self-attention on the performance of the model, we did a series of experiments. Tab. 3 shows the experiment results, while results of previous works on the test set are used for comparison. In order to eliminate the influence of different random initialization, we use different random seeds to initialize, and for each model, the average of the five successful trials are reported in the table. For LF, standard deviations are in parentheses.

Table 3: Result of CSDG Parsing. SA for self-attention and K for knowledge

Model	NEWS				TEXTBOOKS			
	LF	UF	NLF	NUF	LF	UF	NLF	NUF
Ding et al. [3]	62.29	80.56	39.93	64.29	71.94	85.24	50.67	69.97
Wang et al. [6]	63.30	81.14	51.16	66.92	72.92	85.71	61.91	72.74
Shen et al. [9]	69.66	84.25	53.34	66.01	80.40	90.05	65.97	74.35
Our Model	**70.98(**±0.12)	**85.47**	58.85	**72.14**	**83.17**(±0.08)	**92.09**	**72.87**	**80.47**
w/o K	70.79(±0.13)	85.18	**58.95**	71.07	83.01(±0.16)	92.05	72.35	79.99
w/o SA	70.59(±0.08)	85.05	57.51	70.32	82.93(±0.11)	92.06	72.51	80.15
w/o K & SA	69.93(±0.10)	84.56	57.52	70.68	82.80(±0.12)	91.85	72.60	80.10
$\tau = 0.5$	70.47(±0.23)	84.86	53.16	66.94	83.03(±0.09)	92.01	70.49	79.33

5.2.1 Comparison with Previous Works

From Tab. 3, we can see that our model significantly outperforms all early works in both NEWS and TEXTBOOKS, especially in NLF and NUF. In NEWS, the NLF and NUF are improved by 10.33% and 9.29% respectively compared to Shen's work, while in TEXTBOOKS, the two values are improved by 10.46% and 8.23% respectively. Given the existence of non-local dependencies in CSDG data, a word may have multiple head nodes, which brings a great challenge for CSDG parsing: It is difficult for the model to accurately judge how many head nodes a word has, ultimately resulting in a negative impact on the performance of the model. In this work, we raise the accuracy of non-local dependencies to a higher level, consequently improving the value of LF as well, which is a great breakthrough.

To achieve this result, in addition to improving the structure and features of the model, we studied a peculiarity in the task itself. From Fig. 3, we can intuitively see that LF and UF are highly correlated. To prove it mathematically, we calculated the Pearson correlation coefficient of the two metrics by formula:

$$\rho(X, Y) = \frac{\sum_{i=1}^{n}(X_i - \mu_X)(Y_i - \mu_Y)}{\sqrt{\sum_{i=1}^{n}(X_i - \mu_X)^2}\sqrt{\sum_{i=1}^{n}(Y_i - \mu_Y)^2}} \tag{20}$$

where μ means the average. The Pearson correlation coefficient $\rho(UF, LF)$ is 0.98, so there is a very strong positive correlation between these two metrics.

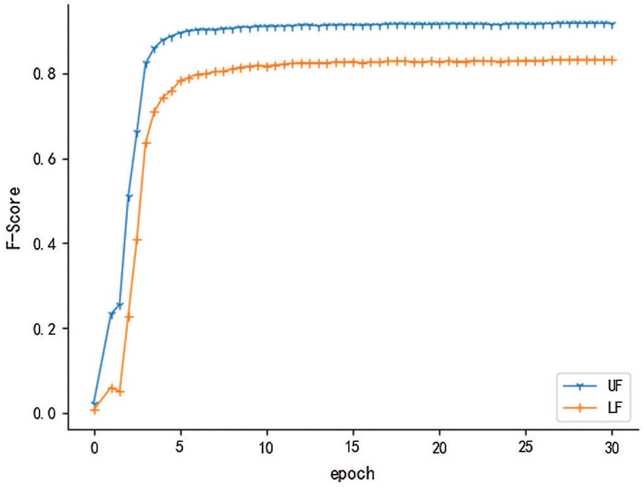

Figure 3: LF and UF in development set

Shen et al. [9] proved non-local dependencies prediction can benefit from cascading the edge and relation predictions and using gating mechanism to filter out low probability edges, motivated by which, we propose a simpler approach with the same effect: The probability of the relationship $p_{i,j}^{*rel}$ is no longer considered when determining the edge and we raise the threshold τ from 0.5 [8] to 0.95. Since this task defines more than 160 relationships, the probability of each relationship being assigned through the softmax layer is very small. Therefore, the main factor that determines whether an edge belongs to the dependency graph is $p_{i,j}^{*edge}$. So raising τ functions the same as the gating mechanism in filtering out low probability edges. The experiment results with different τ are shown in Tab. 3.

5.2.2 Influence of Knowledge

From Tab. 3, we can see that the model with self-attention and external knowledge outperforms other models in the vast majority of metrics, which proves that external knowledge information is helpful for Chinese semantic dependency parsing. At the same time, we see that the effect of external knowledge on model is not significant. According to the statistical results of the *corpus* in Tab. 1, we speculate that it is because entities in the *corpus* are sparse: Many sentences may have only one or two entities, while some may not even contain one. Therefore, knowledge cannot improve the performance of the model as much as POS. In addition, pure distant supervision may miss some entity mentions; and the macro average is not ideal judging from the result of entity typing, all this further limits the role of external knowledge. However, as mentioned above, external knowledge is crucial to understanding sentences correctly, therefore, it is essential.

Comparing the results on the test set before and after adding knowledge, we have several observations:

- For an entity composed of multiple words, such as "中国 长江 三峡 工程 开发 总 公司 (China Three Gorges Project Development Corporation)", the boundary of the entity can be determined by adding the knowledge label tagged by BIO strategy. Therefore, the range of heads of each word in the entity can be predicted correctly. While, for entities without knowledge tags, heads of multiple internal words are often outside the entity (as a whole semantic unit, an entity can only allow one head of a word to be outside its boundary).
- For some words as semantic roles in sentences (such as agent with subjective initiative and experiencer without subjective initiative), the model often makes confusions upon these roles without external knowledge, which can be avoided by adding knowledge.

5.2.3 Influence of Self-attention

In self-attention mechanism, attention is calculated between each word and all other words, which means that the maximum path length is only 1 no matter how far they are. So self-attention is thought to be able to capture long-distance dependence. In our experiment, the average length of news data is much longer than that of textbooks, and the improvement of the model with self-attention mechanism in news is greater than that in textbooks. The experimental results show that the model combining self-attention with Bi-LSTM inherits the advantages of both the two modules.

6 Conclusion

In this paper, we propose to add external knowledge as a strong feature to the Chinese semantic dependency parsing model. To recognize entities in sentences and acquire semantic knowledge contained in entity categories, we propose a two-stage entity recognition and typing method. We conduct word-end-pooling and successfully apply Bert to CSDG parsing. And we combine self-attention mechanism with Bi-LSTM to enhance the ability of CSDG parser to deal with long-distance dependence. Experiment results show that our methods are effective and we raise the accuracy of non-local dependencies to a higher level. Using these methods proposed, our parser achieves state-of-the-art performance in CSDG parsing.

Acknowledgement: We gratefully acknowledge the assistance of Peipei Sun in preparing the examples of Fig. 1. We thank Yongkang Song for excellent technical assistance. Thanks also go to OwnThink for its free knowledge graph.

Funding Statement: This research project is supported by the National Natural Science Foundation of China (61872402), the Humanities and Social Science Project of the Ministry of Education (17YJAZH068), Science Foundation of Beijing Language and Culture University (supported by "the Fundamental Research Funds for the Central Universities") (18ZDJ03), the Open Project Program of the National Laboratory of Pattern Recognition (NLPR), the Fundamental Research Funds for the Central Universities, and the Research Funds of Beijing Language and Culture University (20YCX147).

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] J. Zhou and W. Xu, "End-to-end learning of semantic role labeling using recurrent neural networks," in *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th Int. Joint Conf. on Natural Language Processing*, Stroudsburg, USA, pp. 1127–1137, 2015.

[2] J. J. Robinson, "Dependency structures and transformational rules," *Language*, vol. 46, no. 2, pp. 259–285, 1970.

[3] Y. Ding, Y. Shao, W. Che and T. Liu, “Dependency graph based chinese semantic parsing,” in *Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data*. Cham, Switzerland: Springer, pp. 58–69, 2014.

[4] J. D. Choi and A. Mccallum, “Transition-based dependency parsing with selectional branching,” in *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, USA, pp. 1052–1062, 2013.

[5] Y. Wang, J. Guo, W. Che and T. Liu, “Transition-based chinese semantic dependency graph parsing,” in *China National Conf. on Chinese Computational Linguistics Int. Sym. on Natural Language Processing Based on Naturally Annotated Big Data*, Cham, Switzerland, pp. 12–24, 2016.

[6] Y. Wang, W. Che, J. Guo and T. Liu, “A neural transition-based approach for semantic dependency graph parsing,” in *Proc. of Thirty-Second AAAI Conf. on Artificial Intelligence*, MenloPark, USA, pp. 5561–5568, 2018.

[7] T. Dozat and C. D. Manning, “Deep biaffine attention for neural dependency parsing,” in *Proc. of the 5th Int. Conf. on Learning Representations*, Toulon, France, 2017.

[8] T. Dozat and C. D. Manning, “Simpler but more accurate semantic dependency parsing,” in *Proc. of the 56th Annual Meeting of the Association for Computational Linguistics*, Stroudsburg, USA, pp. 484–490, 2018.

[9] Z. Shen, H. Li, D. Liu and Y. Shao, “Dependency-gated cascade biaffine network for chinese semantic dependency graph parsing,” in *NLPCC 2019, Lecture Notes in Computer Science*. Cham, Switzerland, pp. 840–851, 2019.

[10] C. Dyer, A. Kuncoro, M. Ballesteros and N. A. Smith, “Recurrent neural network gramma,” in *Proc. of the 2016 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Stroudsburg, USA, pp. 199–209, 2016.

[11] H. Lu, L. Hou and J. Li, “How important is pos to dependency parsing? joint pos tagging and dependency parsing neural networks,” in *CCL 2019, Lecture Notes in Computer Science*. Cham, Switzerland, pp. 625–637, 2019.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* “Attention is all you need,” in *Advances in Neural Information Processing Systems*, Cambridge, MA, USA: MIT Press, vol. 30, pp. 5998–6008, 2017.

[13] D. Jacob, M. Chang, L. Kenton and T. Kristina, “Bert: Pre-training of deep bidirectional transformers for language understanding,” in *Proc. of the 2019 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Stroudsburg, USA, pp. 4171–4186, 2019.

[14] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Advances in Neural Information Processing Systems 26 (NIPS 2013)*. Cambridge, MA, USA: MIT Press, pp. 3111–3119, 2013.

[15] W. Che, Y. Shao, T. Liu and Y. Ding, “Semeval-2016 task 9: chinese semantic dependency parsing,” in *Proc. of the 10th International Workshop on Semantic Evaluation*, Stroudsburg, USA, pp. 1074–1080, 2016.

[16] H. Zhou, T. Shen, X. Liu, Y. Zhang, P. Guo *et al.,* “Knowledge graph: A survey of approaches and applications knowledge graph,” *Journal on Artificial Intelligence*, vol. 2, no. 2, pp. 89–101, 2020.

[17] B. Niu and Y. Huang, “An improved method for web text affective cognition computing based on knowledge graph,” *Computers, Materials & Continua*, vol. 59, no. 1, pp. 1–14, 2019.

[18] T. Li, H. Li, S. Zhong, Y. Kang, Y. Zhang *et al.,* “Knowledge graph representation reasoning for recommendation system,” *Journal of New Media*, vol. 2, no. 1, pp. 21–30, 2020.

[19] W. Lu, F. Meng, S. Wang, G. Zhang, X. Zhang *et al.,* “Graph-based chinese word sense disambiguation with multi-knowledge integration,” *Computers, Materials & Continua*, vol. 61, no. 1, pp. 197–212, 2019.

[20] M. Kim, J. Kim and M. Shin, “Word embedding based knowledge representation with extracting relationship between scientific terminologies,” *Intelligent Automation & Soft Computing*, vol. 26, no. 1, pp. 141–147, 2020.

[21] J. Lafferty, A. McCallum and F. Pereira, “Conditional random fields: probabilistic models for segmenting and labeling sequence data,” in *Proc. of the Eighteenth Int. Conf. on Machine Learning*, New York, USA, pp. 282–289, 2001.

[22] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer, “Neural architectures for named entity recognition,” in *Proc. of the 2016 Conf. of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Stroudsburg, USA, pp. 260–270, 2016.

[23] H. Jin, L. Hou, J. Li and T. Dong, "Fine-grained entity typing via hierarchical multi graph convolutional networks," in *Proc. of the 2019 Conf. on Empirical Methods in Natural Language Processing and the 9th Int. Joint Conf. on Natural Language Processing*, Stroudsburg, USA, pp. 4968–4977, 2019.

[24] P. Perera and V. M. Patel, "Learning deep features for one-class classification," *IEEE Transactions on Image Processing*, vol. 28, no. 11, pp. 5450–5463, 2019.