Adversarial Domain Adaptation for Chinese Semantic Dependency Graph Parsing

Huayong Li, Zizhuo Shen, DianQing Liu, and Yanqiu Shao[⋆]

Information Science School, Beijing Language and Culture University,
Beijing 100083, China
yqshao163@163.com

Abstract. The Chinese Semantic Dependency Graph (CSDG) Parsing reveals the deep and fine-grained semantic relationship of Chinese sentences, and the parsing results have a great help to the downstream NLP tasks. However, most of the existing work focuses on parsing in a single domain. When transferring to other domains, the performance of the parser tends to drop dramatically. And the target domain often lacks the annotated data, so it is difficult to train the parser directly in the target domain. To solve this problem, we propose a lightweight yet effective domain adaptation component for CSDG parsing that can be easily added to the architecture of existing single domain parser. It contains a data sampling module and an adversarial training module. Furthermore, we present CC SD, the first Chinese Cross-domain Semantic graph Dependency dataset. Experiments show that with the domain adaptation component we proposed, the model can effectively improve the performance in the target domain. On the CCSD dataset, our model achieved state-of-the-art performance with significant improvement compared to the strong baseline model.

Keywords: Chinese Semantic Dependency Graph Parsing · Adversarial Domain Adaptation · Cross-Domain Parsing.

1 Introduction

Chinese Semantic Dependency Graph (CSDG) Parsing is one of the key technologies in Chinese natural language processing. CSDG Parsing focuses on the analysis of the deep semantic relationship between words in Chinese sentences. Unlike the restricted representation of tree structure, CSDG Parsing allow a word to have multiple parent nodes (***non-local***), and dependent arcs to cross each other (***non-projection***). Therefore, CSDG Parsing can fully and naturally represent the linguistic phenomenon of natural language[5]. Fig. 1 shows an example of CSDG Parsing.

In recent years, semantic dependency parsing has made great progress[11][18]. The neural network approach has become the mainstream technology for this task. Transition-based methods[4][6][19][20] and graph-based methods[11][10][17]

[⋆] Corresponding Author

have been successfully applied to dependency parsing tasks. Especially, the Biaffine network[11], has gained more and more attention in the field of semantic dependency parsing.

Most studies on semantic dependency parsing focused on single domain parsing, which means the data used in the research comes from only one domain. Yet, the phenomena of natural language in different domains vary greatly. Therefore, even if the model achieves a high *in-domain* score, it tends to be much worse on the *out-domain* dataset, which greatly limits the application value of the results of semantic dependence parsing[8]. In order to make semantic dependence parsing practical, it is necessary to solve the problem of domain adaptation[12]. Recently, the semi-supervised and few-shot domain adaptation has received more and more attention. However, building a robust cross-domain semantic dependency parser is still a very challenging job.

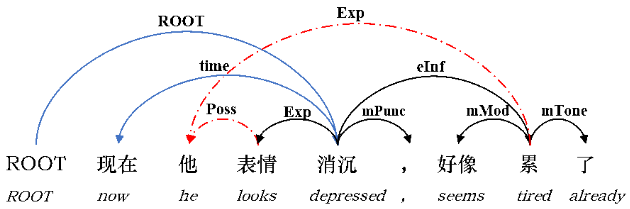

ROOT
Exp
time
eInf
Poss
Exp
mPunc
mMod
mTone
ROOT 现在 他 表情 消沉 ， 好像 累 了
ROOT now he looks depressed , seems tired already

Fig. 1. An example of CSDG Parsing. Here, “他” is the argument of “表情” and it is also an argument of “累” (*non-local*). Arc “累” → “他” and “消沉” → “表情” cross each other (*non-projection*).

In this paper, we propose a domain adaptation component for CSDG parsing by integrating shared information in different domains. We address the lack of out-domain annotation using adversarial cross-domain learning to effectively utilize the annotated data in the source domain. Since there is no cross-domain semantic dependency parsing dataset in Chinese, we release the CCSD, the first Chinese Cross-domain Semantic graph Dependency dataset. Experiments show that our model can effectively improve the performance in the target domain. Our contributions can be summarized as follows:

- We are the first to apply adversarial domain adaptation for Chinese Semantic Dependency Graph Parsing;
- We release the first Chinese Cross-domain Semantic Graph Dependency Parsing dataset, the dataset and code are available on Github[1];
- We propose two data sampling strategies and three adversarial learning methods and analyze their performance;

[1] https://github.com/LiangsLi/Domain-Adaptation-for-Chinese-Semantic-Dependency-Graph-Parsing

2 Related Work

2.1 Semantic Dependency Graph Parsing

In order to be able to express more linguistic phenomena, [5] extended semantic dependency tree to graph structure. To parse semantic dependency graphs, [9][18] proposed a neural transition-based approach, using a variant of list-based arc-eager transition algorithm, and [11] proposed a graph-based model with Biaffine attention mechanism.

2.2 Domain Adaptation

Most of studies on domain adaptation in the field of dependency parsing are about syntactic parsing. In the domain adaptation track of CoNLL 2007 shared task [13], [3] applied a tree revision method which learns how to correct the mistakes made by the base parser on the adaptation domain, and [14] used two models to parse unlabeled data in the target domain to supplement the labeled out-of-domain training set. [21] proposed producing dependency structures using a large-scale HPSG grammar to provide general linguistic insights for statistical models, which achieved performance improvements on out-domain tests. [8] proposed a data-oriented method to leverage ERG, a linguistically-motivated, hand-crafted grammar, to improve cross-domain performance of semantic dependency parsing.

In other tasks, [12] proposed an adversarial approach to domain adaptation. [7] proposed adversarial multi-criteria learning for Chinese word segmentation. [16] applied adversarial domain adaptation to the problem of duplicate question detection across different domains.

3 Method

3.1 Deep Biaffine Network for Single Domain CSGD Parsing

Before introducing cross-domain parsing method, we need to review the current best single domain parsing network. Following the work of [11], we use the Biaffine network as our general architecture for single domain CSGD parsing.

Representation Layer Each word x_i is represented as the concatenation of word embedding e_i^{word}, POS tag embedding e_i^{pos} and chars' representation h_i^{char}:

$$x_i = e_i^{word} \oplus e_i^{pos} \oplus h_i^{char} \tag{1}$$

Feature extraction layer We use Bi-LSTM as feature extraction layer. In order to improve the efficiency of training, Highway mechanism [22] is used:

$$h_t^{lstm} = HighwayLSTM\left(x_i; W_H; b_H\right) \tag{2}$$

Where, h_t^{lstm} represents the output of the Highway LSTM, and W_H,b_H represents the parameters of the Highway LSTM, respectively.

Biaffine Scorer Layer We use Biaffine network [10] to predict dependency edges and dependency labels respectively. For predicting dependency edge, we first feed the Bi-LSTM encoded representation h_i^{lstm} into two single-layer feed-forward networks (FNN) to get the head representation and the dependent representation.

$$h_i^{edge-head} = FNN^{edge-head}\left(h_i^{lstm}\right) \tag{3}$$

$$h_i^{edge-dep} = FNN^{edge-dep}\left(h_i^{lstm}\right) \tag{4}$$

Then, we use Biaffine transformation (Eq. 5) to obtain the scoring matrix of all possible edges in a sentence. Finally, we calculate the probability of each edge.

$$Biaffine\left(x_1, x_2\right) = x_1^T U x_2 + W\left(x_1 \oplus x_2\right) + b \tag{5}$$

$$s_{i,j}^{edge} = Biaffine^{edge}\left(h_i^{edge-dep}, h_j^{edge-head}\right) \tag{6}$$

$$p_{i,j}^{*edge} = sigmoid\left(s_{i,j}^{edge}\right) \tag{7}$$

The method of dependency label prediction is nearly similar to that of dependency edge prediction. It's worth noting that we use softmax function to calculate the label probability (Eq. 8).

$$p_{i,j}^{*label} = softmax\left(s_{i,j}^{label}\right) \tag{8}$$

3.2 Cross-Domain CSDG Parsing Method

The task of Cross-Domain CSDG Parsing is to transfer the CSDG parser from the source domain to the target domain. Since the amount of annotated training data in the target domain is very limited, some of which may not even have annotated training data. Therefore, the *Shared-Private* (or *Global-Local*) network architecture [7] is difficult to apply to our task because there is not enough data to train the private (or local) module for each target domain. In this paper, we propose an effective domain adaptation component that can be easily added to the single domain parser described in 3.1. As shown in Fig. 2, the component consists of two parts: *data sampling module* and *adversarial training module.*

Data Sampling Module In cross-domain parsing, we first need to decide how to sample data in two domains. Here we propose two approaches of data sampling, uniform and proportional sampling [15]. We choose which domain the data comes from based on the probability of source P_{source} and the probability of target P_{target}.

In uniform sampling, P_{source} and P_{target} are equal, both equal to 0.5. In proportional sampling, the probability of sampling a domain is defined as follow:

$$P_{source} = \frac{N_{source}}{N_{source} + N_{target}} \tag{9}$$

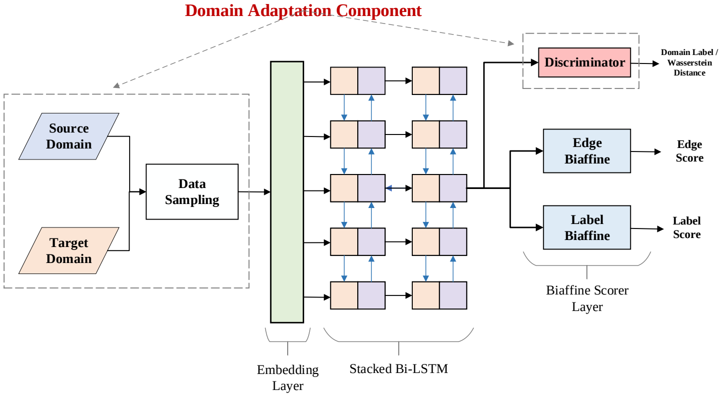


Fig. 2. Architecture of cross-domain CSDG paring with adversarial training

$$P_{target} = \frac{N_{target}}{N_{source} + N_{target}} \tag{10}$$

Where N_{source} indicates the size of source domain training set, N_{target} indicates the size of target domain training set.

Adversarial Training Module The adversarial training [1][7][12][16] module contains a discriminator and three different adversarial training strategies. We adopt two main approaches to achieve adversarial learning. One is based on domain classification and the other is based on *Wasserstein distance*. In particular, the classification-based approach contains two different adversarial training strategies.

In the classification-based approach, we adopt a domain classifier as the discriminator which is optimized to correctly distinguish which domain the data comes from. The loss of the discriminator needs to be minimized when training the discriminator as follows:

$$\min_{\theta^{dis}} J_{dis}\left(\theta^{dis}\right) = \sum_{i=1}^{n}\sum_{j=1}^{m} \log p\left(L_j | X_i, \Theta^f, \theta^{dis}\right) \tag{11}$$

where ^{findicates} the parameters of feature extraction layer, ^{dis} indicates the parameters of discriminator.

In contrast, the feature extraction layer is optimized to confuse the discriminator in order to make the Bi-LSTM encoded feature representation more general. Therefore, we need to maximize the adversarial loss to optimize the parameters of feature extraction layer. Inspired by [7][12][16], we use two different adversarial training strategies that contain two different adversarial losses .

The first is the cross-entropy loss training strategy[2], which is to maximize the cross-entropy of the discriminator when training the parameters of feature

[2] This method is also known as *Gradient Reversal*

extraction layer:

$$\max_{\theta^f} J_{adv}\left(\Theta^f\right)=\sum_{i=1}^{n}\sum_{j=1}^{m}\log p\left(L_j|X_i,\theta^f,\theta^{dis}\right) \tag{12}$$

The second is the entropy training strategy, which is to maximize the entropy of domain distribution predicted by discriminator when training the parameters of feature extraction layer:

$$\max_{o^f} J_{adv}\left(\Theta^f\right)=\sum_{i=1}^{n}\sum_{j=1}^{m}H\left(p\left(L_j|X_i,\Theta^f,\Theta^{dis}\right)\right) \tag{13}$$

$$H(p)=-\sum_{i}p_i\log p_i \tag{14}$$

where $H(p)$ is the entropy of domain distribution p.

In the Wasserstein approach [2][1], we minimize the Wasserstein distance between source domain distribution P_s and target domain distribution P_t in order to train discriminator.

$$W\left(P_s,P_t\right)=\sup_{\|f\|_L\leq 1}E_{x\sim P_s}[f(x)]-E_{x\sim P_t}[f(x)] \tag{15}$$

$$\min_{\Theta^{dis}} J_{dis}\left(\Theta^{dis}\right)=\min W\left(P_S,P_t\right) \tag{16}$$

where Θ^{dis} indicates the parameters of discriminator, and f is a *Lipschitz-1 continuous function.*

In contrast, the feature extraction layer needs to be trained by maximizing Wasserstein distance:

$$\max_{\Theta^f} J_{adv}\left(\Theta^f\right)=\max W\left(P_s,P_t\right) \tag{17}$$

According to [1], the adversarial training based on Wasserstein distance can obtain more stable training process.

3.3 Jointly Training

First, we calculate the loss of the parser J_{parser}, which consists of two parts, the loss of edge Biaffine J_{edge} and the loss of label Biaffine J_{label}:

$$J_{parser}\left(\Theta^p\right)=\beta J_{label}\left(\Theta^p\right)+(1-\beta)J_{edge}\left(\Theta^p\right) \tag{18}$$

where Θ^p indicates the parameters of CSDG parser (note that the parameters of the discriminator are not included in Θ^p), β is the combined ratio of two losses. The loss of edge Biaffine and the loss of label Biaffine is calculated as follows:

$$J_{edge}\left(\Theta^p\right)=-p_{i,j}^{edge}\log p_{i,j}^{*edge}-\left(1-p_{i,j}^{edge}\right)\log\left(1-p_{i,j}^{*edge}\right) \tag{19}$$

Algorithm 1 Adversarial domain adaptation learning for CSDG parsing

```
Input: source domain data and a target domain data $D \in (X_s \cup X_t)$
Hyper-parameters: the learning rates of parser and adversarial competent $\alpha_1$, $\alpha_2$; adversarial interpolation $\lambda$
Parameters to be trained: $\Theta^p$,$\Theta^f$,$\Theta^{dis}$
Sampling probability:$P_{sampling}$
Loss function: $J_{parser}$,$J_{dis}$,$J_{adv}$

while $\Theta^p$ and $\Theta^f$ not converge do
    Pick a batch of data from $D$ according to $P_{sampling}$
    Calculate the parser loss $J_{parse}$ using $\theta^p$ for this batch
    Calculate the adversarial loss $J_{adv}$ using $\Theta^{dis}$, $\Theta^f$ for this batch
    $\Theta^p = \Theta^p - \alpha_1 \nabla^p_\Theta J_{parser}$
    $\Theta^{dis} = \Theta^a - \alpha_2 \nabla^a_\Theta J_{dis}$
    $\Theta^p = \Theta^f + \alpha_1 \nabla^f_\Theta J_{adv}$
```

$$J_{label}\left(\Theta^p\right) = -\sum_{label} \log p_{i,j}^{*label} \tag{20}$$

Finally, the entire model is optimized by the a jointly losses as follow:

$$J(\Theta) = J_{parser}\left(\theta^p\right) + J_{dis}\left(\theta^{dis}\right) + \lambda J_{adv}\left(\Theta^f\right) \tag{21}$$

Where λ is the weight that controls the interaction of the loss terms. The details of training procedure are described as Algorithm 1.

4 CCSD Dataset

Since there is no cross-domain semantic dependency parsing dataset in Chinese, we present a Chinese Cross-domain Semantic Dependency dataset, named ***CCSD***. It contains one source domain and four target domains.

4.1 Corpus Collection

The data sources for each domain are as follows:

Source Domain Source domain contains a lot of balanced corpus. The sentences in source domain are selected from the SemEval-2016 Task 9 dataset [5] and the textbook *Boya Chinese* (Chinese: 博雅汉语).

Target Domain Target area consists of four different domains. The data sources for different target domains are as follows:

- **Novel**. Novel domain contains 1562 sentences selected from the short story *the Little Prince* (Chinese: 小王子) and 3438 sentences selected from novel *Siao Yu* (Chinese: 少女小渔).

- **Drama**. Drama domain contains 5000 sentences selected from drama *My Own Swordsman* (Chinese: 武林外传).
- **Prose**. Prose domain contains 5000 sentences selected from the prose collection *Cultural Perplexity in Agonized Travel* (Chinese: 文化苦旅). This domain contains a lot of rhetorical figures.
- **Inference**. Inference domain contains 22,308 sentences, selected from the *CNLI* (Chinese Natural Language Inference) dataset. Compared with other domains, the sentences in this domain are much simpler.

4.2 Dataset Construction and Statistics

For each domain, we divide the data into training set, validation set, and test set. The detailed statics are described in Table 1.

Table 1. Statistics about the dataset. The table shows the number of sentences. It is worth noting that there is no annotated training data in the *Inference* domain.

	Domain	**Train Set**	**Dev Set**	**Test Set**	**Unannotated**
Source	*Source*	24,003	2,000	2,000	——
Target	*Novel*	3,000	1,000	1,000	——
	Prose	3,000	1,000	1,000	——
	Drama	3,000	1,000	1,000	——
	Inference	——	2,000	2,000	18,308

It is worth noting that we have manually annotated all the data in the Source domain, Novel domain, Prose domain and the Drama domain. But for the data in the Inference domain, we only annotated the data of the validation set and the test set. However, Inference domain contains a lot of unannotated data.

Table 2. Statistics of the four target domains. **Ave-len** represents the average sentence length of the corpus of each domain. **Non-local** represents the proportion of non-local phenomena, and **Non-projective** represents the proportion of non-projective phenomena. **Unigrams** and **Bigrams** show the proportion of n-grams that shared between the source and each target domains.

Domain	**Ave-len**	**Non-local**	**Non-projective**	**Unigrams**	**Bigrams**
Novel	12.49	12.02%	5.23%	97.42%	35.48%
Drama	10.16	6.47%	2.40%	97.06%	30.06%
Prose	17.22	13.98%	4.14%	94.61%	28.88%
Inference	11.72	5.29%	2.22%	96.98%	25.38%

As shown in Table 2, We compared the distribution of data in different domains. According to the statistical results, the data in the Inference domain is the simplest, and the data distribution of Novel domain and Drama domain is closest to the source domain.

5 Experiments

5.1 Setup

Baselines and Evaluation Metrics we use two baseline models: Non-transfer model and Pre-trained model. Non-transfer model is trained directly in the target domain. Pre-trained model is to train a fundamental model using source domain data, and then fine-tune model using target domain data. We use the labeled attachment score (LAS) in the target domain as evaluation metric.

Hyperparameters The dimension of LSTM is 400. We dropout word embedding with 20% probability and the inputs of LSTM and Biaffine network with 33% probability. The joint optimization ratio of biaffine classifiers is 0.5. We set the interpolation coefficients λ of adversarial loss to 0.1. Other configurations are same as [11].

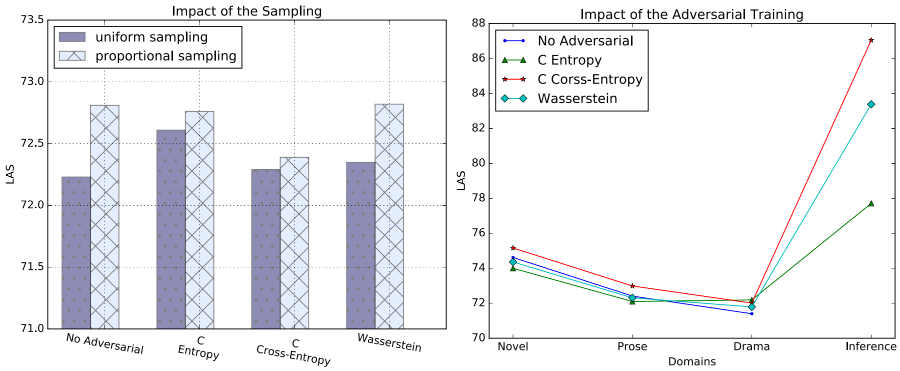
5.2 Overall Results

Table 3 shows the experiment results of our proposed models on test sets of target domains. Note that we do not show the metrics in source domain, because in this task we use the metrics of the target domain to measure the pros and cons of the model.

Since there is no training data in Inference domain, we only use the unsupervised data of the target domain to train the discriminator, and train the parser with the data of the source domain. In Novel Domain, our best model gains 6.41% and 1.67% improvement compared with two strong baseline models. In Prose Domain, our best model gains the 4.77% and 2.79% improvement

Table 3. Results of proposed models on target domains. In the Sampling column, U, P indicate uniform sampling, and proportional sampling. In the Adv Learning column, E, CE, W indicate entropy method, cross-entropy method, Wasserstein distance method.

Model	**Sampling**	**Adversarial**	**Result(LAS)**			
			novel	*prose*	*drama*	*inference*
Non-transfer	–	–	68.75	68.22	68.01	–
Pre-trained	–	–	73.49	70.2	69.42	87.1
Our model	U	–	74.51	71.97	71.22	–
	U	E	73.85	72.26	71.72	**87.47**
	U	CE	73.52	72.22	71.14	84.1
	U	W	74.13	71.95	70.96	86.13
	P	–	74.62	72.41	71.4	–
	P	E	74	72.1	**72.19**	77.71
	P	CE	**75.16**	**72.99**	72.01	87.05
	P	W	74.36	72.31	71.79	83.38



(a) (b)

Fig. 3. Results of the comparative experiment.(a) is comparison between different sampling strategies. (b) is comparison between different adversarial training strategy

respectively. In Drama Domain, our best model gains the 4.18% and 2.77% improvement respectively. In Inference domain, our best model gains the 0.37% improvement on LAS.

The experimental results can strongly prove the ability of our proposed models. Due to the differences of language characteristics, the optimal domain adaption methods of each domain are not same. According to the experimental results, the proportional sampling and adversarial learning based on cross-entropy method achieves the highest LAS in most cases.

Analysis on the Data Sampling Strategy Figure 3(a) studies the impact of sampling strategies on parser performance. In order to better compare the impact of sampling strategies, we performed experiments of uniform sampling and proportional sampling in four target areas based on different adversarial strategies. Then we average the results of the four target domains.

As shown in Figure 3(a), no matter which adversarial strategy is used, the probability sampling method is obviously better than the uniform sampling method. We believe that this is due to the limited amount of training data in the target domain. When using uniform sampling, the model is easy to overfit the training data, thus affecting the generalization ability of the model.

Analysis on the Adversarial Training Strategy Figure 3(b) studies the impact of the choice of adversarial strategy. All of the above comparison experiments were done under proportional sampling.

As shown in Figure 3(b), the performance of the model using the classification-based cross-entropy method is the best in all four areas. And its performance is significantly better than the model without the adversarial strategy. It is worth noting that on Novel and Prose domain, the performance of the model using the other two adversarial is not much different or slightly worse than that of the model without the adversarial strategy. This is because the data distribution of Novel and Prose is very close to the source domain, so the effect of adversarial is less obvious. Because the sentences in Inference domain are relatively simple, even in the unsupervised case, the model still achieves good performance after using the adversarial strategy.

Conclusions as a result, the adversarial strategy, especially classification-based cross-entropy method, can significantly improve the performance of the model in the target domain.

6 Conclusions

Performance of Chinese Semantic Dependency Graph Parser tends to drop significantly when transferring to new domains, especially when the amount of data in the target domain is small. In order to solve this problem, we propose an effective domain adaptation component, which has two parts: data sampling module and adversarial training module. Since there is no open Chinese cross-domain semantic dependency graph parsing dataset, we release a new dataset CCSD. Experiments show that our model performance is significantly improved compared to the strong baseline model. On the CCSD dataset, the model using proportional sampling and cross-entropy adversarial learning achieves the highest LAS score in most cases.

Acknowledgement

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).

References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein gan. arXiv preprint arXiv:1701.07875 (2017)
2. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International conference on machine learning. pp. 214–223 (2017)
3. Attardi, G., Dell’ Orletta, F., Simi, M., Chanev, A., Ciaramita, M.: Multilingual dependency parsing and domain adaptation using desr. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL). pp. 1112–1118 (2007)
4. Che, W., Liu, Y., Wang, Y., Zheng, B., Liu, T.: Towards better ud parsing: Deep contextualized word embeddings, ensemble, and treebank concatenation. arXiv preprint arXiv:1807.03121 (2018)
5. Che, W., Zhang, M., Shao, Y., Liu, T.: Semeval-2016 task 9: Chinese semantic dependency parsing. pp. 378–384 (06 2012). https://doi.org/10.18653/v1/S16-1167

6. Chen, D., Manning, C.: A fast and accurate dependency parser using neural networks. In: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). pp. 740–750 (2014)
7. Chen, X., Shi, Z., Qiu, X., Huang, X.: Adversarial multi-criteria learning for chinese word segmentation. arXiv preprint arXiv:1704.07556 (2017)
8. Chen, Y., Huang, S., Wang, F., Cao, J., Sun, W., Wan, X.: Neural maximum subgraph parsing for cross-domain semantic dependency analysis. In: Proceedings of the 22nd Conference on Computational Natural Language Learning. pp. 562–572 (2018)
9. Ding, Y., Shao, Y., Che, W., Liu, T.: Dependency graph based chinese semantic parsing. In: Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data, pp. 58–69. Springer (2014)
10. Dozat, T., Manning, C.D.: Deep biaffine attention for neural dependency parsing. arXiv preprint arXiv:1611.01734 (2016)
11. Dozat, T., Manning, C.D.: Simpler but more accurate semantic dependency parsing. arXiv preprint arXiv:1807.01396 (2018)
12. Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., Lempitsky, V.: Domain-adversarial training of neural networks. The Journal of Machine Learning Research **17**(1), 2096–2030 (2016)
13. Nivre, J., Hall, J., Kübler, S., McDonald, R., Nilsson, J., Riedel, S., Yuret, D.: The conll 2007 shared task on dependency parsing. In: Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL) (2007)
14. Sagae, K., Tsujii, J.I.: Dependency parsing and domain adaptation with data-driven lr models and parser ensembles. In: Trends in Parsing Technology, pp. 57–68. Springer (2010)
15. Sanh, V., Wolf, T., Ruder, S.: A hierarchical multi-task approach for learning embeddings from semantic tasks. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 33, pp. 6949–6956 (2019)
16. Shah, D.J., Lei, T., Moschitti, A., Romeo, S., Nakov, P.: Adversarial domain adaptation for duplicate question detection. arXiv preprint arXiv:1809.02255 (2018)
17. Wang, W., Chang, B.: Graph-based dependency parsing with bidirectional lstm. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). vol. 1, pp. 2306–2315 (2016)
18. Wang, Y., Che, W., Guo, J., Liu, T.: A neural transition-based approach for semantic dependency graph parsing. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)
19. Yu, J., Elkaref, M., Bohnet, B.: Domain adaptation for dependency parsing via self-training. In: Proceedings of the 14th International Conference on Parsing Technologies. pp. 1–10 (2015)
20. Zhang, X., Du, Y., Sun, W., Wan, X.: Transition-based parsing for deep dependency structures. Computational Linguistics **42**(3), 353–389 (2016)
21. Zhang, Y., Wang, R.: Cross-domain dependency parsing using a deep linguistic grammar. In: Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP. pp. 378–386 (2009)
22. Zilly, J.G., Srivastava, R.K., Koutník, J., Schmidhuber, J.: Recurrent highway networks. In: Proceedings of the 34th International Conference on Machine Learning-Volume 70. pp. 4189–4198. JMLR. org (2017)