Unsupervised Domain Adaptation Method with Semantic-Structural Alignment for Dependency Parsing

Boda Lin[1], **Mingzheng Li**[1], **Si Li**[1*] **and Yong Luo**[2*]
[1]School of Artificial Intelligence, Beijing University of Posts and Telecommunications
[2]School of Computer Science, Wuhan University
{linboda, limz, lisi}@bupt.edu.cn luoyong@whu.edu.cn

Abstract

Unsupervised cross-domain dependency parsing is to accomplish domain adaptation for dependency parsing without using labeled data in target domain. Existing methods are often of the pseudo-annotation type, which generates data through self-annotation of the base model and performing iterative training. However, these methods fail to consider the change of model structure for domain adaptation. In addition, the structural information contained in the text cannot be fully exploited. To remedy these drawbacks, we propose a Semantics-Structure Adaptative Dependency Parser (SSADP), which accomplishes unsupervised cross-domain dependency parsing without relying on pseudo-annotation or data selection. In particular, we design two feature extractors to extract semantic and structural features respectively. For each type of features, a corresponding feature adaptation method is utilized to achieve domain adaptation to align the domain distribution, which effectively enhances the unsupervised cross-domain transfer capability of the model. We validate the effectiveness of our model by conducting experiments on the CODT1 and CTB9 respectively, and the results demonstrate that our model can achieve consistent performance improvement. Besides, we verify the structure transfer ability of the proposed model by introducing Weisfeiler-Lehman Test.

1 Introduction

Dependency parsing is to extract the dependency structure of a sentence that shows its grammatical structure and the relationships between "head" words and associated "dependent" words. Dependency parsing can provide the syntactic structure information for sentence, which can be used to enhance other Natural Language Processing task such as Named Entity Recognition (Vakare et al., 2019) and sentence Semantic Similarity (Jie et al., 2017). Existing in-domain dependency parsing model has achieved promising performance in the domains that have abundant labeled data such as news and magazines (Ma et al., 2018). But in some other domains such as web blogs and novels, the performance of the dependency parser is often unsatisfactory due to the label deficiency issue. Since the cost of dependency labeling is extremely high, some cross-domain dependency parsing (CDP) approaches have been developed in recent years.

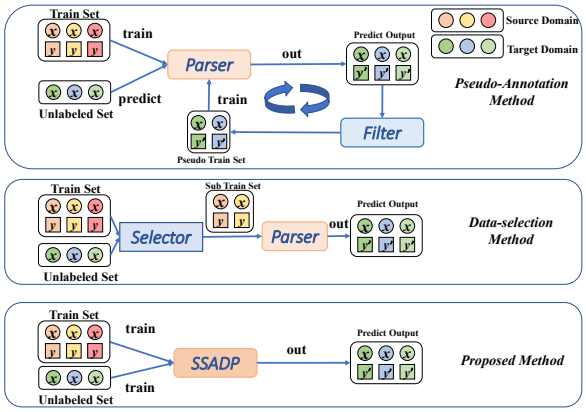

Figure 1: Difference between previous methods and proposed method. The proposed SSADP method is more advantageous in that knowledge transfer and parsing are conducted in a joint manner.

CDP is to use the abundant label information of source domain to train a dependency parsing model that can be used in the target domain. According to the different labeling settings of target domain data, CDP can be divided into semi-supervised and unsupervised (Peng et al., 2019). In the semi-supervised setting, the target domain is assumed to have a few labeled data, while in the unsupervised scenario, the target domain only has some unlabeled data. We focus on the latter in this paper.

There are mainly two ways to achieve unsupervised CDP: (1) pseudo-annotation iterative methods, such as self-training (Yu et al., 2015), co-training (Nivre et al., 2007) and tri-training (Dredze

*Corresponding authors

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 2158–2167
November 7–11, 2021.

et al., 2007), which mainly rely on a model trained on the source domain to generate credible pseudo-labeled data in the target domain; (2) data selection methods, which mainly focus on selecting a sub-dataset in the source domain that is similar to the target domain for training (Plank and Van Noord, 2011; Khan et al., 2013). These approaches focus on designing training or data filtering strategies.

However, feature-based transfer is not considered being taken into the previous two ways, which means the transfer is separated with the parsing and the hidden feature produced by parser is not used for domain adaptation. Besides, previous methods only use the unlabeled data for filtering, and more information (such as feature statistics) contained in such data can not be fully exploited.

To remedy this drawback, we propose a Semantic-Structural Adaptative Dependency Parser (SSADP) to accomplish transfer and parsing simultaneously for unsupervised CDP without using pseudo-annotation and data-selection strategy. The proposed method is graph-based, which means that the words are treated as nodes and sentence as a graph. The cross-domain transfer is mainly achieved by: 1) using Biaffine (Dozat and Manning, 2017) to extract semantic feature and query-key CNN (QKCNN) to extract structural features from the domain data respectively to enhance the ability in describing the domain of model; 2) integrating metric-learning method Local Maximum Measure Discrepancy (LMMD) (Zhu et al., 2020) and Graph Attention Network (GAT) (Velicković et al., 2017) to align the domains according to characteristics of the extracted semantic and structural features.

Effectiveness of the proposed SSADP is demonstrated by the extensive experiments on the quite new Chinese Open Dependency Treebank 1.0 (CODT1) (Li et al., 2019) and Chinese Tree Bank 9.0 (CTB9) (Xue et al., 2016) datasets, where we propose a data division standard for CTB9. We introduce the Weisfeiler-Lehman Test to verify that the proposed SSADP has the ability of transfer structural information.

Our main contributions of this paper are summarized as follows:

1) We propose a model termed SSADP for unsupervised CDP by performing transfer and parsing simultaneously, and our model does not rely on pseudo-annotation and data selection.

2) To the best of our knowledge, this is the first work that applies metric-based domain adaptation to parsing and especially the idea of graph structure alignment is novel in domain adaptation.

3) The experiments on CODT1 and CTB9 demonstrate the proposed SSADP is effective and we improve Weisfeiler-Lehman Test by the Jaccard Distance to verify the structure transfer ability of SSADP.

2 Related Work

2.1 Semi-supervised Cross-Domain Dependency Parsing

The main idea of semi-supervised cross-domain dependency parsing is to make full use of the domain information of the data while using a small amount of labeled data in target domain for supervised parsing. Sato et al. (2017) proposed a parser within adversarial domain adaptation to utilize the labeled data in target domain. Li et al. (2019) propose adding domain embedding to achieve semi-supervised cross-domain dependency parsing.

2.2 Unsupervised Cross-Domain Dependency Parsing

The previous works on unsupervised cross-domain dependency parsing can be divided into two main categories: pseudo-annotation self-iterative method and data-selection method. Yu et al. (2015) firstly proposed to present cross-domain dependency parsing via self-training. Lien et al. (2015) proposed another self-training method within K-means cluster for cross-domain dependency parsing. Cohen et al. (2012) adopted co-training within using Latent Dirichlet Allocations (LDA) to learn a domain-specific selectional preferences.

2.3 Metric-based Domain Adaptation

Metric domain adaptation is a commonly used unsupervised domain adaptation method. There is a usual way to achieve unsupervised domain adaptation by using the Maximum Mean Discrepancy (MMD) distance. MMD is used with kernel method to compare the difference between different sample distribution in Reproducting Kernel Hilbert Space (RHKS) (Borgwardt et al., 2006). It also can be used in neural network to measure the discrepancy between the hidden representations of source domain and target domain (Ghifary et al., 2014) . Beyond the original MMD, there are many other variant of MMD used in unsupervised domain adaptation (Tzeng et al., 2014; Yan et al., 2017; Zhu et al., 2020).

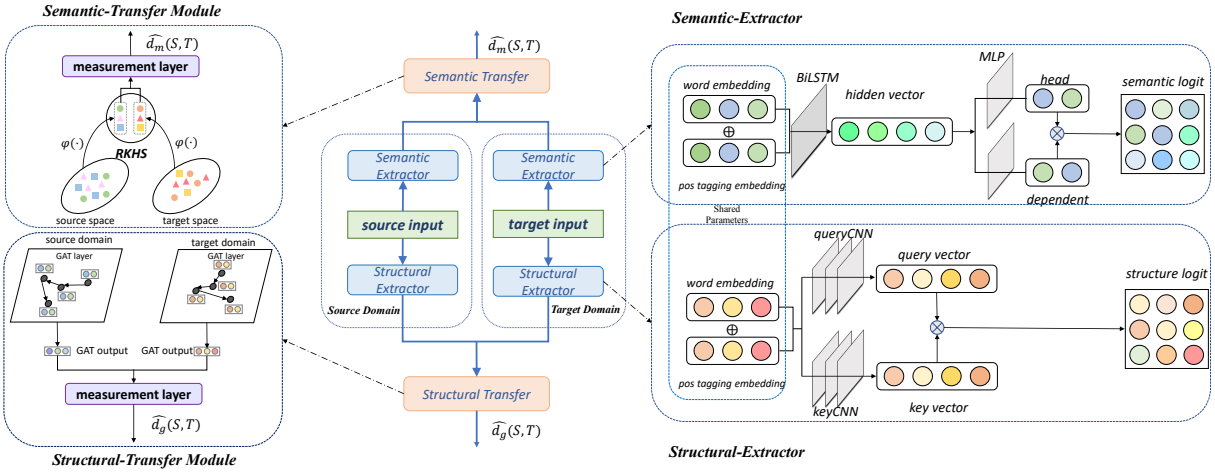

Figure 2: The framework of Semantic-Structural Adaptative Dependency Parsing (SSADP).

3 Methods

The proposed model is mainly composed of four parts: semantic feature extractor, structural feature extractor, semantic feature transfer module and structural feature transfer module. Semantic feature extractor and structural feature extractor are designed to extract domain features. Semantic feature transfer module and structural transfer module are used to align domain feature. The architecture of the proposed SSADP is shown in the Figure 2, and each of the four modules is depicted as follows.

3.1 Semantic Extractor

The semantic feature extractor is the cornerstone of the entire model, which is mainly utilized to achieve dependency parsing task. To keep the conciseness and effectiveness of the semantic feature extractor, a classic graph-based dependency parser Biaffine (Dozat and Manning, 2017) is adopted.

In this module, the word embedding $\boldsymbol{e}_i^w$ and the part-of-speech tagging embedding $\boldsymbol{e}_i^p$ are concatenated as $\boldsymbol{e}_i$ to represent the i-th word in a sentence. A bi-directional LSTM (Huang et al., 2015) is employed to extract hidden semantic feature $\boldsymbol{h}_i = BiLSTM(\mathbf{e})$. Then two Multilayer Perceptrons (MLPs) are used to extract role-specific lower information $r_i^{(head)}$ and $r_i^{(dep)}$.

Finally, a biaffine transformation is used to produce the final affinity score between each pair of words in a sentence:

$$\boldsymbol{v}^{(se)} = [\boldsymbol{r}^{(dep)}; \mathbf{1}]\mathbf{U}[\boldsymbol{r}^{(head)}; \mathbf{1}]^\top \quad (1)$$

where $\boldsymbol{U}$ is the parameter matrix. $\boldsymbol{V}^{(se)}$ as the semantic affinity logit matrix, will be leveraged to compute final affinity logit matrix $\boldsymbol{V}$.

3.2 Structural Extractor

The syntactic dependency tree is highly structured, which inspires us to capture more structural information from parser model. Convolutional Neural Network (CNN) (LeCun et al., 1998) has been proven that retains the capacity of capturing local structure (Niu et al., 2019). Therefore, we adopt a parallel CNN structure query-key CNN (QKCNN) as the structural information extractor (Yang et al., 2018), where QKCNN is composed of query CNN and key CNN. For a sentence, each word embedding $\boldsymbol{e}_i$ is fed into both query and key CNN. Thus, two outputs of query and key CNN are represented as $\boldsymbol{q}_i$ and $\boldsymbol{k}_i$ for the i-th word. Then, the structural affinity score matrix is given as:

$$v_{ij}^{(st)} = \frac{(ReLU(\boldsymbol{k}_i^\top \boldsymbol{q}_j + b))^2}{\sum_{i'}(ReLU(\boldsymbol{k}_{i'}^\top \boldsymbol{q}_j + b))^2} \quad (2)$$

where the bias $\boldsymbol{b}$ is a scalar parameter.

3.3 Semantic Transfer Module

We consider using the method of metric transfer to align the domain features in different spaces. The proposed SSADP uses a method based on maximum mean discrepancy (MMD) to achieve the alignment of domain semantic features. Domain features from different space are mapped into the same Reproducing Kernel Hilbert Space (RKHS) (Schneider et al., 1988) via kernel function.

We define the source domain as $D_s = (x_i^s, y_i^s)_{i=1}^{n_s}$ and target domain as $D_t = (x_i^t)_{i=1}^{n_t}$, where n_s is the number of samples in the source

domain and n_t is the number of samples in the target domain. Assume that D_s and D_t are subject to distribution $\widetilde{p}$ and distribution $\widetilde{q}$. A measurement item $\hat{d}(\widetilde{p}, \widetilde{q})$ is defined to express the distance between $\widetilde{p}$ and $\widetilde{q}$ in the RKHS. For the original MMD measurement, the $\hat{d}$ is represented by the expectation of difference in results of distributions mapped by the kernel function between source and target domain, which is given as:

$$d_H(\widetilde{p}, \widetilde{q}) \triangleq ||E_{\widetilde{p}}[\phi(x^s)] - E_{\widetilde{q}}[\phi(x^t)]||^2_{\mathcal{H}} \quad (3)$$

where $\mathcal{H}$ is a RKHS with kernel functions. $\phi(\cdot)$ is a mapping function which maps the samples from the original space to the RKHS. However, it is hard to explicitly express the mapping function $\phi(\cdot)$ in practice. Instead, some kernel tricks are applied to expand the original MMD to the function calculation formula (Schneider et al., 1988) and thus the computation can be processed easily.

Directly applying the MMD to the dependency parsing task is not such appropriate since rich types of dependencies cannot be fully utilized. Instead of directly using MMD for semantic transfer, we add the inner class iteration to better catch rich types of dependencies information. This variant MMD is called Local-MMD (LMMD) (Zhu et al., 2020), and is formulated as:

$$\hat{d_H}(p, q) = \frac{1}{C}\sum_{c=1}^{C} \| \sum_{\boldsymbol{v}_i^s \in D_s} \omega_i^{sc}\phi(\boldsymbol{v}_i) - \sum_{\boldsymbol{v}_j^t \in D_t} \omega_j^{tc}\phi(\boldsymbol{v}_j) \|^2_{\mathcal{H}} \quad (4)$$

where kernel function $k(x^s, x^t) = \langle \phi(x^s), \phi(x^t) \rangle$, $\langle \cdot, \cdot \rangle$ means inner dot of vectors. C is the number of types of dependency relations and ω^c is the weight of x with the relation c. The w_i^c is calculated as:

$$w_i^c = \frac{r_{ic}}{\sum\limits_{(x_j, r_j) \in D} r_{jc}} \quad (5)$$

where r_{ic} is the c-th entry of one-hot dependency relation vector $\boldsymbol{r}_i$. Considering no labeled data in target domain, the predicted result of parser is used as pseudo label in the unsupervised domain adaptation.

3.4 Structure Transfer Module

For graph-based parsing model, the affinity score matrix can be seen as the adjacency matrix of a special directed weighted graph. Different from aligning the distribution of semantic features in MMD metric, structure transfer module aims to force the structural features to be close via cosine similarity metric to achieve adaptation.

A general metric of measuring the similarity of graph structure is the graph kernel. But the traditional graph kernel methods meet the hard-encode problem. In addition, graph kernel cannot flexibly utilize the node features produced by structural extractor. Thus we cannot adopt the graph kernel metric for structural transfer directly. Instead, we consider employing a graph neural network (GNN) which can be seen as an approximate solution of graph kernel (Kipf and Welling, 2017; Hamilton et al., 2017). Meantime, hard-encode problem can be avoided and node features can be flexibly manipulated by message passing in GNN.

In this paper, we adopt GNN to further encode internal structural information of the dependency graph based on the output of the structural extractor. As mentioned in section 3.2, $\boldsymbol{v}^{(st)}$ contains structural features generated by the QKCNN module. To incorporate this structural features into the word embedding information, we regard the $\boldsymbol{v}^{(st)}$ as the attention coefficient to weight the word embedding as:

$$\boldsymbol{h}_i^0 = \sum_j v_{ij}^{(st)} \boldsymbol{e}_i \quad (6)$$

Then, considering the dependency graph generated in the calculation process is a non-pairwise directed graph, and the Laplacian matrix of directed graphs does not form a unified theory, we introduce Graph Attention Network (GAT) (Veličković et al., 2017) as our GNN encoder in this paper. GAT uses the local neighborhood aggregator to describe the local structure of the dependency graph, and uses the pooling operator to encode the whole picture information. The detailed GAT used for structural transfer is described next.

In GAT, a self-attention mechanism with parameterized weight matrix $\boldsymbol{w}$ is applied to calculate the coefficients:

$$\alpha_{ij} = softmax(a(W\boldsymbol{h}_i, W\boldsymbol{h}_j)) \quad (7)$$

where a is the attention mechanism (Vaswani et al., 2017) and initialized node feature $\boldsymbol{h}_i^0$ is the weighted outputs of structural extractor. The normalized coefficients are used to aggregate the node features by their neighbor nodes as:

$$\boldsymbol{h}_i^{'} = \sigma(\sum_{j \in \mathbb{N}_i} \alpha_{ij} W \boldsymbol{h}_j) \quad (8)$$

Algorithm 1: Model Training

Require : samples $D_s = (x_i^s, y_i^s)_{i=1}^{n_s}$, $D_t = (x_i^t)_{i=1}^{n_t}$
learning rate ℓ
logit coefficient α
loss coefficient λ
Maximum Iterations $MaxIter$
Output : Syntactic Dependency Tree $\mathscr{T}_t$
Forward:
for $iter \leftarrow 1$ **to** $MaxIter$ **do**
  Extract semantic feature $\boldsymbol{r}_s^{(se)}, \boldsymbol{r}_t^{(se)}$
  Extract structural feature $\boldsymbol{r}_s^{(st)}, \boldsymbol{r}_t^{(st)}$
  Extractor produce $\boldsymbol{V}_s^{(se)}, \boldsymbol{V}_s^{(st)}$
  Extractor produce $\boldsymbol{V}_t^{(se)}, \boldsymbol{V}_t^{(st)}$
  $\boldsymbol{V}_s = \alpha \boldsymbol{V}_s^{(se)} + (1-\alpha)\boldsymbol{V}_s^{(st)}$
  $\boldsymbol{V}_t = \alpha \boldsymbol{V}_t^{(se)} + (1-\alpha)\boldsymbol{V}_t^{(st)}$
  $\hat{d}(S,T) = \hat{d}_h(\boldsymbol{v}_s^{(se)}, \boldsymbol{v}_t^{(se)}) + \hat{d}_g(\boldsymbol{r}_s^{(st)}, \boldsymbol{r}_t^{(st)})$
Update:
  $\theta \leftarrow \theta - \ell(J(x^s, y^s) + \lambda \hat{d}(S,T))$
Decode:
  $\mathscr{T}_t = MST(\boldsymbol{V}_t)$

where σ is the readout function. After the aggregation in GAT, a mean pooling is implemented to pool word (node) representations to the sentence (graph) representation. Then we can use cosine similarity to describe the structural similarity between two sentences:

$$sim(S,T) = \langle \boldsymbol{g}_s, \boldsymbol{g}_t \rangle \tag{9}$$

where $\boldsymbol{g}_s$ and $\boldsymbol{g}_t$ are the pooled sentence-level features in source and target domain.

3.5 Overall optimization goal

The task aims to train a model with parameters set θ just using the labeled data of D_s and unlabeled data of D_t for dependency parsing. So the overall objective function can be formulated as:

$$\min_f \frac{1}{n_s} \sum_{i=1}^{n_s} J(f(x_i^s, y_i^s)) + \lambda \hat{d}(\widetilde{p}, \widetilde{q}) \tag{10}$$

where J is Cross-Entropy loss function for dependency parsing task. The $\hat{d}$ is a measurement function that measures the difference between two domains, that can be split as $\hat{d} = \hat{d}_h + \hat{d}_g$, where $\hat{d}_h$ is the LMMD measurement and $\hat{d}_g$ is the graph measurement. We directly use LMMD measurement as loss item $\hat{d}_h$. Particularly, for $\hat{d}_h$, the square value of the difference between the two structure features is used as the loss item given as:

$$\hat{d}_g(\widetilde{p}, \widetilde{q}) = ||\boldsymbol{g}_s - \boldsymbol{g}_t||^2 \tag{11}$$

Table 1: Data statistics in sentence number of CODT1

Domain	Train Set	Dev Set	Test Set	Unlabeled Set
BC	16.3K	1K	2K	–
PB	5.1K	1.3K	2.6K	291K
PC	6.6K	1.3K	2.6K	349K
ZX	1.6K	0.5K	1.1K	33K

Table 2: Data statistics in sentence number of CTB9

Domain	Train Set	Dev Set	Test Set	Unlabeled Set
BS	16K	0.8K	1.9K	–
BN	–	0.6K	0.9K	8.5K
BC	–	0.6K	0.7K	10.7K
WB	–	0.5K	0.6K	9K
DF	–	0.8K	1.8K	17.3K
SC	–	1.5K	3.7K	38.7K
CS	–	0.8K	1.9K	16K

Following the Biaffine (Dozat and Manning, 2017), the classical Maximum Spanning Tree Algorithm (MST) (McDonald et al., 2005) algorithm is adapted to decode a syntactic dependency tree from dependency graph corresponding the affinity logit matrix $V = \alpha \boldsymbol{V}^{(se)} + (1-\alpha)\boldsymbol{V}^{st}$. The whole optimization algorithm is shown in Algorithm 1. And the final loss of SSADP is given as:

$$\mathcal{L} = \frac{1}{n_s} \sum_{i=1}^{n_s} J(f(x_i^s, y_i^s)) + \lambda(\hat{d}_h(\widetilde{p}, \widetilde{q}) + \hat{d}_g(\widetilde{p}, \widetilde{q})) \tag{12}$$

4 Experiments

4.1 Datasets

We conduct experiments on Chinese Open Dependency Treebank 1.0 (CODT1) (Li et al., 2019) and Chinese Tree Bank 9.0 (CTB9) (Xue et al., 2016) datasets.

CODT1 has one source domain: Standard Balanced Corpus (BC) and three target domains with unlabeled data: Taobao Product Blog (PB), Taobao Product Comments (PC) and Network Novel "ZhuXian" (ZX). The statistic of CODT1 is shown in Table 1.

CTB9 (Xue et al., 2016) is a dependency treebank with 8 genres: Newswire (NW), Magazine Articles (MZ), Broadcast News (BN), Broadcast Conversations (BC), Webblogs (WB), Discussion Forums (DF), SMS/Chat Messages (SC) and Conversational Speech (CS). We notice that genres of some classical in-domain treebanks such as CTB5 (Zhang and Clark, 2008) and HIT-CDT (Li et al., 2012) is mostly news and magazines. So we use the

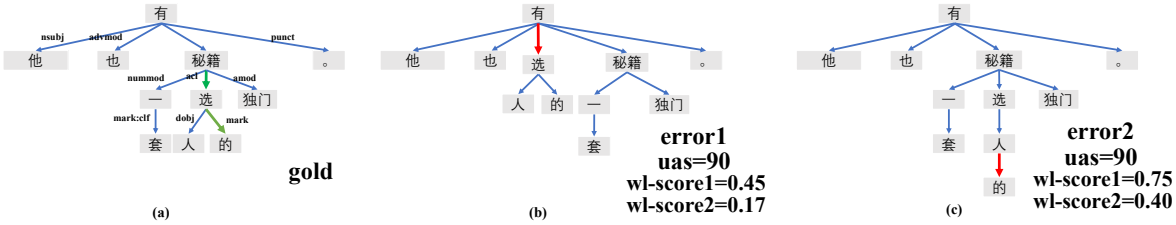

Figure 3: Dependency Syntactic Tree Example Diagram. The original sentence is "他/也/有/一/套/选人/的/独门/秘籍。(He also has a set of unique cheats for choosing people.)" which is in WB domain of CTB9.

Table 3: Train, Dev and Test data from CTB9

Domain	Train Set	Dev Set	Test Set	Unlabeled Set
BS	[001–815; 1001-1136]	[900-931; 1148-1151]	[816-885; 1137-1147]	–
BN	–	[4051-4111]	[3041-3145]	[2000-3040]
BC	–	[4193-4194]	[4195-4197]	[4112-4192]
WB	–	[4337-4345]	[4346-4411]	[4198-4336]
DF	–	[5481-5500]	[5501-5558]	[5000-5480]
SC	–	[6606-6638]	[6639-6700]	[6000-6605]
CS	–	[7015-7015]	[7016-7017]	[7000-7014]

Table 4: Hyperparameters

Module	Parameter	Value
QKCNN	hidden layers	3
	in channels	300
	out channels	128
	dropout	0.3
GAT	hidden layer	2
	hidden dim	128
	attention heads	4
	dropout	0.6

data in the NW and MZ domains as the source domain called Basic Source (BS). Follow the data segmentation rules of CTB5 (Zhang and Clark, 2008), we split the original data of source domain and target domain. Specific segmentation details of the data are shown in Table 2 and Table 3. For the unlabeled set of target domain, we discard its original annotation label to simulate the situation encountered in actual applications where only unlabeled data is available on the target domain.

4.2 Experimental Settings

Considering that there exists few work on improving the model structure of unsupervised CDP, we set two feature-based models as baselines in our experiments: 1) Follow the baseline setting of Li et al. (2019), we reproduce Biaffine (Dozat and Manning, 2017) as a baseline. 2) Follow the idea of adversarial domain adaptation, we proposed BiaffineAdv within the Biaffine as the generator and the TextCNN (Kim, 2014) as the domain discriminator (Sato et al., 2017).

Hyperparameters We use a 300-dimensional embedding layer, and other settings of semantic extractor are consistent with original Biaffine (Dozat and Manning, 2017). During training, we utilized Adam optimizer (Kingma and Ba, 2015) with a 0.001 learning rate. The coefficient λ and α are both 0.6. Other important hyperparameters are shown in the Table 4.

4.3 Evaluation Metrics

We use Unlabeled Accuracy Score (UAS) and Labeled Accuracy Score (LAS) as evaluation metrics.

Moreover, to explore whether our model can improve the transfer ability of structural information, we introduce Weisfeiler-Lehman Test (WL-Test), a test used to judge whether two graphs are isomorphic in graph theory. WL-Test produces a unique feature label set $\Gamma_{\mathbb{G}}$ for input graph $\mathbb{G}$ and gives a boolean result of the isomorphism finally. In order to quantitatively compare the transfer ability of structural information with different models, we improve the Isomorphic Score S of graph $\mathbb{G}$ and graph $\mathbb{H}$ to a real value between [0, 1] via Jaccard Distance:

$$\begin{aligned} S(\mathbb{G},\mathbb{H}) &= Jaccard(\mathbb{G},\mathbb{H}) \\ &= \frac{|\Gamma_{\mathbb{G}} \cap \Gamma_{\mathbb{H}}|}{|\Gamma_{\mathbb{G}} \cup \Gamma_{\mathbb{H}}|} \end{aligned} \tag{13}$$

where the $S(\mathbb{G},\mathbb{H}) \in [0,1]$. The higher as $S(\mathbb{G},\mathbb{H})$, the higher the isomorphism similarity of the two graphs $\mathbb{G}$ and $\mathbb{H}$. When the $S(\mathbb{G},\mathbb{H}) = 1$, the two graphs are completely isomorphic.

Different from UAS, Isomorphic Score can reflect the deep structural information of the syntactic tree as shown in Figure 3: UAS of Figure 3(b) and Figure 3(c) are both 90, the Isomorphic Score S of Figure 3(c) is higher than Figure 3(b). We can also intuitively see that, taking Figure 3(a) as a bench-

Table 5: Results on test data of CODT1

Model	BC→ PB		BC→ ZX		BC→ PC		Average Gain	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Biaffine (Li et al., 2019)	67.55	61.01	68.44	59.55	–	–	–	–
Biaffine (Ours)	67.75	60.95	69.41	61.55	39.95	26.96	–	–
BiaffineAdv	67.74	60.91	69.49	61.73	41.01	27.30	0.38	0.16
SSADP	**68.55**	**61.59**	**70.82**	**63.61**	**41.10**	**27.67**	1.12	1.14

Table 6: Results on test data of CTB9

Model	BS→ BC		BS→ BN		BS→ DF		BS→ WB		BS→ SC		BS→ CS		Average Gain	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Biaffine (Ours)	79.06	74.15	91.91	89.38	88.18	85.21	87.85	84.31	84.88	80.86	71.03	63.48	–	–
BiaffineAdv	78.81	73.54	91.57	88.96	88.14	85.31	87.32	83.67	84.85	81.20	71.52	63.75	-0.12	-0.16
SSADP	**79.60**	**74.68**	**92.19**	**89.70**	**88.58**	**85.82**	**88.16**	**84.57**	**85.46**	**81.65**	**71.57**	**64.45**	0.44	0.59

mark, Figure 3(b) has more changes in structure than Figure 3(c).

4.4 Results

We compare the performance of the proposed SSADP with mentioned baselines on CODT1 shown in Table 5 and CTB9 shown in Table 6. We have three observations described as follows: 1) Compared with Biaffine (Li et al., 2019), Biaffine (Ours) outperforms the former on three domains of CODT1. And on this basis, compared with the Biaffine (Ours), the proposed SSADP obtains significant performance gain on both CODT1 and CTB9 as shown in the last column of each tabale. These significant improvements demonstrate the effectiveness of SSADP; 2) The performance gain in CTB9 is lower than the performance gain obtained in CODT1, but still significant. It is noteworthy that, despite the amount of unlabeled data in CTB9 being far less than CODT1, where CODT: CTB9 is about 14:1, SSADP still obtains improvements on all domains in CTB9. This shows that our model has good adaptation ability for CDP; 3) Our SSADP is very stable even in the case of lack of unlabeled data. BiaffineAdv can achieve some performance improvement in CODT1, but BiaffineAdv produces negative transfer in almost all domains on CTB9. The magnitude of unlabeled data on BN, BC and WB is less than BS shown in Table 2. And we can observe obvious negative transfer in these domains between BiaffineAdv and Biaffine (Ours) baseline, which further indicates that unlabeled data is critical to other unsupervised domain adaptation approaches, where the proposed SSADP retains the effective transfer ability even in scenarios where unlabeled data is scarce.

4.5 Weisfeiler-Lehman Test

In our WL-Test, we treat each predicted syntactic dependency tree as a predicted graph and each gold syntactic dependency tree as a gold graph.

As shown in Figure 4, the proposed SSADP achieves the highest Isomorphic Score S on both PB and ZX domains, while on PC domain, there is a trivial difference between SSADP and Biaffine (Ours). Comparing the Isomorphic Score S under two ablation settings, we can also see that although the UAS and LAS of SSADP w/o LMMD and SSADP w/o GAT are similar, the Isomorphic Scores S of SSADP w/o LMMD are all higher than SSADP w/o GAT, which further proves that structural transfer indeed transfer more structural information than semantic transfer. Structure transfer can better transfer text structural information in the domain. We can see the Isomorphic Score S of BiaffineAdv is similar with Biaffine (Ours) in three domains. In conclusion, we can infer that the adversarial domain adaptation can hardly learn common structural information between domains.

4.6 Ablation Study

In this section, the impact of the two transfer modules is revealed. We remove the structural transfer and semantic transfer modules from the proposed SSADP respectively, and then perform training and test on the three domains on CODT1.

The results of ablation study are shown in Table 7. Two independent modules, SSADP w/o GAT (retains semantic transfer) and SSADP w/o LMMD (retains structural transfer), are tested. The results show that both modules take effect on three domains compared with the Biaffine (Ours). The detail performance improvement is shown in the last

Table 7: Ablation result of test data on CODT1

Model	BC→ PB		BC→ ZX		BC→ PC		Average Gain	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Biaffine (Li et al., 2019)	67.55	61.01	68.44	59.55	–	–	–	–
Biaffine (Ours)	67.75	60.95	69.41	61.55	39.95	26.96	–	–
SSADP w/o GAT	68.40	61.32	70.20	62.39	40.12	26.49	0.54	0.25
SSADP w/o LMMD	68.30	61.30	70.28	62.90	40.34	26.84	0.60	0.53
SSADP	**68.55**	**61.59**	**70.82**	**63.61**	**41.10**	**27.67**	1.12	1.14

Table 8: Result of SSADP with pseudo annotation

Model	BC→ PB		BC→ ZX		Avg Gain	
	UAS	LAS	UAS	LAS		
Biaffine (Li et al., 2019)	67.55	61.01	68.44	59.55	–	–
Biaffine (Ours)	67.75	60.95	69.41	61.55	–	–
SyntaxError (2019)	71.48	65.43	73.90	66.54	4.11	4.74
SSADP	68.55	61.59	70.82	63.61	1.11	1.35
SSADP with annotation	**71.68**	**65.85**	**74.93**	**67.64**	4.73	5.50

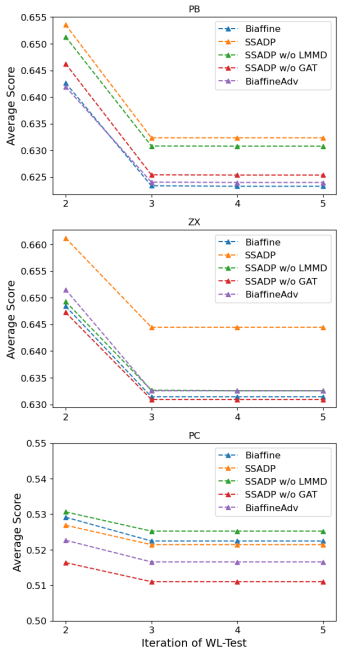

Figure 4: Weisferiler-Lehman Test Curve of CODT1.

column of Table 7. The performance improvement of each transfer method is still signigicant, which demonstrates each transfer module of SSADP is effective. The proposed SSADP achieves higher performance of 0.28 and 0.55 on UAS and LAS compared with semantic transfer and 0.52 and 0.61 compared with structure transfer. It indicates that the proposed SSADP enhances the transfer ability via complementarily achieving the alignment of semantic and structural information simultaneously. More exploration results can be found in the Appendix.

4.7 SSADP with Pseudo-label Annotation

The proposed SSADP focuses on the design of model structure, which is complementary with pseudo-label annotation method in theory. Thus we extend the proposed SSADP with pseudo-label annotation strategy to get a better performance gain.

The SSADP with pseudo-label annotation strategy is conducted as follows:

1) Using the trained baseline and SSADP to label the unlabeled data of target domain.

2) Filtering the predicted sentence which get the same predictions of two models.

3) Combining the remaining sentences of the target domain into the original training set, and retrain the model.

4) Repeat the step 1-3 until the performance of retrained model is stable.

We conduct experiment on the PB and ZX domain of CODT1 and the results are shown in Table 8. The proposed SSADP outperforms the SyntaxError. SyntaxError is the winner of NLPCC2019 Shared Task1-subtask1, which use character-level feature to enhance the ability of model and integrate adversarial training, self-training and tri-training to achieve CDP, the multi-model vote strategy is also be used for final predict (Peng et al., 2019). This result indicates that SSADP can combine with pseudo-annotation methods.

5 Conclusion

We propose a novel parsing model termed SSADP that can achieve unsupervised cross-domain dependency parsing by extracting semantic and structural features and performing domain alignment without using pseudo annotation and data selection. The experimental results on the CODT1 and CTB9 datasets demonstrate the effectiveness of our model. Moreover, we adopted Weisferiler-Lehman Test to verify the structural transfer ability of the proposed SSADP and other baselines. Finally, by extending SSADP with pseudo-annotation method, we show that proposed SSADP can be combined with the

previous pseudo-annotation cross-domain methods and achieve better performance.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (61702047).

References

Karsten M. Borgwardt, Arthur Gretton, Malte J. Rasch, Hans Peter Kriegel, Bernhard Schölkopf, and Alex J. Smola. 2006. Integrating structured biological data by Kernel Maximum Mean Discrepancy. *Bioinformatics*, 22(14):1–10.

Raphael Cohen, Yoav Goldberg, and Michael Elhadad. 2012. Domain Adaptation of a Dependency Parser with a Class-Class Selectional Preference Model. *Proceedings of ACL 2012 Student Research Workshop (ACL '12)*, (July):43–48.

Timothy Dozat and Christopher D. Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. *In Proceedings of ICLR*.

Mark Dredze, John Blitzer, Partha Talukdar, Kuzman Ganchev, Joao Graca, and Fernando Pereira. 2007. Frustratingly hard domain adaptation for dependency parsing. pages 1051–1055.

Muhammad Ghifary, W. Bastiaan Kleijn, and Mengjie Zhang. 2014. Domain adaptive neural networks for object recognition. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8862:898–904.

William L. Hamilton, Rex Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):1025–1035.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging.

Zhanming Jie, Aldrian Obaja Muis, and Wei Lu. 2017. Efficient dependency-guided named entity recognition. *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, pages 3457–3465.

Mohammad Khan, Markus Dickinson, and Sandra Kübler. 2013. Towards domain adaptation for parsing web data. *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 357–364.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar. Association for Computational Linguistics.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A Method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, pages 1–15.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pages 1–14.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323.

Zhenghua Li, Ting Liu, and Wanxiang Che. 2012. Exploiting multiple treebanks for parsing with quasi-synchronous grammars. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 675–684, Jeju Island, Korea. Association for Computational Linguistics.

Zhenghua Li, Xue Peng, Min Zhang, Rui Wang, and Luo Si. 2019. Semi-supervised domain adaptation for dependency parsing. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pages 2386–2395.

Jostein Lien, Erik Velldal, and Lilja Øvrelid. 2015. Improving cross-domain dependency parsing with dependency-derived clusters. *Proceedings of the 20th Nordic Conference of Computational Linguistics (NODALIDA 2015)*, (Nodalida):117–126.

Xuezhe Ma, Zecong Hu, Jingzhou Liu, Nanyun Peng, Graham Neubig, and Eduard Hovy. 2018. Stack-pointer networks for dependency parsing. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:1403–1414.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajĭc. 2005. Non-projective dependency parsing using spanning tree algorithms. *HLT/EMNLP 2005 - Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, (October):523–530.

Guocheng Niu, Hengru Xu, Bolei He, Xinyan Xiao, Hua Wu, and Sheng Gao. 2019. Enhancing local feature extraction with global representation for neural text classification. *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, pages 496–506.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The conll 2007 shared task on dependency parsing. *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language*

Processing and Computational Natural Language Learning (EMNLP-CoNLL), pages 915–932.

Xue Peng, Zhenghua Li, Min Zhang, Rui Wang, Yue Zhang, and Luo Si. 2019. *Overview of the NLPCC 2019 Shared Task: Cross-Domain Dependency Parsing*, volume 11839 LNAI. Springer International Publishing.

Barbara Plank and Gertjan Van Noord. 2011. Effective measures of domain similarity for parsing. *ACL-HLT 2011 - Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 1:1566–1576.

Motoki Sato, Hitoshi Manabe, Hiroshi Noji, and Yuji Matsumoto. 2017. Adversarial training for cross-domain universal dependency parsing. In *CoNLL 2017 - SIGNLL Conference on Computational Natural Language Learning, Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 71–79.

P.-A. Schneider, H Bounameaux, J N Cox, Schneider P.-A., Bounameaux H., and Cox J.N. 1988. Endarterectomy by Simpson catheter in arterial stenosis of the lower limbs . *Schweizerische Medizinische Wochenschrift*, 118(52):1997–2000.

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. 2014. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*.

Tanmay Vakare, Kshitij Verma, and Vedant Jain. 2019. Sentence Semantic Similarity Using Dependency Parsing. *2019 10th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2019*, pages 2019–2022.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, pages 5998–6008.

Petar Velicković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2017. Graph attention networks. *arXiv*, pages 1–12.

Nianwen Xue, Xiuhong Zhang, Zixin Jiang, Martha Palmer, Fei Xia, Fu-Dong Chiou, and Meiyu Chang. 2016. Chinese treebank 9.0 ldc2016t13. web download. philadelphia: Linguistic data consortium.

Hongliang Yan, Yukang Ding, Peihua Li, Qilong Wang, Yong Xu, and Wangmeng Zuo. 2017. Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:945–954.

Zhilin Yang, Jake Zhao, Bhuwan Dhingra, Kaiming He, William W. Cohen, Ruslan Salakhutdinov, and Yann LeCun. 2018. GLoMo: Unsupervisedly learned relational graphs as transferable representations. *arXiv*, pages 1–12.

Juntao Yu, Mohab El-karef, and Bernd Bohnet. 2015. Domain adaptation for dependency parsing via self-training. *Proceedings of the 14th International Conference on Parsing Technologies*, pages 1–10.

Yue Zhang and Stephen Clark. 2008. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. pages 562–571.

Yongchun Zhu, Fuzhen Zhuang, Jindong Wang, Guolin Ke, Jingwu Chen, Jiang Bian, Hui Xiong, and Qing He. 2020. Deep Subdomain Adaptation Network for Image Classification. *IEEE Transactions on Neural Networks and Learning Systems*, 1:1–10.

Appendix

Table 9: Result of MMD Test on CODT1

Model	BC→ PB		BC→ ZX		BC→ PC	
	UAS	LAS	UAS	LAS	UAS	LAS
Biaffine (2019)	67.55	61.01	68.44	59.55	–	–
Biaffine (Ours)	67.75	60.95	69.41	61.55	39.95	26.96
Biaffine-QKCNN-MMD	68.20	61.34	69.34	61.96	39.79	26.43
SSADP (w/o GAT)	**68.40**	**61.32**	**70.20**	**62.39**	**40.12**	**26.49**

As mentioned in Section 3.3 of the text, Maximum Mean Discrepancy (MMD) is an efficient but rough domain adaptation method. We use Local Maximum Measure Discrepancy (LMMD) (Zhu et al., 2020) in the proposed Semantic-Structural Adaptative Dependency Parser (SSADP) instead of MMD. In order to verify that LMMD can make better use of dependency labels mentioned above, we try to conduct replacement experiments on CODT1 (Li et al., 2019) to compare the effects of MMD and LMMD. In order to eliminate the influence of structural transfer module, we just use three modules of the proposed SSADP: semantic extractor (Biafine (Dozat and Manning, 2017)) + structural extractor (QKCNN (Yang et al., 2018)) + semantic transfer module (MMD or LMMD).

From the results shown in Table 9, we can see MMD method is effective in PB domain and ZX domain, but it shows negative transfer in PC domain. The results demonstrate LMMD outperforms MMD in unsupervised cross-domain dependency parsing, which is consistent with the previous theoretical analysis.