Semi-supervised Domain Adaptation for Dependency Parsing via Improved Contextualized Word Representations

Ying Li[1], **Zhenghua Li**[1*] and **Min Zhang**[1]
[1]Institute of Artificial Intelligence, School of Computer Science and Technology,
Soochow University, China
yingli_hlt@foxmail.com, {zhli13, minzhang}@suda.edu.cn

Abstract

In recent years, parsing performance is dramatically improved on in-domain texts thanks to the rapid progress of deep neural network models. The major challenge for current parsing research is to improve parsing performance on out-of-domain texts that are very different from the in-domain training data when there is only a small-scale out-domain labeled data. To deal with this problem, we propose to improve the contextualized word representations via adversarial learning and fine-tuning BERT processes. Concretely, we apply adversarial learning to three representative semi-supervised domain adaption methods, i.e., direct concatenation (CON), feature augmentation (FA), and domain embedding (DE) with two useful strategies, i.e., fused target-domain word representations and orthogonality constraints, thus enabling to model more pure yet effective domain-specific and domain-invariant representations. Simultaneously, we utilize a large-scale target-domain unlabeled data to fine-tune BERT with only the language model loss, thus obtaining reliable contextualized word representations that benefit for the cross-domain dependency parsing. Experiments on a benchmark dataset show that our proposed adversarial approaches achieve consistent improvements, and fine-tuning BERT further boosts the parsing accuracy by a large margin. Our single model achieves the same state-of-the-art performance as the top submitted system in the NLPCC-2019 shared task, which uses ensemble models and BERT.

1 Introduction

Dependency parsing aims to capture syntax with a dependency tree and is proven to be helpful for various natural language processing (NLP) tasks, such as semantic role labeling (Xia et al., 2019), natural language generation (Park and Kang, 2019), and machine translation (Hadiwinoto and Ng, 2017). Given an input sentence $\mathbf{s} = w_1 w_2 \dots w_n$, a dependency tree, as depicted in Figure 1, is defined as $\mathbf{d} = \{(h, m, l), 0 \leq h \leq n, 1 \leq m \leq n, l \in \mathcal{L}\}$, where (h, m, l) is a dependency from the head word w_h to the child word w_m with the relation label $l \in \mathcal{L}$, and w_0 is a pseudo word that points to the root word of the sentence.

In recent years, neural network based approaches have achieved remarkable improvement and outperformed the traditional discrete-feature based approaches by a large margin in dependency parsing (Chen and Manning, 2014; Kiperwasser and Goldberg, 2016; Andor et al., 2016; Dozat and Manning, 2017). Most remarkably, Dozat and Manning (2017) propose a simple yet effective deep BiAffine parser and achieve the state-of-the-art accuracy on a variety of datasets and languages.

However, the domain adaptation problem, i.e., how to improve parsing performance on texts that are very different from the training data, remains a key challenge for the parsing community, especially when trying to apply the parsing technique to real-life web data. Taking the examples in Figure 1, we can see that as user-generated texts, the left sentence from the product comment (PC) domain is quite non-canonical and contains a lot of ellipsis phenomena. In contrast, the right one from the balanced corpus (BC) domain is a typical sentence from newswire texts and is much more formal. Hence, domain

*Corresponding author.
This work is licensed under a Creative Commons Attribution 4.0 International License. License details: http://creativecommons.org/licenses/by/4.0/.

Proceedings of the 28th International Conference on Computational Linguistics, pages 3806–3817
Barcelona, Spain (Online), December 8-13, 2020

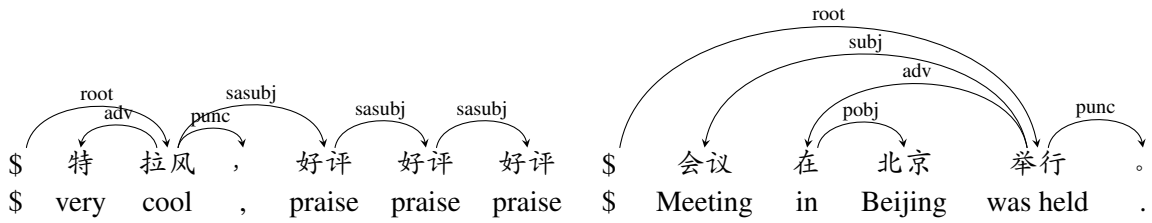

Figure 1: Examples of dependency trees. The left sentence is from the target-domain PC data and the right one is from the source-domain BC data.

differences can be represented with both sentence and parse tree distribution changes due to new words and phrases, new expression structures, etc. The key for domain adaptation is how to model differences and commonalities between different domains.

Most previous works focus on *unsupervised* cross-domain parsing, assuming there is no target-domain labeled data. Typical methods include self-training (McClosky and Charniak, 2008; Yu et al., 2015) and co-training (Sarkar, 2001). However, due to the intrinsic difficulty of domain adaptation, progress in this direction is very slow. In the past few years, *semi-supervised* cross-domain parsing attracts more attention due to the emergence of more labeled data. Particularly, Li et al. (2019b) release large-scale labeled and unlabeled datasets, and they find that their proposed domain embedding (DE) approach is more effective than the direct concatenation (CON) method. The feature augmentation (FA) method, as another typical technique for semi-supervised domain adaptation, is first proposed by Daumé III (2007). Kim et al. (2016) successfully apply it to a neural model which leverages multiple BiLSTMs to extract shared and private domain features. To learn the differences and commonalities between source and target domains, the DE method uses explicit domain indicators as extra inputs, whereas the FA method employs a shared and two private BiLSTM encoders for the feature separation.

This work proposes to improve the contextualized word representation by adversarial learning and fine-tuning BERT, thus further modeling more pure yet effective domain-specific and domain-invariant representations. To alleviate the domain-invariant representations from being contaminated by domain-specific ones, we apply adversarial learning to enhance three typical semi-supervised approaches, i.e., CON, FA, and DE with two useful strategies, i.e., fused target-domain word representations and orthogonality constraints. At the same time, we utilize a large-scale target-domain unlabeled data to fine-tune BERT and obtain more reliable contextualized word representations, leading to a large improvement over using off-the-shelf BERT representations. Our final single model achieves nearly the same state-of-the-art performance as the ensemble models with BERT of Li et al. (2019c), which won the first place in the cross-domain parsing shared task recently organized at the international conference on natural language processing and Chinese computing (NLPCC-2019). Although we focus on semi-supervised domain adaptation for dependency parsing, the techniques and findings may be applicable to domain adaptation for other NLP tasks. All codes are released publicly available for the research purpose [1].

2 Base Model

In this work, we select the state-of-the-art BiAffine parser as our strong baseline model. As shown in the left part of Figure 2, the parser mainly contains four components: *Input layer*, *BiLSTM encoder*, *MLP layer*, and *BiAffine layer*.

Input layer. Given an input sentence $\mathbf{s} = w_0 w_1 \dots w_n$, the input layer directly maps it into vector representations $\mathbf{x}_0 \mathbf{x}_1 \dots \mathbf{x}_n$. Each vector representation $\mathbf{x}_i$ is the concatenation of its word and POS-tag embeddings:

$$\mathbf{x}_i = \mathbf{emb}_{w_i}^{\texttt{word}} \oplus \mathbf{emb}_{t_i}^{\texttt{tag}} \quad (1)$$

where $\mathbf{emb}_{w_i}^{\texttt{word}}$ is the sum of a fixed word2vec representation and a fine-tuned word embedding.

[1] https://github.com/suda-yingli/COLING2020-adv

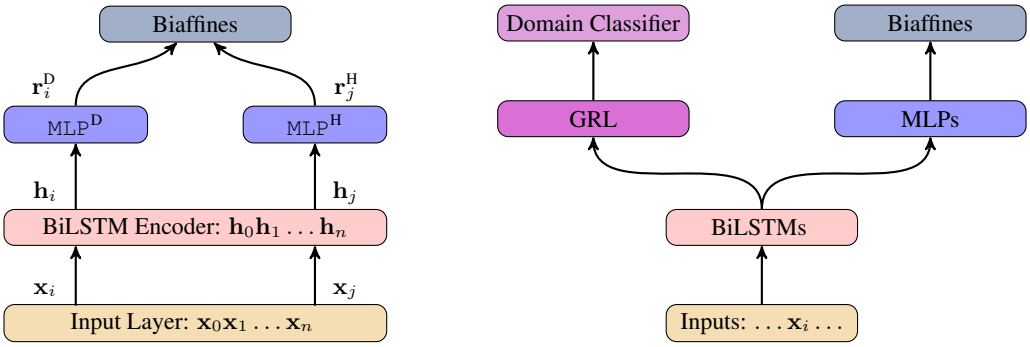

Figure 2: The left part is the framework of BiAffine parser, and the right is the framework of adversarial CON model.

$\mathbf{emb}_{t_i}^{\mathtt{tag}}$ is a fine-tuned POS-tag embedding. Additionally, we also enhance model performance by replacing the word embedding $\mathbf{emb}_{w_i}^{\mathtt{word}}$ with its BERT representation $\mathbf{rep}_{w_i}^{\mathtt{BERT}}$.

BiLSTM encoder. The BiLSTM encoder takes $\mathbf{x}_0\mathbf{x}_1 ... \mathbf{x}_n$ as inputs and obtains context-aware word representations $\mathbf{h}_0\mathbf{h}_1 ... \mathbf{h}_n$. First, a three-layer BiLSTM is applied to sequentially encode the input words from forward and backward two directions. Then, the two sequences of hidden states are obtained, represented as $\overrightarrow{\mathbf{h}}_0 \overrightarrow{\mathbf{h}}_1 ... \overrightarrow{\mathbf{h}}_n$ and $\overleftarrow{\mathbf{h}}_0 \overleftarrow{\mathbf{h}}_1 ... \overleftarrow{\mathbf{h}}_n$. Finally, we concatenate $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$ at each step as the final hidden states $\mathbf{h}_i$. We omit the detailed computation of BiLSTM encoder and write it as follows:

$$\mathbf{h}_0\mathbf{h}_1 \ldots \mathbf{h}_n = \mathrm{BiLSTM}(\mathbf{x}_0\mathbf{x}_1 \ldots \mathbf{x}_n, \theta_{\mathrm{BiLSTM}}) \tag{2}$$

where the θ_{BiLSTM} represents all the parameters of the BiLSTM encoder.

MLP (multi-layer perceptron) layer. The MLP layer takes $\mathbf{h}_i$ as input and uses two separate MLPs to get two lower-dimensional representation vectors.

$$\begin{aligned} \mathbf{r}_i^{\mathrm{H}} &= \mathrm{MLP}^{\mathrm{H}}(\mathbf{h}_i) \\ \mathbf{r}_i^{\mathrm{D}} &= \mathrm{MLP}^{\mathrm{D}}(\mathbf{h}_i) \end{aligned} \tag{3}$$

where $\mathbf{r}_i^{\mathrm{H}}$ is the representation vector of w_i as a head word, and $\mathbf{r}_i^{\mathrm{D}}$ as a dependent, and $\mathrm{MLP}^{\mathrm{H/D}}$ both have a single hidden layer with the ReLU activation function.

BiAffine layer. The scores of all dependencies are computed via a BiAffine operation,

$$\mathtt{score}(i \leftarrow j) = \begin{bmatrix} \mathbf{r}_i^{\mathrm{D}} \\ 1 \end{bmatrix}^{\mathrm{T}} \mathbf{W}^b \mathbf{r}_j^{\mathrm{H}} \tag{4}$$

where $\mathtt{score}(i \leftarrow j)$ is the score of the dependency (j, i) and the matrix $\mathbf{W}^b$ is a BiAffine parameter. The arc-factorization score of a dependency tree is computed with extra MLPs, which can be seen in Dozat and Manning (2017). After obtaining the scores, the highest-scoring tree can be decoded with the dynamic programming algorithm known as maximum spanning tree (McDonald et al., 2005).

Parser loss. Assuming w_j is the gold-standard head of w_i, the BiAffine parser loss for each position i is

$$L_{parser} = -\log \frac{e^{\mathtt{score}(i \leftarrow j)}}{\sum_{0 \le k \le n, k \ne i} e^{\mathtt{score}(i \leftarrow k)}} \tag{5}$$

The BiAffine parser treats the classification of dependency labels as a separate task after finding the highest-scoring dependency tree.

3 Approaches

In this work, we propose to improve contextualized word representations by adversarial learning and fine-tuning BERT processes to boost the performance of cross-domain dependency parsing. Concretely, we apply adversarial learning to three typical semi-supervised approaches with two useful strategies, thus obtaining more pure word representations. Simultaneously, we propose to fine-tune BERT with all target-domain unlabeled data to obtain more reliable word representations.

3.1 The Adversarial CON Method

The CON method is the most common technique for semi-supervised cross-domain dependency parsing, which ignores domain differences and directly trains the BiAffine parser with all source- and target-domain labeled data. To capture the domain-invariant information that is not special to a particular domain as much as possible, we employ an adversarial network on BiAffine parser, which is shown in the right of Figure 2.

Following Ganin and Lempitsky (2015), we use a Gradient Reversal Layer (GRL) for adversarial learning to prevent the domain classifier from making an accurate prediction about the domain types of the word. First, the inputs from different domains are parameterized by the same BiLSTM, and its output $\mathbf{h}_i$ is used for adversarial learning and dependency parsing. For adversarial learning, the GRL takes $\mathbf{h}_i$ as its input, and the forward and backward propagations of the GRL are defined as follows:

$$\begin{aligned} \mathrm{GRL}_\lambda(\mathbf{h}_i) &= \mathbf{h}_i \\ \frac{\mathrm{dGRL}_\lambda(\mathbf{h}_i)}{\mathrm{d}(\mathbf{h}_i)} &= -\lambda\mathbf{I} \end{aligned} \tag{6}$$

where λ is a hyper-parameter. Over the GRL, the domain classifier utilizes an MLP to compute the domain scores and a softmax to obtain the probabilities of domain distribution for each word w_i,

$$\mathbf{z}_i = \mathrm{softmax}\left(\mathbf{W}_2\mathrm{ReLU}(\mathbf{W}_1\mathbf{h}_i + b_1) + b_2\right) \tag{7}$$

where $\theta_d = \{\mathbf{W}_1, \mathbf{W}_2, b_1, b_2\}$ denotes the parameters of domain classifier. The adversarial network is trained to minimise the cross-entropy of the predicted and true distributions,

$$L_{adv} = \sum_{i=0}^{n}\sum_{j=1}^{m} \hat{z}_i \log\left(z_i^j\right) \tag{8}$$

where $\hat{z}_i$ is the gold domain of word w_i, z_i^j represents the predicted probability of word w_i belonging to domain j, n is the word number of one sentence, and m is the domain number. Finally, the adversarial CON model is jointly trained with parser and adversary losses, where α is a hyper-parameter to balance the parsing and adversarial learning tasks.

$$L_{con}^{*} = L_{parser} + \alpha L_{adv} \tag{9}$$

3.2 The Adversarial FA Method

The FA method is another popular technique for domain adaptation, which applies a shared and m private BiLSTMs to learn domain-invariant and domain-specific features (Kim et al., 2016). To alleviate the shared and private latent feature spaces from interfering with each other, we apply the adversarial learning to the FA model with two useful strategies, i.e., fused target-domain word representations and orthogonality constraints.

As shown in the left of Figure 3, we employ a shared and two private BiLSTM encoders for feature separation. First, the input $\mathbf{x}_i$ is fed into a shared BiLSTM and its corresponding private BiLSTM, thus obtaining domain-invariant representation $\mathbf{h}_i^{\mathtt{inv}}$ and domain-specific one $\mathbf{h}_i^{\mathtt{spe}}$. Then, we employ two

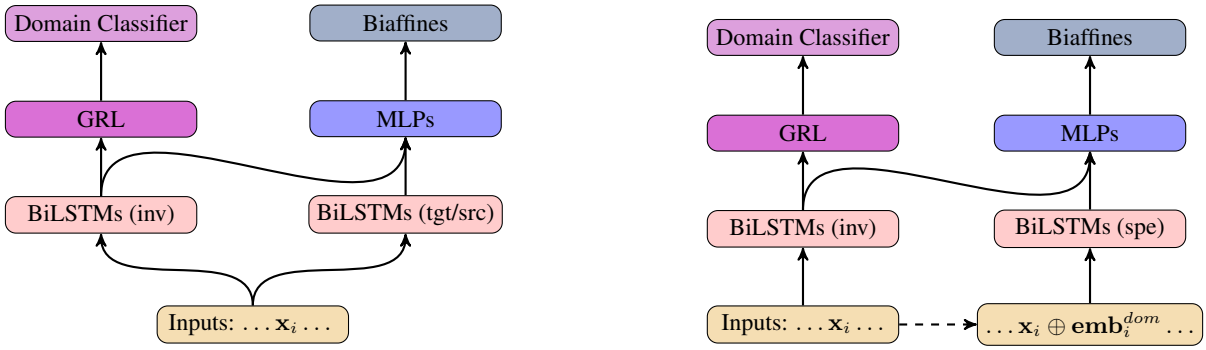

Figure 3: The left is the framework of adversarial FA model, and the right is the adversarial DE model.

useful strategies to prevent the domain-invariant representations from being contaminated by domain-specific features.

Fused target-domain word representations. Due to the lack of target-domain labeled data, the parameters of the target-domain private BiLSTM encoder may not be fully optimized. Hence, we employ the fused BiLSTM outputs as the final domain-specific representations $\mathbf{h}_i^{\texttt{spe}}$ when the input word is from the target domain. Otherwise, we keep the raw private BiLSTM outputs as $\mathbf{h}_i^{\texttt{spe}}$.

$$\mathbf{h}_i^{\texttt{spe}} = \begin{cases} \gamma \mathbf{h}_i^{\texttt{src}} + (1-\gamma)\mathbf{h}_i^{\texttt{tgt}}, & \text{if } w_i \in \{target\ domain\} \\ \mathbf{h}_i^{\texttt{src}}, & \text{if } w_i \in \{source\ domain\} \end{cases} \tag{10}$$

where $\mathbf{h}_i^{\texttt{src}}$ and $\mathbf{h}_i^{\texttt{tgt}}$ are the outputs of source- and target-domain private BiLSTMs.

Orthogonality constraints. Following Bousmalis et al. (2016), we encourage the domain-specific features to be mutually exclusive with the shared features by imposing the orthogonality constraints. The loss of orthogonality constraints is computed as follows:

$$L_{ort} = \sum_{i=0}^{n} \left\| (\mathbf{h}_i^{\texttt{inv}})^{\mathrm{T}} \mathbf{h}_i^{\texttt{spe}} \right\| \tag{11}$$

We then use the combination of $\mathbf{h}_i^{\texttt{inv}}$ and $\mathbf{h}_i^{\texttt{spe}}$ as final contextualized word representations $\mathbf{h}_i^{'}$ for the dependency parsing, while $\mathbf{h}_i^{\texttt{inv}}$ is used for adversarial learning to make the shared space more pure. Finally, our adversarial FA model is jointly trained with the total loss L_{fa}^*, which is defined as follows:

$$L_{fa}^* = L_{parser} + \alpha L_{adv} + \beta L_{ort} \tag{12}$$

where α and β are hyper-parameters.

3.3 The Adversarial DE Method

The DE method is recently proposed by Li et al. (2019b), which trains the BiAffine parser by concatenating the primary input vector $\mathbf{x}_i$ and a fine-tuned domain embedding $\mathbf{emb}_{d_i}^{\texttt{dom}}$ as the new input $\mathbf{x}_i^{'}$.

$$\mathbf{x}_i^{'} = \mathbf{x}_i \oplus \mathbf{emb}_{d_i}^{\texttt{dom}} \tag{13}$$

Since $\mathbf{emb}_{d_i}^{\texttt{dom}}$ enables to explicitly represent which domain the input comes from and the adversarial learning is helpful to detect domain-invariant knowledge, we propose a novel adversarial DE method for effective feature separation.

As shown in the right of Figure 3, we employ two independent BiLSTM encoders to capture domain-specific and domain-invariant features by the utilization of domain embedding and adversarial learning. Concretely, a BiLSTM takes $\mathbf{x}_i$ as the input and its output $\mathbf{h}_i^{\texttt{inv}}$ is fed into the GRL for the adversarial learning. Simultaneously, the other BiLSTM uses $\mathbf{x}_i^{'}$ as the input and obtains the output $\mathbf{h}_i^{\texttt{spe}}$.

Dataset	BC	PC	PB	ZX
train	16,339	6,885	5,129	1,645
dev	997	1,300	1,300	500
test	1,992	2,600	2,600	1,100
unlabeled	-	349,922	291,481	33,792

Table 1: Data statistics in sentence number

Then, we concatenate $\mathbf{h}_i^{\texttt{inv}}$ and $\mathbf{h}_i^{\texttt{spe}}$ as the final contextualized word representation $\mathbf{h}_i'$, which is used for dependency parsing by shared MLP and biaffine operations. In addition, the orthogonality loss is used to divergent the domain-specific and domain-invariant representations. Finally, the entire model is optimized by a joint loss, which is the same defined as L_{fa}^*.

3.4 Fine-tuning BERT with All Target-domain Unlabeled Data

Recently proposed contextualized word representations, such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2018) can further improve parsing performance by a large margin (Clark et al., 2018; Li et al., 2019a). Remarkably, BERT has been proven effective on a variety of natural language processing tasks (Devlin et al., 2019). Recently, researchers pay more attention to updating BERT representations with additional corpus and achieve great progress on BERT applications (Gururangan et al., 2020). Motivated by the successful utilization of BERT and BERT's strong capability of word representations, we propose to fine-tune BERT model parameters with all unlabeled data to obtain more reliable representations.

First, we use the released Chinese BERT-Based model as the original BERT model.[2] Then, we fine-tune BERT on the unlabeled data using the parameters in the original BERT model as the starting point. To save computational resource, we merge all train/unlabeled data of all domains as one unlabeled dataset for fine-tuning BERT once. Thus, the same fine-tuned BERT model is used for all three-target domains. Since the product comment (PC) and product blog (PB) data are user-generated independent sentences without context information, we remove the next sentence loss and tune the BERT model parameters with only language model loss. Following Li et al. (2019a), we train all BERT-enhanced models by replacing the pre-trained word embedding $\mathbf{emb}_{w_i}^{\texttt{word}}$ with the fixed BERT representation $\mathbf{rep}_{w_i}^{\texttt{BERT}}$. For $\mathbf{rep}_{w_i}^{\texttt{BERT}}$, we first compute the mean value of the 4-top layer BERT outputs, and then a linear map is used to reduce the high dimensional outputs into a low dimensional vector.

4 Experiments

Datasets. We use the Chinese multi-domain dependency parsing datasets released at the NLPCC-2019 shared task[3], containing four domains: one source domain which is a balanced corpus (BC) from newswire, three target domains which are the product comments (PC) data from Taobao, the product blog (PB) data from Taobao headline, and a web fiction data named "ZhuXian" (ZX). The detailed data statistics are shown in Table 1.

Evaluation. We use unlabeled attachment score (UAS) and labeled attachment score (LAS) to evaluate the dependency parsing accuracy. Each parser is trained for at most 1000 iterations, and the performance is evaluated on the dev data after each iteration for model selection. We stop the training if the peak performance does not increase in 100 consecutive iterations.

Hyper-parameters. We follow the hyper-parameter settings of Dozat and Manning (2017), such as learning rate and dropout ratios. The loss weights α and β are set to 0.001. The GRL hyper-parameter λ is 10^{-5}. For pre-trained word embeddings, we train word2vec embeddings on Chinese Gigaword Third Edition (Mikolov et al., 2013), consisting of about 1.2 million sentences.

4.1 Single-domain Training

Table 2 presents the parsing accuracy on dev data when each parser is trained on a single-domain training data. First, although PC-train is much smaller than BC-train, the PC-trained parser outperforms the BC-

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

[3] http://hlt.suda.edu.cn/index.php/Nlpcc-2019-shared-task

Trained on	BC	PC	PB	ZX
BC-train	**75.16**	28.16	60.95	64.93
PC-train	48.79	**58.21**	54.82	43.37
PB-train	60.32	31.22	**72.41**	51.58
ZX-train	56.33	20.44	49.82	**69.74**

Table 2: Performance (LAS) on dev data of each parser trained on a single-domain training data.

	PC		PB		ZX		AVG	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Non-adversarial Models								
CON	68.54	60.22	79.55	74.60	**80.92**	**76.15**	**76.34**	**70.32**
FA	**68.58**	**60.46**	78.86	73.34	79.72	75.15	75.72	69.65
DE	68.12	59.83	**79.81**	**74.89**	80.84	75.95	76.26	70.22
Adversarial Models								
CON	68.37	59.85	79.96	74.94	80.52	76.19	76.28	70.33
FA	**69.18**	**61.11**	**80.28**	**74.96**	**81.24**	**76.59**	**76.90**	**70.88**
w/o Ort	69.01	60.98	79.24	74.09	79.40	74.59	75.88	69.89
w/o Fused	69.04	60.72	79.47	74.33	79.56	74.51	76.02	69.85
w/o Fused & Ort	69.10	60.79	78.86	73.34	78.36	73.79	75.44	69.31
DE	**69.35**	**60.37**	**80.31**	**75.21**	**81.12**	**76.71**	**76.93**	**70.76**
w/o Ort	68.74	60.00	79.93	75.03	80.68	75.95	76.45	70.33

Table 3: Results of non-adversarial and adversarial models on dev data. The "w/o Fused" indicates the adversarial model removing the fused target-domain feature representations and "w/o Ort" means training the adversarial models without the orthogonality constraint loss.

trained parser by about 30%, indicating that the target-domain labeled data is useful and important to train a parser specially when there is a large divergence between two domains. Second, the gap between PB-trained and BC-trained parsers is about 11% while the scale of PB-train and PC-train is very close, demonstrating that PB-train is much similar with BC-train. Third, the accuracy of ZX-trained parser is about 5% higher than the BC-trained one. The reason may be that the BC-train data are from the newswire which may contain novels. Overall, the results clearly demonstrate that the model easily achieves good performance when the training and testing data are from the same domain.

4.2 Combining Two Training Datasets

We first train the three representative non-adversarial models with the combination of source- and target-domain data. Then, we conduct detailed ablation study on adversarial models to gain in-depth insight about the effect of different model components.

Results of non-adversarial models. As shown in the top block of Table 3, we can see that CON obviously outperforms FA on PB and ZX domains, but underperforms on the PC domain, demonstrating that the FA approach performs well only when there is a large difference between source and target domains. In addition, we find that the DE model achieves nearly the same accuracy as the CON, indicating that both domain-invariant features in the CON model and domain-specific features in the DE model are equally important for cross-domain dependency parsing.

Results of adversarial models. The results of comparison experiments on adversarial approaches are shown in the bottom block of Table 3. First, we can see that directly applying adversarial network on non-adversarial models even slightly reduces the model performance specially on the CON and FA. The reason may be that the target-domain related parameters are trained inadequately with only a small-scale labeled data. Second, the utilization of the fused word representation and orthogonality constraints enables to obviously enhance the performance of the vallina adversarial models, indicating that the two strategies are helpful for feature separation representations. Finally, we find that our proposed adversarial models consistently outperform the non-adversarial ones, demonstrating that pure word representation is an effective knowledge to improve the accuracy of cross-domain dependency parsing.

	PC		PB		ZX		AVG		PC		PB		ZX		AVG	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
	Non-adversarial Models with BERT								Adversarial Models with BERT							
CON	**73.75**	66.40	**84.40**	80.29	86.13	82.08	81.43	**76.26**	73.65	66.60	84.46	80.66	86.33	82.29	81.48	76.52
FA	72.77	65.03	83.66	79.50	84.45	80.44	80.29	74.99	**74.20**	**66.86**	84.64	80.40	86.05	82.08	81.63	76.45
DE	73.66	66.01	84.38	**80.31**	**86.45**	**82.20**	**81.50**	76.17	73.51	66.27	**84.85**	**80.81**	**86.97**	**83.13**	**81.78**	**76.74**
	Non-adversarial Models with Fine-tuned BERT								Adversarial Models with Fine-tuned BERT							
CON	**74.98**	67.20	84.78	80.79	86.57	82.89	82.11	76.96	74.88	67.38	84.96	81.10	87.01	83.12	82.28	77.20
FA	74.01	66.94	83.98	79.73	85.53	81.72	81.17	76.13	74.90	68.16	**85.20**	81.00	86.65	82.91	82.25	77.36
DE	74.94	**67.64**	**84.90**	**80.93**	**87.05**	**83.25**	**82.30**	**77.27**	**75.63**	**68.68**	85.05	**81.12**	**87.50**	**83.81**	**82.73**	**77.87**

Table 4: Results of different models on dev data regarding the utilization of BERT.

	PC		PB		ZX		AVG	
	UAS	LAS	UAS	LAS	UAS	LAS	UAS	LAS
Yu et al. (2019)*	72.18	64.12	82.57	77.83	80.53	75.84	78.43	72.60
Peng et al. (2019)^{E}	73.16	64.33	83.05	78.57	82.09	77.08	79.43	73.33
Li et al. (2019)^{*B}	**75.25**	**67.77**	**85.53**	**81.51**	**86.14**	**81.65**	**82.30**	**76.98**
	Non-adversarial models							
CON	68.93	60.25	79.02	74.24	78.32	73.25	75.42	69.25
FA	69.33	60.66	78.93	73.96	77.93	72.90	75.40	69.17
DE	69.39	61.08	79.04	74.26	78.54	73.71	75.66	69.68
DE^{B}	73.92	66.20	84.43	80.38	85.02	81.03	81.12	75.87
DE^{FB}	**75.08**	**67.04**	**84.87**	**80.85**	**85.60**	**81.45**	**81.85**	**76.45**
	Adversarial models							
CON	69.57	61.04	79.12	74.25	78.70	73.55	75.80	69.61
FA	70.74	62.33	79.26	74.46	79.01	74.32	76.34	70.37
DE	70.31	61.45	79.26	74.34	79.11	74.46	76.23	70.08
DE^{B}	74.58	66.86	84.77	80.62	85.22	80.98	81.52	76.15
DE^{FB}	**75.93**	**68.34**	**85.07**	**80.99**	**85.94**	**81.45**	**82.31**	**76.93**

Table 5: Final results on test data. With the limited length of the page, we use "*" to denote "ensemble models", "*E*" to denote "model with ELMo", "*B*" to denote "model with BERT", and "*FB*" to denote "model with fine-tuning BERT".

4.3 Utilization of Unlabeled Dataset

In order to obtain more reliable domain-related word representations that benefit for cross-domain dependency parsing, we exploit the large-scale target-domain unlabeled data to fine-tune BERT model parameters. Detailed comparative experiments are conducted to verify the effectiveness of fine-tuned BERT representations, and the results are shown in Table 4. First, we find that BERT as deep and contextualized word representation has a strong representational capacity and achieve higher performances among all models. Second, we can see that fine-tuning BERT with unlabeled data can significantly improve the performances of both adversarial and non-adversarial models, demonstrating that BERT can learn domain-related knowledge and produce more reliable contextualized word representations by fine-tuning operation. Third, the performance gaps between all BERT-enhanced models reduces sharply, but the adversarial models still consistently improve the accuracy of non-adversarial ones, indicating adversarial learning and fine-tuning BERT are complementary for word representations that can benefit from each other. Overall, we find that fine-tuning BERT is an effective method to leverage unlabeled data and the adversarial learning is still useful on BERT-enhanced models.

4.4 Final Results

Table 5 shows the final results and makes a comparison with previous works on test data. We report the parsing accuracy of our baseline models in the second block and our proposed adversarial models in the last block. First, comparing the results on the two blocks, we can clearly see that all adversarial models outperform the non-adversarial ones, indicating that adversarial learning is helpful to detect pure yet effective domain-invariant and domain-specific representations. Second, the utilization of BERT can improve the accuracy of both non-adversarial and adversarial models by a large margin, and the fine-

tuned BERT enables to further enhance parsing performances. The reason may be that fine-tuning BERT with a large-scale target-domain unlabeled data is extremely useful to learn more reliable word representations. Finally and foremost, although the baseline becomes much stronger with fine-tuned BERT, our proposed adversarial approach still achieves higher performance, demonstrating the adversarial learning and fine-tuning BERT are complementary and mutual benefit for word representations.

We also give the main results newly submitted at NLPCC-2019 shared task in the top block of Table 5. Yu et al. (2019) attempt to combine the power of self-training and ensemble models to improve the model performance. Peng et al. (2019) re-implement the DE method to learn explicit domain information and further improve the parsing accuracy with ELMo. Li et al. (2019c) directly update the BERT representations by the parsing loss, and tri-training is used to augment the target domain training data. Our final single model achieves nearly the same performance as the top submitted system at the shared task (Li et al., 2019c) without the complex model ensemble process.

5 Related Work

Domain adaptation has been a long-standing yet challenging research topic. Here we try to briefly summarize the representative approaches for both unsupervised and semi-supervised domain adaptation.

5.1 Unsupervised Domain Adaptation

Due to the lack of target-domain labeled data, previous researches mostly focus on the unsupervised domain adaptation. *Self-training* is a simple method to incorporate unlabeled data into the new model, which first annotates the unlabeled data with the existing model, and then train a new model with the combination of newly generated data and actual labeled data (Yarowsky, 1995). As a typical unsupervised approach, self-training has proven effective on cross-domain constituency parsing (McClosky et al., 2006) and dependency parsing (Yu et al., 2015), but there are also many failed works. Charniak (1997) reports either minor improvements or significant damage for parsing by using self-training. Clark et al. (2003) show the same findings on POS-tagging task. *Co-training* is another way to utilize the unlabeled data (Blum and Mitchell, 1998). It leverages multiple learners to annotate the unlabeled data respectively, and then arguments the training data with the newly labeled data when multiple learners agree on the annotation labels. Sarkar (2001) and Steedman et al. (2003) demonstrate that co-training is helpful for unsupervised cross-domain parsing. However, it still is a challenge to select the appropriate labeled data for self-training and co-training.

5.2 Semi-supervised Domain Adaptation

Semi-supervised domain adaptation assumes the model is trained with all source- and target-domain labeled data. Most recently, Li et al. (2019c) and Yu et al. (2019) reveal that newly generated target-domain data by self-training or tri-training and model ensemble can improve the cross-domain parsing performance significantly. The *model ensemble* method is a commonly used strategy to integrate different parsing models in dependency parsing (Nivre and McDonald, 2008). However, all these approaches require to retrain parser repeatedly, making them difficult for practical applications.

Daumé III (2007) for the first time proposes the *FA* method on sequence labeling task, which distinguishes domain-specific and domain-invariant with different feature extractors. Kim et al. (2016) successfully employ the FA technique on neural network, which uses a shared and m private BiLSTM encoders for feature separation. As another direction, Li et al. (2019b) propose to utilize an extra domain embedding to indicate the domain information of the input word, and they find that the parsing accuracy of the DE model is obviously higher than other semi-supervised approaches.

The *adversarial learning* is a commonly used strategy to extract pure domain-invariant representations that does not belong to a particular domain as much as possible (Goodfellow et al., 2014; Bousmalis et al., 2016; Kim et al., 2017; Britz et al., 2017; Cao et al., 2018; Guo et al., 2018; Zeng et al., 2018; Adams et al., 2019). Most relevantly, Sato et al. (2017) employ adversarial network to the FA and CON methods, finding that there is little gains and even damage the performance, specially when the scale of target-domain labeled training data is small. Motivated by these works, we apply adversarial learning on

three typical semi-supervised domain adaptation, i.e., CON, FA, and DE with two useful strategies, i.e., fused target-domain word representation and orthogonality constraints to detect more pure yet effective word representations, thus further boosting the performance of cross-domain dependency parsing.

6 Conclusions

This work successfully exploits adversarial learning and fine-tuning BERT to model pure yet effective word representations that benefit for the cross-domain dependency parsing. We have demonstrated the effectiveness of adversarial learning and fine-tuning BERT by applying them to three representative semi-supervised approaches. Experimental results show that our proposed adversarial approaches achieve consistent improvement, and fine-tuning BERT further boosts parsing accuracy by a large margin. The detailed comparison experiments demonstrate that both the fused target-domain word representation and orthogonality loss are useful for adversarial models to alleviate the domain-invariant representations from being contaminated by domain-specific ones. The analysis on the utilization of BERT indicates that the fine-tuning BERT with the target-domain unlabeled data encourages BERT to learn more reliable contextualized word representations, leading to a large improvement over using off-the-shelf BERT on both non-adversarial and adversarial models.

6.1 Acknowledgments

We thank our anonymous reviewers for their helpful comments. This work was supported by National Natural Science Foundation of China (Grant No. 61525205, 61876116), a project funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions, and was partially supported by the joint research project of Soochow University.

References

Oliver Adams, Matthew Wiesner, Shinji Watanabe, and David Yarowsky. 2019. Massively multilingual adversarial speech recognition. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of NAACL-HLT*, pages 96–108. Association for Computational Linguistics.

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. In *Proceedings of ACL*, pages 2442–2452.

Avrim Blum and Tom M. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In Peter L. Bartlett and Yishay Mansour, editors, *Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998, Madison, Wisconsin, USA, July 24-26, 1998*, pages 92–100. ACM.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016*, pages 343–351.

Denny Britz, Quoc V. Le, and Reid Pryzant. 2017. Effective domain mixing for neural machine translation. In *Proceedings of WMT*, pages 118–126.

Pengfei Cao, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2018. Adversarial transfer learning for chinese named entity recognition with self-attention mechanism. In *Proceedings of EMNLP*, pages 182–192.

Eugene Charniak. 1997. Statistical parsing with a context-free grammar and word statistics. In *Proceedings of AAAI*, pages 598–603.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of EMNLP*, pages 740–750.

Stephen Clark, James R. Curran, and Miles Osborne. 2003. Bootstrapping pos-taggers using unlabelled data. In *Proceedings of HLT-NAACL*, pages 49–55.

Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc V. Le. 2018. Semi-supervised sequence modeling with cross-view training. In *Proceedings of EMNLP*, pages 1914–1925.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of ACL*, pages 256–263.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186.

Timothy Dozat and Christopher Manning. 2017. Deep biaffine attention for neural dependency parsing. abs/1611.01734.

Yaroslav Ganin and Victor S. Lempitsky. 2015. Unsupervised domain adaptation by backpropagation. In *Proceedings of ICML*, pages 1180–1189.

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron C. Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proceedings of NIPS*, pages 2672–2680.

Jiang Guo, Darsh J. Shah, and Regina Barzilay. 2018. Multi-source domain adaptation with mixture of experts. In *Proceedings of EMNLP*, pages 4694–4703.

Suchin Gururangan, Ana Marasovic, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel R. Tetreault, editors, *Proceedings of ACL*, pages 8342–8360. Association for Computational Linguistics.

Christian Hadiwinoto and Hwee Tou Ng. 2017. A dependency-based neural reordering model for statistical machine translation. In *Proceedings of AAAI*, pages 109–115.

Young-Bum Kim, Karl Stratos, and Ruhi Sarikaya. 2016. Frustratingly easy neural domain adaptation. In *Proceedings of COLING, Osaka, Japan*, pages 387–396.

Young-Bum Kim, Karl Stratos, and Dongchan Kim. 2017. Adversarial adaptation of synthetic or stale data. In *Proceedings of ACL*, pages 1297–1307.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *TACL*, 4:313–327.

Ying Li, Zhenghua Li, Min Zhang, Rui Wang, Sheng Li, and Luo Si. 2019a. Self-attentive biaffine dependency parsing. In *Proceedings of IJCAI*, pages 5067–5073.

Zhenghua Li, Xue Peng, Min Zhang, Rui Wang, and Luo Si. 2019b. Semi-supervised domain adaptation for dependency parsing. In *Proceedings of ACL*, pages 2386–2395.

Zuchao Li, Junru Zhou, Hai Zhao, and Rui Wang. 2019c. Cross-domain transfer learning for dependency parsing. In *Proceedings of NLPCC*, pages 835–844.

David McClosky and Eugene Charniak. 2008. Self-training for biomedical parsing. In *Proceedings of ACL*, pages 101–104.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the main conference on human language technology conference of the North American Chapter of the Association of Computational Linguistics*, pages 152–159.

Ryan T. McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajic. 2005. Non-projective dependency parsing using spanning tree algorithms. In *Proceedings of HLT/EMNLP*, pages 523–530.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Joakim Nivre and Ryan T. McDonald. 2008. Integrating graph-based and transition-based dependency parsers. In *Proceedings of ACL*, pages 950–958.

Youngmin Park and Sangwoo Kang. 2019. Natural language generation using dependency tree decoding for spoken dialog systems. *IEEE Access*, 7:7250–7258.

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. In *Proceedings of NLPCC*, pages 760–771.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Anoop Sarkar. 2001. Applying co-training methods to statistical parsing. In *Proceedings of NAACL*.

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

Mark Steedman, Anoop Sarkar, Miles Osborne, Rebecca Hwa, Stephen Clark, Julia Hockenmaier, Paul Ruhlen, Steven Baker, and Jeremiah Crim. 2003. Bootstrapping statistical parsers from small datasets. In *Proceedings of EACL*, pages 331–338.

Qingrong Xia, Zhenghua Li, Min Zhang, Meishan Zhang, Guohong Fu, Rui Wang, and Luo Si. 2019. Syntax-aware neural semantic role labeling. In *Proceedings of AAAI*, pages 7305–7313.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In Hans Uszkoreit, editor, *Proceedings of ACL*, pages 189–196. Morgan Kaufmann Publishers / ACL.

Juntao Yu, Mohab Elkaref, and Bernd Bohnet. 2015. Domain adaptation for dependency parsing via self-training. In *Proceedings of IWPT*, pages 1–10.

Nan Yu, Zonglin Liu, Ranran Zhen, Tao Liu, Meishan Zhang, and Guohong Fu. 2019. Domain information enhanced dependency parser. In *Proceedings of NLPCC*, pages 801–810.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In *Proceedings of EMNLP*, pages 447–457.