# Tri-training for Dependency Parsing Domain Adaptation

SHU JIANG, ZUCHAO LI, HAI ZHAO, BAO-LIANG LU, and RUI WANG,
Department of Computer Science and Engineering, Shanghai Jiao Tong University, Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Qing Yuan Research Institute, Shanghai Jiao Tong University, MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University

In recent years, the research on dependency parsing focuses on improving the accuracy of the domain-specific (in-domain) test datasets and has made remarkable progress. However, there are innumerable scenarios in the real world that are not covered by the dataset, namely, the out-of-domain dataset. As a result, parsers that perform well on the in-domain data usually suffer from significant performance degradation on the out-of-domain data. Therefore, to adapt the existing in-domain parsers with high performance to a new domain scenario, cross-domain transfer learning methods are essential to solve the domain problem in parsing. This paper examines two scenarios for cross-domain transfer learning: semi-supervised and unsupervised cross-domain transfer learning. Specifically, we adopt a pre-trained language model BERT for training on the source domain (in-domain) data at the subword level and introduce self-training methods varied from tri-training for these two scenarios. The evaluation results on the NLPCC-2019 shared task and universal dependency parsing task indicate the effectiveness of the adopted approaches on cross-domain transfer learning and show the potential of self-learning to cross-lingual transfer learning.

CCS Concepts: • **Computing methodologies → Lexical semantics**; **Transfer learning;**

Additional Key Words and Phrases: Tri-training, dependency parsing, domain adaptation, transfer learning

**ACM Reference format:**
Shu Jiang, Zuchao Li, Hai Zhao, Bao-Liang Lu, and Rui Wang. 2021. Tri-training for Dependency Parsing Domain Adaptation. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 21, 3, Article 48 (December 2021), 17 pages.
https://doi.org/10.1145/3488367

Shu Jiang and Zuchao Li contributed equally to this research.

This article is partially supported by Key Projects of National Natural Science Foundation of China (U1836222 and 61733011). Bao-Liang Lu is supported in part by the National Natural Science Foundation of China (61976135), SJTU Trans-med Awards Research (WF540162605), the Fundamental Research Funds for the Central Universities, the 111 Project, and the China Southern Power Grid (Grant No. GDKJXM20185761). Rui Wang is supported by the National Natural Science Foundation of China (6217020129), CCF-Tencent Open Fund (RAGR20210119), and Shanghai Municipal Science and Technology Major Project (2021SHZDZX0102).

Authors' address: S. Jiang, Z. Li, H. Zhao (corresponding author), B.-L. Lu (corresponding author), and R. Wang, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Qing Yuan Research Institute, Shanghai Jiao Tong University, MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, 800 Dongchuan Rd, Minhang District, Shanghai 200240, China; emails: jshmjs45@gmail.com, charlee@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn, {bllu, wangrui12}@sjtu.edu.cn.



2375-4699/2021/12-ART48 $15.00
https://doi.org/10.1145/3488367

## 1 INTRODUCTION

Dependency parsing is a critical task for understanding textual content, to reveal the syntactic structure of linguistic components by analyzing their dependencies, whose results can help the downstream task model better understand the input text [3, 5, 7, 21]. Since dependency syntax is an artificially defined language structure, making high-quality labeled data relies on human analysis, and it is very time-consuming and painful. While most dependency parsers currently demonstrate very excellent performance [2, 9, 10, 22], the existing labeled data for dependency parsing is very limited in domain aspects, which means that the parser has very few domains to work with though currently performing well. If the model trained from the existing domain data is directly applied to the new domain, its performance will be significantly downgraded [52]. He et al. [15] show that high-precision dependency syntax can be beneficial for downstream tasks, while low-precision syntax is unhelpful and even harmful to the performance [15]. Hence, cross-domain dependency parsing has become a significant issue in applying syntactic analysis results in the downstream **natural language processing** (**NLP**) systems.

Transfer learning refers to the source domain $\mathcal{D}_S$ and the source task $\mathcal{T}_S$ to improve the effectiveness of the target domain $\mathcal{D}_T$ and the target task $\mathcal{T}_T$, i.e., the information of $\mathcal{D}_S$ and $\mathcal{T}_S$ is transferred to $\mathcal{D}_T$ and $\mathcal{T}_T$. In this article, we concentrate on cross-domain transfer learning, namely, domain adaptation, a type of isomorphic transfer learning where $\mathcal{T}_S = \mathcal{T}_T$. According to whether the target domain or the target task has labeled data or not, transfer learning can be divided into three types: supervised, semi-supervised, and unsupervised transfer learning (domain adaptation).

With recent advances in NLP transfer learning, two typical approaches are very effective: pre-trained language model and tri-training. Pre-trained language models [33, 34] are proved very useful for several NLP tasks like **part-of-speech** (**POS**) tagging, **named entity recognition** (**NER**), constituent parsing, dependency parsing, and **machine reading comprehension** (**MRC**). Utilizing large-scale unsupervised (unlabeled) text corpus data to train a language model, we use supervised target task data (labeled) to fine-tune the language model and train the target model at the same time. In this way, the fine-tuned language model emphasizes the more language information included in specific tasks. Tri-training [54] aims at picking up some high-quality auto-labeled training instances from unlabeled data using bootstrapping methods. Ruder and Plank [36] found that the classic tri-training, with some additions, provides a more robust baseline for unsupervised transfer learning, and the results are even better than the current state-of-the-art systems trained on the same domain.

In this article, we describe our domain adaptation for dependency parsing in the NLPCC-2019 shared task [32] and universal dependency parsing [47] benchmarks. Our system performs dependency parsing training using the pre-trained language model BERT as our encoder and the biaffine attention as the scorer of dependency arcs and relations at the subword level. Then, it applies the graph-based dependency tree search algorithm with the token mask to obtain the final dependency tree at the word level. The pre-trained language model BERT is used to learn the language features from a large-scale unlabeled corpus (such as Wikipedia). We adopt the tri-training variant method to use the unlabeled in-domain data for iterative training and use the provided development set for model selection during model iteration. For the unsupervised sub-task, we only use the in-domain unlabeled data for tri-training, while for the semi-supervised sub-task, the in-domain training data and the auto-parsed data are mixed for tri-training. Briefly, our contributions can be concluded as follows:

— We propose a subword-level dependency parser, adopting a token mask-based inference algorithm to avoid the incorrect intra- and inter-word dependencies and restore a well-defined dependency tree. We also show the difference between subword-level parsing and word-level

parsing. The empirical evaluation demonstrates that our proposed parser at the subword level is a strong baseline and limited decoding on a more complete representation is a better choice.

—Based on our proposed strong baselines, we also study the domain adaptation task in dependency parsing with both supervised and unsupervised settings. We introduce and adopt a tri-training approach for better domain adaptation performance. Evaluation results of models using tri-training on popular benchmarks verify its effectiveness.

—Furthermore, we model the cross-lingual transfer learning task in dependency parsing as a special domain adaptation task. With this modeling, we apply our proposed strong baseline with tri-training to cross-lingual dependency parsing, and the experiment results show its feasibility and success.

## 2 RELATED WORK

### 2.1 Dependency Parsing

Our work is based on a typical parser style, graph-based dependency parsing, and the other typical parser style is transition-based dependency parsing.

**Graph-based parsers** treat dependency parsing as a task in which a maximum spanning dependency tree is constructed in a graph composed of all lexical nodes (words) and their possible arcs with different weights or probabilities. Before applying **deep neural networks** (**DNN**), the parsing community mainly studied higher order inference algorithms [19, 27, 29] and approximation algorithms [48, 49] for graph-based dependency parsing. After the DNN is proven effective, Kiperwasser and Goldberg [18] present a neural graph-based parser in which the authors utilize the bidirectional LSTM [16] to obtain the contextualized vector for each word. Then, they concatenate the vector of each possible head-dependent pair and feed the pair vector into a **multi-layer perceptron** (**MLP**)-based attention layer to score the arc existence probability. Dozat and Manning [10] adopt the biaffine attention instead of bilinear or traditional MLP-based attention [13, 18] to score the dependency arc and corresponding relations and achieve state-of-the-art results on **Penn Treebank** (**PTB**) [28].

**Transition-based parsers** regard dependency parsing as a step-by-step action sequence prediction task from left to right (or other order). *Buffer* maintains words that have not yet been parsed, and *stack* stores the words whose head has not been seen or their dependents have not been all parsed. The predicted action changes the contents of the *buffer* and *stack*, and affects the prediction of the next action. For transition-based dependency parsing, the researchers mainly focus on beam search to obtain action sequences with the highest-scores [51], richer features, and dynamic oracle training methods [11] in the pre-DNN era. Chen and Manning [6] present a simple and effective transition-based dependency parser using neural networks. Ma et al. [25] propose a transition-based neural network architecture that combines the pointer networks with an internal *stack* to track the status of the depth-first search in the decoding procedure and achieves comparable state-of-the-art results.

### 2.2 Subword/Character Level Dependency Parsing

Traditional dependency parsing is usually defined at the word level (as shown in the top part of Figure 1). For Chinese or similar language, which consists of continuous characters and lacks obvious boundaries between words, the word segmentation is the preliminary pre-processing step for dependency parsing. However, the parsing pipeline for Chinese and other similar languages will suffer from some limitations such as error propagation and out-of-vocabulary problems. Therefore, dependency parsing based on more fine-grained lexical units like subwords or characters has been

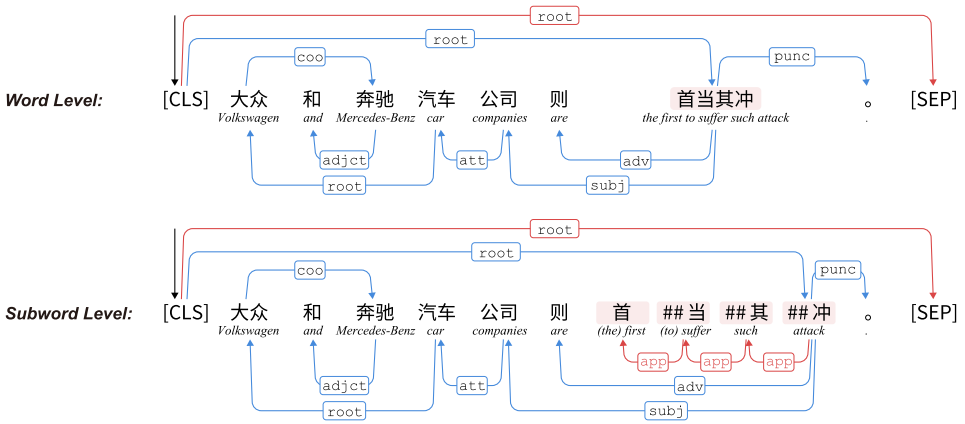


Fig. 1. Dependency tree at the word and subword level.

studied by some researchers. (The bottom of Figure 1 is an example of dependency parsing at the subword level.)

Hatori et al. [14] first propose a transition-based model for Chinese word segmentation, POS tagging, and dependency parsing by introducing a pseudo-inter-character arc inside the word. Zhang et al. [50] further expand the model [14] mentioned above, and regard the internal relation between the characters of a word as a real existed dependency arc. Thus, the dependency is divided into two categories: inter-word dependencies and intra-word dependencies. Kurita et al. [20] is the first neural approach for fully joint Chinese analysis, and it is known to avoid the dangers of error propagation on the pipeline models. Yan et al. [43] propose a unified model for joint the character-level word segmentation and dependency parsing in Chinese, which integrates these two tasks into one biaffine graph-based parsing model by adding a real inter-character dependency like the work by Zhang et al. [50].

### 2.3 Domain Adaptation in NLP

The original intention of transfer learning is to save the time of labeling samples manually. Transfer learning methods can be divided into four categories:

- —Instance-based transfer learning: The data similar to the target domain is found in the source domain, and the weight of the data is adjusted to match the new data with the data in the target domain. Then training and learning are carried out to obtain the model suitable for the target domain.
- —Feature-based transfer learning: When the source domain and the target domain contain some common cross-features, we can transform the features of the source domain and the target domain into the same space by feature transformation. In this way, the data distribution of the source domain and the target domain are the same.
- —Parameter-based transfer learning: Source domain and target domain share the parameters from the model, which is trained by a large amount of data in the source domain and applied to the target domain for prediction.
- —Relation-based transfer learning: When the two domains are similar, they will share some analogous relations and apply the logical relationship learned from the source domain to the target domain for domain transfer learning.

The classical supervised machine learning methods are usually based on isolated learning with a single model and often evaluated on the same domain test dataset. This paradigm requires a

large number of training examples and performs best on the well-defined and closest domain. However, for new domains, the model performance tends to degrade. Recently, some effective domain adaptation methods have emerged in the NLP community.

Yosinski et al. [45] argue that the DNN is an ideal carrier of transfer learning because the DNN obtains the hierarchical feature representation through pre-training and then applies the high-level features to a specific task. The underlying features of the DNN (such as edge information and color information in the computer vision field, and characters information, words information, and syntax information in the NLP field) are invariably for the different task or domain models. Howard and Ruder [17] propose **Universal Language Model Fine-tuning** (**ULMFiT**) technique that can be applied to any NLP task. It is an effective transfer learning method and provides a standard transfer learning process in NLP. The language model learns the underlying features, which confirms the conclusion of Yosinski et al. [45]. After the emergence of BERT [34], it is easier to learn the Transformer structure and a more considerable amount of data through a deeper network, and the DNN-based pre-trained language model begins to take the stage of NLP transfer learning. Clark et al. [9] propose **Cross-View Training** (**CVT**) with DNN, and it performs well in dependency parsing.

The semi-supervised domain adaptation and the unsupervised domain adaptation are challenging due to the small or unlabeled target domain data. For NLP domain adaptation, how to use the sizeable unlabeled text of the target domain has become the focus of research. The current mainstream method is using the existing model to label the target domain data to get auto-labeled data and then picking-up high-quality auto-parsed instances for training with bootstrapping methods, such as self-training [44], co-training [4], and tri-training [54]. Li et al. [24] propose an ambiguity-aware ensemble training framework for semi-supervised dependency parsing, which gains success in dependency parsing with unlabeled data. Chen and Zhang [8] further study the application of unlabeled data to dependency parsing. Yu et al. [46] propose a self-training approach that uses confidence-based methods to select additional training samples and improve parsing accuracy for out-of-domain texts. Rotman and Reichart [35] propose a **Deep Contextualized Self-training** (**DCST**) algorithm for dependency parsing based on the integration of contextualized embedding models into a neural dependency parser. The embedding models are trained on word tagging schemes extracted from the trees generated by the base parser on unlabeled data.

For dependency parsing, because of its specialty of lacking large scale labeled corpus, we can not directly apply the data selection-based domain adaptation method of other tasks (such as Neural Machine Translation). However, we can easily obtain large-scale unlabeled data of the same domain. Therefore, the research based on large-scale unlabeled data is very crucial.

### 2.4 Cross-lingual Dependency Parsing

In our work, we consider cross-lingual dependency parsing as a domain adaptation task. In general, it trains a dependency parser on the source language and then directly performs on the target languages. McDonald et al. [30] use the delexicalized models combined with a standardized POS tagset to transfer models between languages directly. Ma and Xia [26] propose an unsupervised projective dependency parsing approach for resource-poor languages, using existing resources from a resource-rich source language with entropy regularization. Wu and Dredze [40] demonstrate the broader cross-lingual potential of mBERT[1] (multilingual-BERT) [34] as a zero-shot language transfer model. Ahmad et al. [1] explore how the design of neural architectures affects cross-lingual transfer learning and find that order-free models perform better than order-sensitive ones. Sun et al. [37] propose an effective cross-lingual universal dependency parsing framework with

[1] https://github.com/google-research/bert/blob/a9ba4b8d7704c1ae18d1b28c56c0430d41407eb1/multilingual.md.

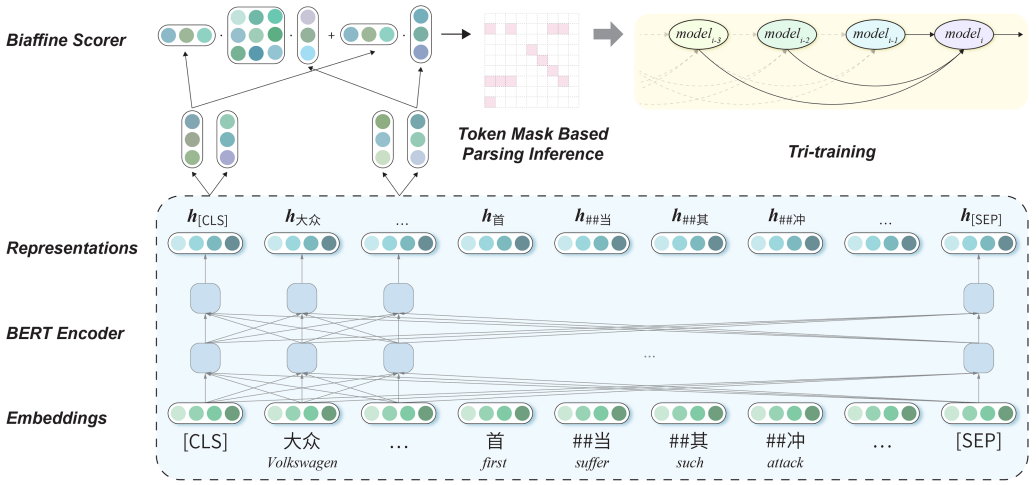


Fig. 2. The architecture of our subword-level dependency parsing model.

self-training strategy for transferring parser from only one source monolingual treebank to any other target languages without treebank available. Cross-lingual word clusters [38] and lexicon mapping [12, 42] are also used in cross-lingual models.

## 3 MODEL

### 3.1 Overview

In this work, we propose a general parser adapted to both monolingual and cross-lingual scenarios. Figure 2 illustrates the structure of our subword-level graph-based dependency parsing model with tri-training [54]. It consists of four parts: BERT encoder, biaffine scorer [10], token mask-based parsing inference, and tri-training.

### 3.2 BERT-encoded Representation

For the encoder, we employ the Transformer [39] encoder with pre-trained weights and subword embeddings from BERT [34] instead of the randomly initialized BiLSTM with the pre-trained word embeddings. We do not use any other information except subword to prevent error propagation, such as POS tag, which reduces the reliance on the model. For the detailed task definition in our model, we use *[CLS]* defined in BERT as the virtual *ROOT* node for dependency parsing and *[SEP]* as the end tag of the sequence. Then, we create a new dependent arc with *root* relation, pointing from *[CLS]* (*ROOT*) to *[SEP]*, which is shown in Figure 1.

In addition, to represent the dependencies within the word (inter-character), we follow the work of Zhang et al. to build the subword-level dependency tree. As shown in Figure 3, the subword end of a word, like "##冲(attack)", is regarded as the node (if a word has no subwords, the subword end is defined as the word itself), where the word creates its dependency with other words. For the subwords, which are not the subword end of a word, like "首(first)", "##当(suffer)", "##其(such)", we add an *app* dependency relation pointing from its successor subword to itself.

Formally, given an input sequence of $n$ tokens $\mathbf{x} = w_1, w_2, \ldots, w_n$, which is tokenized by a subword tokenizer, we obtain the contextualized representation of $w_i$ from the pre-trained BERT model as $\mathbf{h}_i$:

$$\mathbf{h}_i = \text{BERT}(w_i). \tag{1}$$

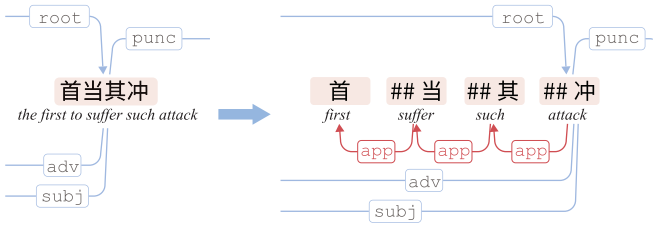


Fig. 3. Converting the word-level dependency tree to subword-level.

### 3.3 Biaffine Scorer

The head-dependent pair is scored with a Biaffine scorer. Based on the BERT-encoded representation $\mathbf{h}_i$, we use the **multi-Layer perceptron** (**MLP**) and deep biaffine attention [10] to score each possible head and dependent for further role-specific representation.

$$\mathbf{h}_i^{(arc-dep)} = \mathrm{MLP}^{(arc-dep)}(\mathbf{h}_i), \tag{2}$$

$$\mathbf{h}_j^{(arc-head)} = \mathrm{MLP}^{(arc-head)}(\mathbf{h}_j), \tag{3}$$

$$\mathbf{y}_i^{(arc)} = H^{(arc-head)} U^{(1)} \mathbf{h}_i^{(arc-dep)} + H^{(arc-head)} \mathbf{u}^{(2)}, \tag{4}$$

where the matrix $H$ is the stack of vectors $\mathbf{h}$ after MLP, $U^{(1)} \in \mathbb{R}^{k \times k}$, $\mathbf{u}^{(2)} \in \mathbb{R}^{k \times 1}$, and $k$ is the dimension of MLP. For relational pair classification, a similar biaffine scorer is adopted. The only difference is that the dimension of the score shifts from 1 to the size of the dependency relation vocabulary.

### 3.4 Token Mask-Based Parsing Inference

In the original dependency parser at the word level, the **Minimum Spanning Tree** (**MST**) algorithm is regarded as the search (inference) algorithm to ensure the dependency tree is well-formed at test time. Since the subword-level has an intra-word (inter-character) dependency arc, we propose a token mask-based MST search algorithm to guarantee the original segmentation of the task (that is, the final dependency tree is restored to the original word level.

Suppose the original word-level MST algorithm is used to search the dependency tree for subword-level dependency graphs. In that case, it may generate incorrect intra-word (inter-character) dependencies and inter-word dependencies, failing to restore a well-defined dependency tree at the word level.[2] Therefore, it is necessary to build some hard constraints on the score (weight) of the graph edges.

Figure 4 illustrates a typical example of three important types of masks:

- —Words with no subwords: its valid choice is the subword end of all words except itself, like "大众(*Volkswagen*)" in the example.
- —Subwords which are not subword end of a word: the dependency of such subwords must be its successor subword, like "首(*first*)","## 当(*suffer*)","## 其(*such*)".
- —Subword ends: subword ends are the same as words without subwords, like "## 冲(*attack*)".

We can multiply the scoring matrix predicted from the model by the mask matrix to ensure a word-level well-defined dependency tree.

[2]For the training phase, there is no need to consider this issue at all. As with other graph-based models, the predicted tree at training time is the one where each word is the dependent of its highest-scoring head, including intra-word and inter-word dependencies.


Fig. 4. An example of token mask for parsing inference.

Although our proposed model has features from the subword level, it does not contain the word segmentation procedure and utilizes the original word segmentation information at inference time. Thus, we can use this gold word segmentation information to obtain the token range within the word and between the words to remove the illegal head by the token mask.

### 3.5 Training Objective in Tri-training

The model is trained to optimize the probability of the dependency tree $y$ when given a sentence $x$: $P_\theta(y|x)$, which can be factorized as:

$$P_\theta(y|x) = \prod_{i=1}^{l} P_\theta\left(y_i^{arc}, y_i^{rel}|x_i\right),$$

where $\theta$ represents learnable parameters, $l$ denotes the length of the processing sentence, and $y_i^{arc}$, $y_i^{rel}$ denote the highest scoring head and dependency relation for node $x_i$. It is represented as the negative likelihood loss $\mathcal{L}$:

$$\mathcal{L} = (-\log P_\theta(y^{arc}|x)) + (-\log P_\theta(y^{rel}|x)).$$

When training with the combined labeled and auto-parsed data in supervised tri-training, the objective is to maximize the mixed likelihood (minimize the negative likelihood loss):

$$\mathcal{L} = \mathcal{L}_g + \alpha \cdot \mathcal{L}_a,$$

where $\alpha$ is the confidence for auto-parsed data at the subword level, which is variable according to the number of tri-training iterations.

## 4 EXPERIMENTS

### 4.1 Tasks

In this work, we evaluate our proposed method with two typical tasks. The first task is monolingual cross-domain adaptation. NLPCC-2019 shared task on cross-domain dependency parsing [32] provides 17 K sentences from a balanced corpus as the source domain (*BC*), and three target domains where 10 K sentences are from product comments (*PC*), 8 K sentences are from product blogs (*PB*), and 3 K are sentences from the web fiction named "Zhuxian" (*ZX*). We set up

Table 1. Data Statistics in NLPCC-2019 Shared Task (in Sentence Number)

| | Train | Dev | Test | Unlabeled |
|---|---|---|---|---|
| *BC* | 16.3 K | 1 K | 2 K | 0 |
| *PB* | 5.1 K | 1.3 K | 2.6 K | 300 K |
| *PC* | 6.2 K | 1.3 K | 2.6 K | 350 K |
| *ZX* | 1.6 K | 0.5 K | 1.1 K | 30 K |

four sub-tasks with two cross-domain scenarios, i.e., unsupervised and semi-supervised domain adaptation:

—**Unsupervised** domain adaptation assumes that there is no labeled training data for the target domain. For example, in the unsupervised domain adaptation scenario, when the target domain is *PC*, the labeled training data of *PC* is unavailable, but the unlabeled data of *PC* is allowed to use.
—**Semi-supervised** domain adaptation means that there exists a labeled training dataset for the target domain.

The statistics of the NLPCC-2019 shared task datasets can be seen in Table 1.

Considering the NLPCC-2019 shared task is labeled only on the Chinese, in order to explore the generalization ability of the model, we also experiment in domains of other languages. In the CoNLL-2018 shared task [47], there are 16 languages with two or more treebanks from different sources, also usually from different domains. As CoNLL-2018 has the most domains in English, we choose English as another language for domain adaptation. Among them, the *EWT* is used as the source domain, *GUM*, *LinES*, and *PUD* as the target domain. We also conduct experiments with the two domain adaptation settings described earlier.

In addition, if cross-lingual dependency parsing is also considered as a domain adaptation task, we evaluate it on the CoNLL-2018 shared task based on the same settings. We select English as the source language and some low resource languages (Thai, Vietnamese, Indonesian, Chinese, Japanese, Hindi, and Korean) as the target language.

**ALGORITHM 1:** An variant tri-training method for *unsupervised* domain adaptation

**for** $i \in \{1 \ldots 3\}$ **do**
  $model_i \leftarrow \text{train\_model}(t_S, d_S, random_i)$
**end for**
**for** $i \in \{4 \ldots N\}$ **do**
  $a_T \leftarrow \text{parse}(model_{i-3}, model_{i-2}, u_T)$
  $m_T \leftarrow \text{merge}(a_T, t_S)$
  $model_i \leftarrow \text{finetune\_model}(model_{i-1}, m_T, d_T)$
**end for**

**ALGORITHM 2:** An variant tri-training method for *semi-supervised* domain adaptation

**for** $i \in \{1 \ldots 3\}$ **do**
  $model_i \leftarrow \text{train\_model}(t_S, d_S, random_i)$
**end for**
$model_4 \leftarrow \text{finetune\_model}(model_3, t_T, d_T)$
**for** $i \in \{5 \ldots N\}$ **do**
  $a_T \leftarrow \text{parse}(model_{i-3}, model_{i-2}, u_T)$
  $m_T \leftarrow \text{merge}(a_T, t_T, t_S)$
  $model_i \leftarrow \text{finetune\_model}(model_{i-1}, m_T, d_T)$
**end for**

### 4.2 Model Setup

For the hyper-parameter of models trained on the source domain data, the encoder is initialized by the pre-trained language model: Chinese simplified and traditional BERT with 12-layer, 768-hidden, 12-heads, and 110 M parameters. When not otherwise specified, our model uses 100-dimensional

arc space and 128-dimensional relation space. We follow the downstream task fine-tuning settings by Peters et al., with learning rate $lr = 5e^{-5}$. The maximum number of training epochs is set to 30. For the models in the fine-tuning process, the learning rate is reduced to $2e^{-3}$, and the number of fine-tuning epochs is set to 3.

For the hyper-parameter in the English domain adaptation model, we replace the Chinese BERT with the English BERT-Large model with 24-layer, 1024-hidden, 16-heads, 340 M parameters. While for the cross-lingual dependency parsing, we change it to the multilingual cased BERT-Base model with 104 languages, 12-layer, 768-hidden, 12-heads, and 110 M parameters.

For unsupervised and semi-supervised domain adaptation, we use slightly different tri-training variants as presented in Algorithms 1 and 2. Unlike traditional tri-training methods, we do not select data from auto-parsed data but merge all auto-parsed data with source domain data and target domain data (semi-supervised domain adaptation) instead. The gold data and auto-parsed data are assigned different weights (confidence) to achieve the goal of domain adaptation.[3] In the algorithm, we use $t_S$ and $t_T$ to represent the gold labeled training dataset in the source domain ($BC$) and the target domain ($T \in \{PB, PC, ZX\}$). $d_S$ and $d_T$ are denoted as the development dataset in the corresponding domains. $u_T$, $a_T$, and $m_T$ indicate the *unlabeled data*, the *auto-parsed data,* and the *mixed data* in the target domain. $model_i$ represents the model on the $i$ th training iteration with random seed $random_i$.

The number of iteration tri-training steps $N$ is set to 20. In each model training or fine-tuning process, we use the **labeled attachment score** (**LAS**) on the development dataset to select the model and only save the model with a higher score on the development dataset of the corresponding target domain for subsequent use.[4] When the iteration step $i < 10$, we set the confidence of the auto-parsed data $\alpha = 0.2$, and $\alpha = 0.5$ at $i >= 10$.

## 5 MAIN RESULTS

### 5.1 Cross-domain Dependency Parsing

In Table 2, we compare our full model against previous work on the NLPCC-2019 shared task test dataset. Our baseline is a modification to the model of Dozat and Manning, which uniformly handled the dependency arcs and relations. As for the systems for comparison, *PRIS_DP* is the baseline model of the NLPCC-2019 shared task. *NNU* proposed by the team from Nanjing Normal University is based on the stack-pointer networks. Their evaluation results are fetched from the official website.[5] The model proposed by Li et al. [23] merges the source- and target-domain training data and employs the recent contextualized word representations with fine-tuning. The difference between our model and the other models reported is that we propose a dependency parser at the subword level adopting a token mask-based inference algorithm and employ a novel domain adaptation method, tri-training. Our model outperforms significantly over the previous works for both unsupervised and semi-supervised settings. The proposed tri-training method brings absolute improvements of 17.46% and 6.3% LAS on the unsupervised and semi-supervised settings, respectively, which are on par with the best-published scores.

In addition, for better reflecting the contributions of our model other than the biaffine scorer [10], we also list the results of initializing Transformer without using any pre-trained language

[3]Due to the tri-training iterative training process, the unlabeled data will be much larger than the gold annotation data. In order to balance the training process of the model, we repeat the gold data to achieve the same amount of data as the unlabeled data and then perform data shuffle during training.

[4]The initial score for each model run is set to 0, so at least one model will be saved for each training session.

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

Table 2. Evaluation Results on the NLPCC-2019 Shared Task Test Dataset with Unsupervised and Semi-Supervised Settings

| Systems | | Unsupervised | | | | Semi-supervised | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | PC | PB | ZX | AVG | PC | PB | ZX | AVG |
| PRIS_DP | UAS | 39.81 | 67.31 | 69.55 | 58.89 | 69.30 | 77.37 | 74.35 | 73.67 |
| | LAS | 26.27 | 60.40 | 61.51 | 49.39 | 60.35 | 72.10 | 68.28 | 66.91 |
| NNU [41] | UAS | - | - | - | - | 70.97 | 80.59 | 79.33 | 79.96 |
| | LAS | - | - | - | - | 61.82 | 75.85 | 74.35 | 70.68 |
| Li et al. [23] | UAS | - | 67.55 | 68.44 | - | - | 82.05 | 80.44 | - |
| | LAS | - | 61.01 | 59.55 | - | - | 77.16 | 75.11 | - |
| BiAF + tri-training | UAS | 39.96 | 67.91 | 69.42 | 59.10 | 70.25 | 78.03 | 78.43 | 75.57 |
| | LAS | 26.44 | 61.32 | 61.66 | 49.81 | 60.95 | 71.82 | 72.05 | 68.27 |
| **Ours** | UAS | 60.50 | 81.61 | 79.74 | **73.95** | 75.25 | 85.53 | 86.14 | **82.31** |
| | LAS | 49.49 | 76.77 | 74.32 | **66.86** | 67.77 | 81.51 | 81.65 | **76.98** |

The bold values indicates the significance level $p$-value $< 0.01$.

Table 3. UAS and LAS of Unsupervised Domain Adaptation Experiments on Test Datasets of four English Domains from CoNLL-2018 Shared Task, Together with Baseline, +BERT, and +Tri-training for Comparison

| | **Baseline** | **+BERT** | **+Tri-training** |
|---|---|---|---|
| | UAS [LAS] | UAS [LAS] | UAS [LAS] |
| EWT | 83.32 [80.46] | 93.29 [91.13] | **94.43 [91.27]** |
| GUM | 81.09 [76.68] | 87.70 [84.16] | **89.68 [86.62]** |
| LinES | 80.71 [75.26] | 86.03 [82.26] | **88.08 [83.52]** |
| PUD | 86.77 [83.49] | 93.03 [90.85] | **93.81 [92.09]** |
| **avg** | 82.97 [78.97] | 90.01 [87.10] | **91.50 [88.37]** |

The bold values indicates the significance level $p$-value $< 0.01$.

model, namely *BiAF+tri-training*, in the table. Although there is a gap contrasting with the model using BERT, it has performance advantages compared with system *PRIS_DP*, which only uses BiAF. It shows that tri-training is an effective method with language model pre-training and improves the performance of the parser for both unsupervised and semi-supervised scenarios.

Table 3 presents all test results on four English domains of CoNLL-2018 datasets. We first train the parser for each domain on the data of four domains, respectively, as a baseline and further report the results of parsers with further enhancement using mBERT. Then we report the effect of using tri-training on the strong BERT baseline. Compared with the baseline, our model with pre-trained BERT yields strong performance on all domains without exception and performs better than the baseline method with a large margin of 8.13% LAS on average. Moreover, applying our tri-training method to the robust BERT model can further boost the model performance by 1.27%, demonstrating the effectiveness of our proposed methods. On the other hand, it indicates that tri-training is generally beneficial to domain adaptation problems.

### 5.2 Cross-lingual Dependency Parsing

We also report the scores of cross-lingual dependency parsing in Table 4. For the model training, we first merge the training datasets (if any) of all domains and train the multilingual subword embedding with the unlabeled text using the fastText toolkit [31]. The model is only trained on

Table 4. UAS and LAS of Unsupervised Cross-Lingual Experiments on Test Datasets of Seven Languages from CoNLL-2018 Shared Task, Together with Baseline, +BERT, and +Tri-training for Comparison

| | **Baseline** | **+BERT** | **+Tri-training** |
|---|---|---|---|
| | UAS [LAS] | UAS [LAS] | UAS [LAS] |
| English | 83.32 [80.46] | 92.11 [89.67] | **93.55 [90.76]** |
| Thai | 1.22 [0.57] | 39.25 [22.41] | **41.07 [28.95]** |
| Vietnamese | 33.25 [19.68] | 51.07 [36.73] | **53.24 [43.77]** |
| Indonesian | 45.12 [33.10] | 69.23 [51.01] | **70.15 [52.10]** |
| Chinese | 41.38 [29.96] | 51.71 [35.69] | **52.46 [36.07]** |
| Hindi | 29.37 [11.44] | 35.82 [16.56] | **37.23 [18.85]** |
| Korean | 32.25 [20.12] | 52.92 [36.95] | **54.30 [38.09]** |
| **avg** | 37.98 [27.90] | 56.01 [41.28] | **57.42 [44.08]** |

The bold values indicates the significance level $p$-value $< 0.01$.

the English language *EWT* domain and evaluated on other languages in zero-shot. We find that the overall performance of cross-lingual dependency parsing is improved with our method, which suggests that the mBERT is better for dealing with cross-lingual tasks than the multilingual fastText subword embedding. Specially, we observe that our system shows a remarkable advance on Thai compared with the baseline. This advance can be attributed to the multilingual capacity of BERT that can cover a wide range of language patterns by pre-training on a large corpus. With the aid of the proposed tri-training method, our system improves over the cross-lingual task, verifying that tri-training is valid for domain adaptation and cross-lingual tasks. Based on this result, we can observe that cross-lingual tasks can be carried out as cross-domain tasks due to the commonalities between different languages, though with greater differences. It implies that we can attempt to apply most of the domain adaptation approaches to the cross-lingual transfer learning task, especially after an available pre-trained language model.

## 6 ABLATION STUDY

### 6.1 Tri-training

To explore how the tri-training works, we depict the learning curve on domain adaptation by recording the LAS results on the tri-training among three domains (*PB*, *PC*, *and ZX*), based on the unsupervised setting in the NLPCC-2019 shared task. The illustration is shown in Figure 5. The start point (step = 0) represents the performance of the baseline. We observe that the performance of the model is successively boosted when the training processes. The curve proves that the tri-training method can further improve model performance, and in-domain unlabeled data is generally efficient for dependency parsing. We also show in Figure 6 the curve of performance changing with the tri-training step on the cross-lingual transfer learning task. Comparing Figures 5 and 6, we found that the trends of cross-domain transferring and cross-lingual transferring are basically similar. On the one hand, it illustrates the generalization of our proposed tri-training approach for transfer learning tasks. On the other hand, it illustrates the commonality between cross-domain transferring and cross-lingual transferring in task modeling.

### 6.2 Subword or Word?

NLP models based on deep learning, usually suffer from the representation of OOV issues when encountering rare words like morphologically complex words and named entities [53]. In order to prove the role of subword in the parsing domain adaptation, we also perform an experimental

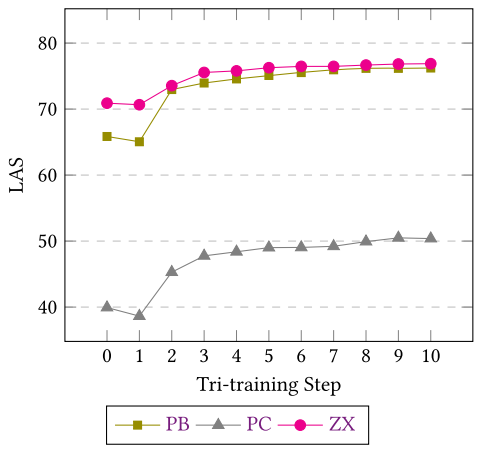


Fig. 5. Performances on dev dataset from NLPCC-2019 shared task with unsupervised setting.

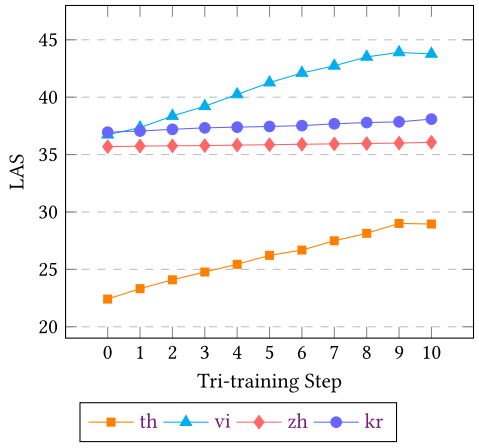


Fig. 6. Performances on test datasets of four languages from CoNLL-2018 shared task with unsupervised setting.

Table 5. Word and Subword Level Evaluation Results on NLPCC-2019 Shared Task Dev Dataset with Both Unsupervised and Semi-supervised Settings

| System | Unsupervised | | | Semi-supervised | | |
|---|---|---|---|---|---|---|
| | PC | PB | ZX | PC | PB | ZX |
| Word | 51.70 | 76.37 | 71.34 | 73.54 | 80.73 | 83.35 |
| | 39.35 | 70.10 | 65.28 | 66.38 | 76.16 | 79.23 |
| Subword | 52.24 | 76.47 | 71.93 | 73.6 | 80.78 | 83.82 |
| | 39.93 | 70.90 | 65.85 | 66.49 | 76.11 | 79.62 |

performance comparison based on subword and word levels, as shown in Table 5. The results show that modeling text into subword units is beneficial for dependency parsing. Employing the subword as the minimal unit of text can reduce the size of vocabulary as well as the parameter size with better performance.

### 6.3 Token Mask or Last Subwords?

For dependency parsing, since the theory is based on word-level, and most of the current mainstream pre-training language models are subword-level, this caused some inconvenience to the application. In order to make up for this difference, one is to extract word-level features from subword-level representation and then perform word-level parsing. The other is to learn from subword-level parsing directly, but the tree conversion from word-level to subword-level is needed. In this work, because of the granularity change resulted by tree conversion, our model utilizes a token mask to build hard constraints on the score (weight) of the graph edges. We also consider using the boundary subwords (like first/last subwords) to represent the word and comparing the performance of these two strategies. As shown in Table 6, the results on test datasets of four English domains demonstrate that the token mask inference is a better strategy for dependency parsing. It avoids generating incorrect intra- and inter-word dependencies, restores a well-defined dependency tree at the word level, and makes the most use of contextualized representation for the encoder.

Table 6. Comparison of Two Strategies on Test Datasets of Four English Domains from CoNLL-2018 Shared Task with Unsupervised Settings

| | **BiAF with Last Subword** UAS [LAS] | **BiAF with Token Mask** UAS [LAS] |
|---|---|---|
| EWT | 94.13 [91.01] | **94.43 [91.27]** |
| GUM | 89.55 [86.40] | **89.68 [86.62]** |
| LinES | 87.59 [83.23] | **88.08 [83.52]** |
| PUD | 93.44 [91.82] | **93.81 [92.09]** |
| **avg** | 91.18 [88.12] | **91.50 [88.37]** |

The bold values indicates the significance level $p$-value $< 0.01$.

## 7 CONCLUSION AND FUTURE WORK

This article presents a domain adaptation model for dependency parsing. The evaluation results on the benchmarks show that our proposed approaches can yield significantly improved results over cross-domain dependency parsing and even cross-lingual dependency parsing. This work discloses the potential of tri-training for dependency parsing domain adaptation. According to the fact that several models with different performance can be strengthened in the process of tri-training, we can perform ensemble on these models to obtain higher cross-domain or cross-lingual performance improvements in the future.

## REFERENCES


[1] Wasi Ahmad, Zhisong Zhang, Xuezhe Ma, Eduard Hovy, Kai-Wei Chang, and Nanyun Peng. 2019. On difficulties of cross-lingual transfer with order differences: A case study on dependency parsing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Association for Computational Linguistics, Minneapolis, Minnesota, 2440–2452. DOI: https://doi.org/10.18653/v1/N19-1253

[2] 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 the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 2442–2452. DOI: https://doi.org/10.18653/v1/P16-1231

[3] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. 2015. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, 344–354. DOI: https://doi.org/10.3115/v1/P15-1034

[4] Avrim Blum and Tom M. Mitchell. 1998. Combining labeled and unlabeled data with co-training. In *Proceedings of the Eleventh Annual Conference on Computational Learning Theory, COLT 1998*, Peter L. Bartlett and Yishay Mansour (Eds.), ACM, Madison, Wisconsin, 92–100. DOI: https://doi.org/10.1145/279943.279962

[5] Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, 1466–1477. DOI: https://doi.org/10.18653/v1/P16-1139

[6] Danqi Chen and Christopher Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Doha, Qatar, 740–750. DOI: https://doi.org/10.3115/v1/D14-1082

[7] Kehai Chen, Rui Wang, Masao Utiyama, Lemao Liu, Akihiro Tamura, Eiichiro Sumita, and Tiejun Zhao. 2017. Neural machine translation with source dependency representation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 2846–2852. DOI: https://doi.org/10.18653/v1/D17-1304

[8] Wenliang Chen and Min Zhang. 2015. *Semi-Supervised Dependency Parsing*. Springer, Singapore. DOI: https://doi.org/10.1007/978-981-287-552-5_4

[9] Kevin Clark, Minh-Thang Luong, Christopher D. Manning, and Quoc Le. 2018. Semi-Supervised sequence modeling with cross-view training. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Brussels, 1914–1925. DOI: https://doi.org/10.18653/v1/D18-1217

[10] Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *Proceedings of the 5th International Conference on Learning Representations 2017*. OpenReview.net, Toulon, France. Retrieved from https://openreview.net/forum?id=Hk95PK9le.

[11] Yoav Goldberg and Joakim Nivre. 2012. A dynamic oracle for arc-eager dependency parsing. In *Proceedings of COLING 2012*. The COLING 2012 Organizing Committee, Mumbai, India, 959–976. Retrieved from https://www.aclweb.org/anthology/C12-1059.

[12] Jiang Guo, Wanxiang Che, David Yarowsky, Haifeng Wang, and Ting Liu. 2015. Cross-lingual dependency parsing based on distributed representations. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Association for Computational Linguistics, Beijing, China, 1234–1244. DOI: https://doi.org/10.3115/v1/P15-1119

[13] Kazuma Hashimoto, Caiming Xiong, Yoshimasa Tsuruoka, and Richard Socher. 2017. A joint many-task model: Growing a neural network for multiple NLP tasks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, 1923–1933. DOI: https://doi.org/10.18653/v1/D17-1206

[14] Jun Hatori, Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2012. Incremental joint approach to word segmentation, POS tagging, and dependency parsing in Chinese. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Jeju Island, Korea, 1045–1053. DOI: https://www.aclweb.org/anthology/P12-1110

[15] Shexia He, Zuchao Li, Hai Zhao, and Hongxiao Bai. 2018. Syntax for semantic role labeling, To Be, Or not to be. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 2061–2071. DOI: https://doi.org/10.18653/v1/P18-1192

[16] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780. DOI: https://doi.org/10.1162/neco.1997.9.8.1735

[17] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 328–339. DOI: https://doi.org/10.18653/v1/P18-1031

[18] Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics* 4 (2016), 313–327. DOI: https://doi.org/10.1162/tacl_a_00101

[19] Terry Koo and Michael Collins. 2010. Efficient third-order dependency parsers. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Uppsala, Sweden, 1–11. https://www.aclweb.org/anthology/P10-1001.

[20] Shuhei Kurita, Daisuke Kawahara, and Sadao Kurohashi. 2017. Neural joint model for transition-based Chinese syntactic analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Vancouver, Canada, 1204–1214. DOI: https://doi.org/10.18653/v1/P17-1111

[21] Omer Levy and Yoav Goldberg. 2014. Dependency-Based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 302–308. DOI: https://doi.org/10.3115/v1/P14-2050

[22] Zuchao Li, Jiaxun Cai, Shexia He, and Hai Zhao. 2018. Seq2seq dependency parsing. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, 3203–3214. Retrieved from https://www.aclweb.org/anthology/C18-1271.

[23] Zhenghua Li, Xue Peng, Min Zhang, Rui Wang, and Luo Si. 2019. Semi-supervised domain adaptation for dependency parsing. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Florence, Italy, 2386–2395. DOI: https://doi.org/10.18653/v1/P19-1229

[24] Zhenghua Li, Min Zhang, and Wenliang Chen. 2014. Ambiguity-aware ensemble training for semi-supervised dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 457–467. DOI: https://doi.org/10.3115/v1/P14-1043

[25] Xuezhe Ma, Zecong Hu, Jingzhou Liu, Nanyun Peng, Graham Neubig, and Eduard Hovy. 2018. Stack-Pointer networks for dependency parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 1403–1414. DOI: https://doi.org/10.18653/v1/P18-1130

[26] Xuezhe Ma and Fei Xia. 2014. Unsupervised dependency parsing with transferring distribution via parallel guidance and entropy regularization. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 1337–1348. DOI: https://doi.org/10.3115/v1/P14-1126

[27] Xuezhe Ma and Hai Zhao. 2012. Fourth-Order dependency parsing. In *Proceedings of COLING 2012: Posters*. The COLING 2012 Organizing Committee, Mumbai, India, 785–796. Retrieved from https://www.aclweb.org/anthology/C12-2077.

[28] Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a large annotated corpus of english: The penn treebank. *Computational Linguistics* 19, 2 (1993), 313–330.

[29] Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Trento, Italy, 81–88. Retrieved from https://www.aclweb.org/anthology/E06-1011.

[30] Ryan McDonald, Slav Petrov, and Keith Hall. 2011. Multi-Source transfer of delexicalized dependency parsers. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Edinburgh, Scotland, UK., 62–72. Retrieved from https://www.aclweb.org/anthology/D11-1006.

[31] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. 2018. Advances in pre-training distributed word representations. In *Proceedings of the 11th International Conference on Language Resources and Evaluation, LREC 2018*, Nicoletta Calzolari, Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Kôiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asunción Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga (Eds.), European Language Resources Association (ELRA), Miyazaki, Japan, 52–55. Retrieved from http://www.lrec-conf.org/proceedings/lrec2018/summaries/721.html.

[32] 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. *Natural Language Processing and Chinese Computing - 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9–14, 2019, Proceedings, Part II* 11839 (2019), 760–771. DOI: https://doi.org/10.1007/978-3-030-32236-6_69

[33] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2227–2237. DOI: https://doi.org/10.18653/v1/N18-1202

[34] Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. Association for Computational Linguistics, New Orleans, Louisiana, 2227–2237. DOI: https://doi.org/10.18653/v1/N18-1202

[35] Guy Rotman and Roi Reichart. 2019. Deep contextualized self-training for low resource dependency parsing. *Transactions of the Association for Computational Linguistics* 7 (March 2019), 695–713. DOI: https://doi.org/10.1162/tacl_a_00294

[36] Sebastian Ruder and Barbara Plank. 2018. Strong baselines for neural semi-supervised learning under domain shift. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 1044–1054. DOI: https://doi.org/10.18653/v1/P18-1096

[37] Kailai Sun, Zuchao Li, and Hai Zhao. 2021. Cross-lingual Universal Dependency Parsing Only from One Monolingual Treebank. arXiv:cs.CL/2012.13163. Retrieved from https://arxiv.org/abs/2012.13163.

[38] Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Montréal, Canada, 477–487. Retrieved from https://www.aclweb.org/anthology/N12-1052.

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*. NIPS, Long Beach, 6000–6010. Retrieved from http://papers.nips.cc/paper/7181-attention-is-all-you-need.

[40] Shijie Wu and Mark Dredze. 2019. Beto, Bentz, Becas: The surprising cross-lingual effectiveness of BERT. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. Association for Computational Linguistics, Hong Kong, China, 833–844. DOI: https://doi.org/10.18653/v1/D19-1077

[41] Zhentao Xia, Likai Wang, Weiguang Qu, Junsheng Zhou, and Yanhui Gu. 2020. Neural network based deep transfer learning for cross-domain dependency parsing. In *Proceedings of the Artificial Intelligence and Security*, Xingming Sun, Jinwei Wang, and Elisa Bertino (Eds.), Springer Singapore, Singapore, 549–558.

[42] Min Xiao and Yuhong Guo. 2014. Distributed word representation learning for cross-lingual dependency parsing. In *Proceedings of the 18th Conference on Computational Natural Language Learning*. Association for Computational Linguistics, Ann Arbor, Michigan, 119–129. DOI: https://doi.org/10.3115/v1/W14-1613

[43] Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. A graph-based model for joint Chinese word segmentation and dependency parsing. *Transactions of the Association for Computational Linguistics* 8 (2020), 78–92. DOI: https://doi.org/10.1162/tacl_a_00301

[44] David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Cambridge, Massachusetts, 189–196. DOI: https://doi.org/10.3115/981658.981684

[45] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. 2014. How transferable are features in deep neural networks? In *Proceedings of the Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger (Eds.), Curran Associates, Inc., Montreal, Quebec, Canada, 3320–3328. Retrieved from http://papers.nips.cc/paper/5347-how-transferable-are-features-in-deep-neural-networks.pdf.

[46] Juntao Yu, Mohab Elkaref, and Bernd Bohnet. 2015. Domain adaptation for dependency parsing via self-training. In *Proceedings of the 14th International Conference on Parsing Technologies*. Association for Computational Linguistics, Bilbao, Spain, 1–10. DOI: https://doi.org/10.18653/v1/W15-2201

[47] Daniel Zeman, Jan Hajič, Martin Popel, Martin Potthast, Milan Straka, Filip Ginter, Joakim Nivre, and Slav Petrov. 2018. CoNLL 2018 Shared Task: Multilingual parsing from raw text to universal dependencies. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*. Association for Computational Linguistics, Brussels, Belgium, 1–21. DOI: https://doi.org/10.18653/v1/K18-2001

[48] Hao Zhang, Liang Huang, Kai Zhao, and Ryan McDonald. 2013. Online learning for inexact hypergraph search. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Seattle, Washington, 908–913. Retrieved from https://www.aclweb.org/anthology/D13-1093.

[49] Hao Zhang and Ryan McDonald. 2012. Generalized higher-order dependency parsing with cube pruning. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, Jeju Island, Korea, 320–331. Retrieved from https://www.aclweb.org/anthology/D12-1030.

[50] Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2014. Character-Level Chinese dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 1326–1336. DOI: https://doi.org/10.3115/v1/P14-1125

[51] Yue Zhang and Stephen Clark. 2008. A tale of two parsers: Investigating and combining graph-based and transition-based dependency parsing. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Honolulu, Hawaii, 562–571. Retrieved from https://www.aclweb.org/anthology/D08-1059.

[52] Yi Zhang and Rui Wang. 2009. 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*. Association for Computational Linguistics, Suntec, Singapore, 378–386. https://www.aclweb.org/anthology/P09-1043.

[53] Zhuosheng Zhang, Hai Zhao, Kangwei Ling, Jiangtong Li, Zuchao Li, Shexia He, and Guohong Fu. 2019. Effective subword segmentation for text comprehension. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 27, 11 (Nov 2019), 1664–1674. DOI: https://doi.org/10.1109/TASLP.2019.2922537

[54] Zhi-Hua Zhou and Ming Li. 2005. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Transactions on Knowledge and Data Engineering* 17, 11 (Nov 2005), 1529–1541. DOI: https://doi.org/10.1109/TKDE.2005.186

Received December 2020; revised May 2021; accepted August 2021