Cross-Domain Transfer Learning for Dependency Parsing

Zuchao Li[1,2,3], Junru Zhou[1,2,3], Hai Zhao[1,2,3,⋆], Rui Wang[4]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering, Shanghai Jiao Tong University, Shanghai, China
[3] MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University, China
[4] National Institute of Information and Communications Technology (NICT), Kyoto, Japan
{charlee,zhoujunru}@sjtu.edu.cn, zhaohai@cs.sjtu.edu.cn, wangrui@nict.go.jp

Abstract. In recent years, the research of dependency parsing focuses on improving the accuracy of in-domain data and has made remarkable progress. However, the real world is different from a single scenario dataset, filled with countless scenarios that are not covered by the dataset, namely, out-of-domain. As a result, parsers that perform well on the in-domain data often suffer significant performance degradation on the out-of-domain data. Therefore, in order to adapt the existing in-domain parsers with substantial performance to the new domain scenario, cross-domain transfer learning techniques are essential to solve the domain problem in parsing. In this paper, we examine two scenarios for cross-domain transfer learning: semi-supervised and unsupervised cross-domain transfer learning. Specifically, we adopt a pretrained language model BERT for training on the source domain (in-domain) data at subword level and introduce two tri-training variant methods for the two scenarios so as to achieve the goal of cross-domain transfer learning. The system based on this paper participated in NLPCC-2019-shared-task on cross-domain dependency parsing and won the first place on the "subtask3-un-open" and "subtask4-semi-open" subtasks, indicating the effectiveness of the approaches adopted.

Keywords: Cross-Domain · Transfer Learning · Dependency Parsing.

1 Introduction

Dependency parsing is a critical task for understanding textual content which is to reveal the syntactic structure of linguistic components by analyzing their

⋆ Corresponding author. This paper was partially supported by National Key Research and Development Program of China (No. 2017YFB0304100) and Key Projects of National Natural Science Foundation of China (U1836222 and 61733011).

dependencies whose results can help the downstream task model better understand the input text [12, 2–4]. 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 demonstrate very good performance currently [7, 5, 1, 13], the existing labeled dependency parsing data are very limited in domain aspects, this means that parser, which currently performs well, has very few domains to work with. If the model trained from existing domain data is directly applied to the new domain, the performance will be greatly downgraded [19]. He et al. [9] shows that high-precision dependency syntax can be helpful for downstream tasks, while low-precision syntax is not only unhelpful but even harmful to the performance. Therefore, cross-domain dependency parsing has become the major challenge for applying syntactic analysis results in realistic downstream natural language processing (NLP) systems.

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

With recent advances in transfer learning of NLP, there are two typical approaches that have shown to be very effective: pretrained language model and tri-training. Pretrained language models [14, 6] have been shown to be very useful features for several NLP tasks like POS Tagging, name entity recognition (NER), constituent parsing, dependency parsing, and machine reading comprehension (MRC). Using large-scale unsupervised (unlabeled) text corpus data to train a language model, and then using supervised target task data (labeled) to finetune the language model and train the target model at the same time, so as to make the finetuned language model more emphasize the language information contained in specific tasks. Tri-training [21] aims to pick up some high-quality auto-labeled training instances from unlabeled data using bootstrapping methods. Ruder and Plank [15] found that the classical bootstrapping algorithms: tri-training, provide a stronger baseline for unsupervised transfer learning with results which are even better than the current state-of-the-art systems trained in the same domain.

In this paper, we report our system participating in NLPCC-2019-shared-task[16]. Our system[5] performs dependency parsing training at the subword level, using pre-trained language model BERT as our encoder, Biaffine attention as the scorer of dependency arcs and relations, and using the graph-based dependency tree search algorithm with the token mask to obtain the final dependency tree at the word level. Among them, we use the pre-trained language model BERT to transfer learn the language features from large-scale of the un-

[5] Our code will be available at `https://github.com/bcmi220/cddp`.

labeled corpus (Wikipedias, etc.). The tri-training variant method is adopted to use the unlabeled in-domain data for iterative training, and the provided development set is used for model selection during model iteration. For unsupervised sub-task, we only use the in-domain unlabeled data for tri-training, while for semi-supervised sub-task, in-domain training data and auto-parsed data were mixed for tri-training. In summary, our contributions can be concluded as follows:

• For Chinese dependency parsing, we need to do word segmentation (CWS) in the first step. Because of the different new word collections of different domains, the dictionary differences of different domains are relatively large, which affects the effect of domain transfer learning. In order to reduce this problem, we perform dependency parsing at Chinese subword level and propose a dependency tree search algorithm for subword level based on token mask, which can restore the dependency tree structure at the word level.

• Tri-training variant methods are proposed, and the results show that they are more effective than the original one for the dependency parsing task.

• The official evaluation results showed that our system achieved the state-of-the-art results on "subtask3-un-open" and "subtask4-semi-open" subtasks, which proved the effectiveness of our method[6].

2 Related Work

2.1 Token Level Dependency Parsing

Traditional dependency parsing is usually defined on word level (as shown in the top part of Figure 1). For Chinese and similar languages, the word segmentation (WS) is the preliminary pre-processing step for dependency parsing. However, the pipeline parsing way for Chinese and other similar languages will suffer from some limitations such as error propagation and out-of-vocabulary (OOV) problems. Therefore, some researchers have studied the dependency parsing based on more fine-grained lexical units (tokens) like subwords, characters, etc (The bottom of Figure 1 is an example of dependency parsing at subword level).

Hatori et al. [8] 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. [18] further expands the model of [8], and regards the internal relation between characters of a word as a real existed dependency arc, thus dividing the dependency into inter-word dependencies and intra-word dependencies. Kurita et al. [11] is the first neural approaches

[6] Subtasks "subtask1-un-closed" and "subtask2-semi-closed" are not our focus. Since our baseline parsing framework is based on BERT and subtasks 1 and 2 prohibit the use of BERT and other external resources, we only use the transformer structure of BERT, without using BERT pretrained weights for initialization. The transformer network of BERT is very deep and the currently offered training dataset is too small to train the deep network well, so we only reached comparable results to other participants. This illustrates the deep neural network need enough data for training.

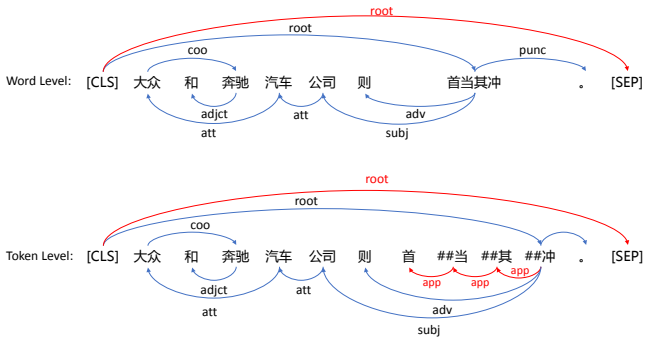

Fig. 1. Dependency tree in word and subword (token) level.

for fully joint Chinese analysis that is known to prevent the error propagation problem of pipeline models. Yan et al. [17] propose a unified model for joint Chinese word segmentation and dependency parsing at the character level which integrates these two tasks in one Biaffine graph-based parsing model by adding a real inter-character dependency like [18].

3 Proposed System

3.1 Overview

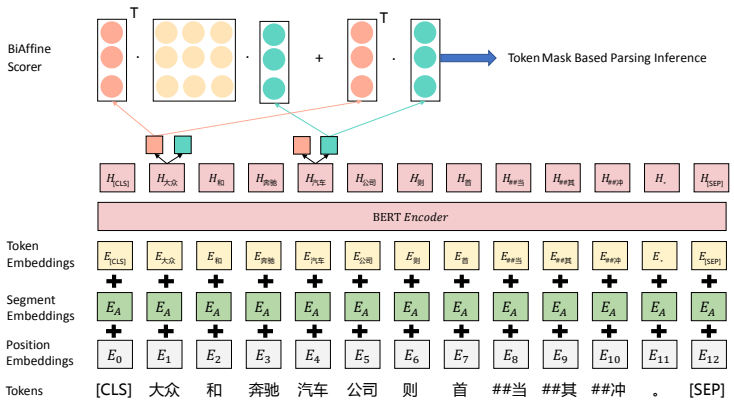

Fig. 2. The system architecture which participated in the shared task.

Figure 2 illustrates the architecture of our subword-level dependency parsing system which participated in the NLPCC-2019-shared-task. Our system is based on graph-based biaffine dependency parser [7], which consists of three parts:

encoder, biaffine scorer and parsing inferencer. We make a few modifications to the graph-based architectures of [7]:

• For the encoder, we use the Transformer encoder with BERT pretrained weights and subword embeddings for initialization instead of the randomly initialized BiLSTM with pretrained word embeddings as input; In order to prevent error propagation, we do not use any other information such as POS tag except subword, which reduces the dependency of the model.

• For the detailed task definition in our system, we use "**[CLS]**" defined in the BERT model as the virtual "**ROOT**" node for dependency parsing and use "**[SEP]**" as the end tag of the sequence, and create a new dependent arc with "root" relation, pointing from "**[CLS]**" ("**ROOT**") to "**[SEP]**". Besides, we follow [18] to add an "**app**" dependency relation to represent the dependencies within the word (inter-character) and we take the subword end of a word as the node (if a word has no subwords, we define the subword end is the word itself) where the word generates its dependency with other words, as shown in Figure 1.

• For the parsing inferencer: in the original word-level dependency parser, the MST algorithm is used 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, and in order to guarantee the original segmentation of the task (that is, the final dependency tree is restored to the original word-level), we propose a token mask based MST search algorithm.

• For the training objective in supervised tri-training phase: In the supervised tri-training phase, since we need to mix golden labeled data with auto-parsed data, we set different confidences on the data to control the loss of training.

3.2 Token Mask Based Parsing Inference

Due to changes in the granularity (from word to subword) of task definitions, the tree search algorithm in the test phase also needs to be changed accordingly. If the original word-level MST algorithm is used to search the dependency tree for subword-level dependency graphs, it may generate incorrect intra-word (inter-character) dependencies and inter-word dependencies, resulting in failure to restore a well-defined dependency tree at the word level[7]. Therefore, it is necessary to make some hard constraints on the score (weight) of the graph edges.

Since we have the original word segmentation information, we can use the word segmentation information to obtain the token range within the word and between the words, so that so that the mask is used to remove the illegal head. Figure 3 is a typical example to illustrate 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.

[7] 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 a dependent of its highest scoring head including intra-word and inter-word dependencies.

	[CLS]	大众	...	首	##当	##其	##冲	...	[SEP]
[CLS]	0	0	0	0	0	0	0	0	0
大众	1	0	1	0	0	0	1	1	0
...									
首	0	0	0	0	1	0	0	0	0
##当	0	0	0	0	0	1	0	0	0
##其	0	0	0	0	0	0	1	0	0
##冲	1	1	1	0	0	0	0	1	0
...									
[SEP]	1	0	0	0	0	0	0	0	0

Fig. 3. An example of token mask for parsing inference.

• Subwords that 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 with no subwords, like "##冲(*attack*)".

Therefore, we multiply the scoring matrix predicted by the model by the mask matrix to ensure a word-level well-defined dependency tree.

3.3 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:

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

where θ 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 implemented as the negative likelihood loss $\mathcal{L}$:

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

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 α is the confidence for auto-parsed data at token level which is variable according to the number of tri-training iterations.

4 Task and Training Details

"NLPCC-2019 Shared Task on Cross-domain Dependency Parsing" [10, 20] provides one source domain (BC) and three target domain (PB, PC, ZX) and setup four subtasks with two cross-domain scenarios, i.e., unsupervised domain adaptation (no target-domain training data) and semi-supervised (with target-domain training data), and two settings, i.e., closed and open.

According to the task requirements, the participant system in the closed task cannot use any external resources. As mentioned earlier, subtask 1 and 2 are not our focus, so the training details here are only for subtask 3 and 4. For the hyper-parameter of models trained on the source domain, The encoder initialized by the pre-trained language model: Chinese simplified and traditional BERT with 12-layer, 768-hidden, 12-heads, 110M parameters. When not otherwise specified, our model uses: 100-dimensional arc space and 128-dimensional relation space. We follow the downstream task finetune settings in [6], with learning rate $lr = 5e^{-5}$. The maximum number of epochs of training is set to 30. While for the models in the tri-training finetune process, the learning rate is reduced to $2e^{-3}$ and the finetune epochs is set to 3.

Algorithm 1 An variant tri-training method for unsupervised DA

for $i \in \{1..3\}$ **do**
  $m_i \leftarrow train\_model(t_s, d_s, random_i)$
end for
for $i \in \{4..N\}$ **do**
  $a_k \leftarrow parse(m_{i-3}, m_{i-2}, u_k)$
  $h_k \leftarrow merge(a_k, t_s)$
  $m_i \leftarrow finetune\_model(m_{i-1}, h_k, d_k)$
end for

Algorithm 2 An variant tri-training method for semi-supervised DA

for $i \in \{1..3\}$ **do**
  $m_i \leftarrow train\_model(t_s, d_s, random_i)$
end for
$m_4 \leftarrow finetune\_model(m_3, t_k, d_k)$
for $i \in \{5..N\}$ **do**
  $a_k \leftarrow parse(m_{i-3}, m_{i-2}, u_k)$
  $h_k \leftarrow merge(a_k, t_k, t_s)$
  $m_i \leftarrow finetune\_model(m_{i-1}, h_k, d_k)$
end for

For unsupervised and semi-supervised domain adaptation (DA), we used slightly different tri-training variants as presented in Algorithm **??**. Unlike traditional tri-training methods, we do not select data from auto-parsed data, but

instead merge all auto-parsed data with source domain data and target domain data (semi-supervised domain adaptation). The golden data and auto-parsed data are assigned different weights (confidence) to achieve the goal of domain adaptation[8]. In the algorithm, we use $t_{s;k}$ to represent the golden labeled training dataset, $d_{s;k}$ denotes the development dataset on the corresponding domain, s represents the source domain (BC), and k represents the target domain ($k \in \{PB, PC, ZX\}$). u_k indicates unlabeled data on the target domain, and a_k indicates auto-parse data and h_k represents the mixed data on the target domain, and m_i represents the model of the i-th iteration training with random seed $random_i$.

We set the number of iteration tri-training steps N=20. In each model training or finetune process, we use the labeled attachment (LAS) score 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[9]. When the iteration step $i < 10$, we set the confidence of auto-parse data to $\alpha = 0.2$, and $\alpha = 0.5$ at $i >= 10$.

Systems	**subtask3-un-open**				**subtask4-semi-open**			
	PC	PB	ZX	AVG	PC	PB	ZX	AVG
PRIS_DP	39.8193	67.3118	69.5582		69.3003	77.3738	74.3534	
	26.2705	60.4097	61.5122	49.3975	60.3548	72.1046	68.2830	66.9141
Nanjing Normal	-	-	-	-	70.9653	80.5866	79.3283	
University	-	-	-	-	61.8239	75.8542	74.3534	70.6772
Ours	60.50	81.61	79.74		75.25	85.53	86.14	
	49.49	76.77	74.32	**66.86**	67.77	81.51	81.65	**76.9767**

Table 1. Official evaluation results of test dataset on subtask 3 and 4.

5 Main Results

Table 1 shows the official evaluation results of test dataset on subtask 3 and 4, the results show that we have obtained the state-of-the-art cross-domain parsing results, among which the advantages of unsupervised domain adaptation are particularly obvious.

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

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

6 Ablation Study

To verify the effect of tri-training, we record the LAS results on tri-training process based on the setting of subtask3, and the results are shown in figure 4. It can be seen from the trend that the tri-training adopted by us is indeed effective.

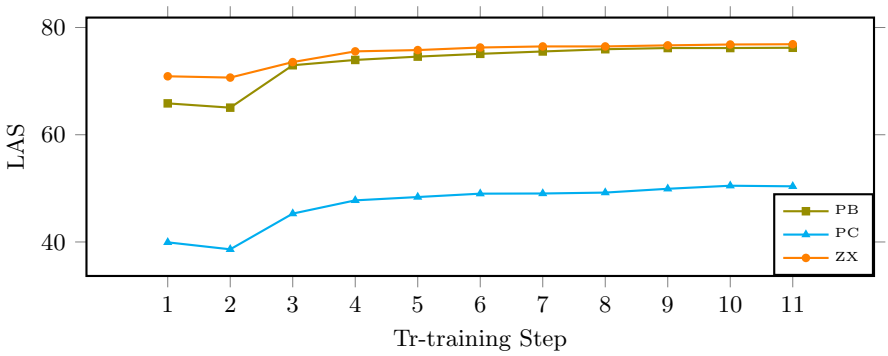

Fig. 4. Performances on dev dataset with settings of subtask3.

In order to demonstrate the role of subword in the parsing domain adaptation, we also performed an extra experimental performance comparison based on subword and word levels as show in table 2. From the comparison results, subword can play an effect in the field migration, but for some areas, the effect may not be very obvious, especially under semi-supervised settings.

System	**subtask3-un-open**			**subtask4-semi-open**		
	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	70.9	65.85	66.49	76.11	79.62

Table 2. Word and subword level evaluation results on dev dataset with setting of subtask 3 and 4.

7 Conclusion

This paper presents our system that participant in the NLPCC2019-shared task. The official evaluation results show that our proposed approaches can yield significantly improved results over cross-domain dependency parsing.

References

1. Andor, D., Alberti, C., Weiss, D., Severyn, A., Presta, A., Ganchev, K., Petrov, S., Collins, M.: Globally normalized transition-based neural networks. In: Proceedings of ACL (2016)
2. Angeli, G., Premkumar, M.J.J., Manning, C.D.: Leveraging linguistic structure for open domain information extraction. In: Proceedings of ACL-IJCNLP (2015)
3. Bowman, S.R., Gauthier, J., Rastogi, A., Gupta, R., Manning, C.D., Potts, C.: A fast unified model for parsing and sentence understanding. In: Proceedings of ACL (2016)
4. Chen, K., Wang, R., Utiyama, M., Liu, L., Tamura, A., Sumita, E., Zhao, T.: Neural machine translation with source dependency representation. In: Proceedings of EMNLP (2017)
5. Clark, K., Luong, M.T., Manning, C.D., Le, Q.: Semi-supervised sequence modeling with cross-view training. In: Proceedings of EMNLP (2018)
6. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
7. Dozat, T., Manning, C.D.: Deep biaffine attention for neural dependency parsing. arXiv preprint arXiv:1611.01734 (2016)
8. Hatori, J., Matsuzaki, T., Miyao, Y., Tsujii, J.: Incremental joint approach to word segmentation, pos tagging, and dependency parsing in chinese. In: Proceedings of ACL (2012)
9. He, S., Li, Z., Zhao, H., Bai, H.: Syntax for semantic role labeling, to be, or not to be. In: Proceedings of ACL (2018)
10. Jiang, X., Li, Z., Zhang, B., Zhang, M., Li, S., Si, L.: Supervised treebank conversion: Data and approaches. In: Proceedings of ACL (2018)
11. Kurita, S., Kawahara, D., Kurohashi, S.: Neural joint model for transition-based chinese syntactic analysis. In: Proceedings of ACL (2017)
12. Levy, O., Goldberg, Y.: Dependency-based word embeddings. In: Proceedings of ACL (2014)
13. Li, Z., Cai, J., He, S., Zhao, H.: Seq2seq dependency parsing. In: Proceedings of COLING. pp. 3203–3214 (2018)
14. Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L.: Deep contextualized word representations. In: Proceedings of NAACL-HLT. pp. 2227–2237 (2018)
15. Ruder, S., Plank, B.: Strong baselines for neural semi-supervised learning under domain shift. In: Proceedings of ACL (2018)
16. Xue Peng, Zhenghua Li, M.Z.R.W.Y.Z.L.S.: Overview of the nlpcc 2019 shared task: Cross-domain dependency parsing. In: Proceedings of NLPCC (2019)
17. Yan, H., Qiu, X., Huang, X.: A unified model for joint chinese word segmentation and dependency parsing. arXiv preprint arXiv:1904.04697 (2019)
18. Zhang, M., Zhang, Y., Che, W., Liu, T.: Character-level chinese dependency parsing. In: Proceedings of ACL (2014)
19. Zhang, Y., Wang, R.: Cross-domain dependency parsing using a deep linguistic grammar. In: Proceedings of ACL-AFNLP (2009)
20. Zhang, Y., Li, Z., Lang, J., Xia, Q., Zhang, M.: Dependency parsing with partial annotations: An empirical comparison. In: Proceedings of IJCNLP (2017)
21. Zhou, Z.H., Li, M.: Tri-training: Exploiting unlabeled data using three classifiers. IEEE TKDE **17**(11), 1529–1541 (2005)