Semi-supervised Domain Adaptation for Dependency Parsing

Zhenghua Li[1], **Xue Peng**[1], **Min Zhang**[1]*, **Rui Wang**[2], **Luo Si**[2]

[1]Institute of Artificial Intelligence, School of Computer Science and Technology, Soochow University

[2]Alibaba Group, China

{zhli13, minzhang}@suda.edu.cn, 20175227031@stu.suda.edu.cn

{masi.wr, luo.si}@alibaba-inc.com

Abstract

During the past decades, due to the lack of sufficient labeled data, most studies on cross-domain parsing focus on unsupervised domain adaptation, assuming there is no target-domain training data. However, unsupervised approaches make limited progress so far due to the intrinsic difficulty of both domain adaptation and parsing. This paper tackles the semi-supervised domain adaptation problem for Chinese dependency parsing, based on two newly-annotated large-scale domain-specific datasets.[1] We propose a simple domain embedding approach to merge the source- and target-domain training data, which is shown to be more effective than both direct corpus concatenation and multi-task learning. In order to utilize unlabeled target-domain data, we employ the recent contextualized word representations and show that a simple fine-tuning procedure can further boost cross-domain parsing accuracy by large margins.

1 Introduction

As a fundamental task in NLP, dependency parsing has attracted a lot of research interest during the past decades due to its multi-lingual applicability in capturing both syntactic and semantic information (Kübler et al., 2009; McDonald et al., 2013). Given an input sentence $S = w_0 w_1 \ldots w_n$, dependency parsing constructs a tree $\mathbf{d} = \{(h, m, l), 0 \leq h \leq n, 1 \leq m \leq n, l \in \mathcal{L}\}$, as depicted in Figure 1, where (h, m, l) is a dependency from the head w_h to the modifier w_m with the relation label l, and w_0 is a pseudo root node.

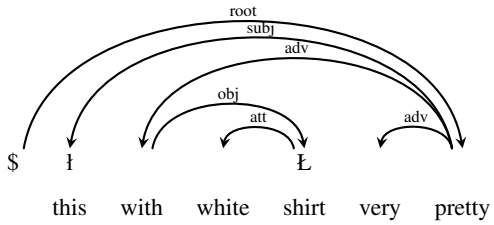

Figure 1: An example from the *product blogs* domain. The English translation is "This looks very pretty with a white shirt."

Recently, dependency parsing has achieved tremendous progress thanks to the strong capability of deep neural networks in capturing long-distance contexts (Chen and Manning, 2014; Dyer et al., 2015; Zhou et al., 2015; Andor et al., 2016; Kiperwasser and Goldberg, 2016; Dozat and Manning, 2017; Ma et al., 2018). Furthermore, contextualized word representations learned from large-scale unlabeled texts under language model training loss (Peters et al., 2018; Radford et al., 2018; Devlin et al., 2018), are proven to be extensively helpful for many NLP tasks including dependency parsing (Che et al., 2018; Clark et al., 2018; Kitaev and Klein, 2018).

However, parsing performance drops dramatically when processing texts that are different from the training data, known as the domain adaptation problem. In fact, with the surge of web data (or user generated content), cross-domain parsing has become the major challenge for applying syntactic analysis in realistic NLP systems. To meet this challenge, the community has organized several shared tasks to attract more research attention (Nivre et al., 2007; Hajič et al., 2009; Petrov and McDonald, 2012).

*Corresponding author

[1]The two domain-specific datasets, plus another one for product comment texts, are also used in the NLPCC-2019 shared task (http://hlt.suda.edu.cn/index.php/Nlpcc-2019-shared-task) on cross-domain Chinese dependency parsing. Please note that the settings for the source-domain training data are different between this work and NLPCC-2019 shared task.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 2386–2395
Florence, Italy, July 28 - August 2, 2019.

Hindered by the lack of sufficient labeled data, most previous works on cross-domain parsing, including the aforementioned shared tasks, assume there is no labeled target-domain training data and thus focus on unsupervised domain adaptation. So far, approaches in this direction have made limited progress, due to the intrinsic difficulty of both domain adaptation and parsing (see discussions in Section 5). On the other hand, due to the extreme complexity and heavy cost, progress on syntactic data annotation on new-domain texts has been very slow, and only several small-scale datasets on web texts have been built, mostly as evaluation data for cross-domain parsing (Foster et al., 2011; Petrov and McDonald, 2012; Kong et al., 2014; Wang et al., 2014).

To meet the above challenges, this paper presents two newly-annotated large-scale domain-aware datasets (over 12K sentences), and try to tackle the task of semi-supervised domain adaptation for Chinese dependency parsing. With the access of both labeled and unlabeled target-domain data, we propose and evaluate several simple approaches and conduct error analysis in order to investigate the following three questions:

Q1: How to effectively combine the source- and target-domain labeled training data?

Q2: How to utilize the target-domain unlabeled data for further improvements?

Q3: Given a certain amount of labeled data, how much data are needed to annotate to reach a certain performance on a new domain?

As our reviewers point out, the semi-supervised domain-adaptation scenario, tackled in this work, is less realistic than the unsupervised counterpart, due to need of labeled target-domain training data, which is usually extremely expensive. However, we believe that this work can be equally valuable and useful when there exist only dozens or hundreds of labeled target-domain training sentences, which may be a feasible compromise for realistic applications of parsing techniques, considering that, as discussed above, purely unsupervised domain adaptation makes very limited progress. We will also release all annotated data at `http://hlt.suda.edu.cn/index.php/SUCDT` and codes at `https://github.com/SUDA-LA/ACL2019-dp-cross-domain`.

2 Data Annotation

In this work, we choose two typical domain-aware web texts for annotation, i.e., product blogs and web fictions. This section introduces the details about the data annotation procedure.

Data selection. The product blog (PB) texts are crawled from the Taobao headline website, which contains articles written by users mainly on description and comparison of different commercial products. After data cleaning and automatic word segmentation, we have collected about 340K sentences. Then, we select 10 thousand sentences with [5, 25] words for manual annotation following the active learning workflow of Jiang et al. (2018). The remaining sentences are used as unlabeled data. For web fictions, we follow the work on cross-domain word segmentation of Zhang et al. (2014), and adopt the popular novel named as "Zhuxian" (ZX, also known as "Jade dynasty"). Among their annotated 4,555 sentences, we select about 3,400 sentences with [5, 45] words for annotation. The remaining 32K sentences of ZX are used as unlabeled data in this work.

Annotation guideline. After comparing several publicly available guidelines for dependency parsing including the universal dependencies (UD) (McDonald et al., 2013), we adopt the guideline released by Jiang et al. (2018) based on three considerations. First, their guideline contains 20 relations specifically designed to capture Chinese dependency syntax for texts of different sources. Second, the 70-page guideline gives very detailed illustrations with many concrete examples. Third, they have constructed a large-scale balanced corpus (BC), which is used as the source-domain labeled data in this work.

Quality Control. We employ about 15 undergraduate students as annotators, and select 5 experienced annotators with linguistic background as the expert annotators. Each annotator is intensively trained to be familiar with the guideline. Based on our browser-based annotation platform, we apply strict double annotation to guarantee the quality of the labeled data. First, each raw sentence with automatic word segmentation is randomly assigned to two annotators. The annotation is accepted if the two submissions are the same. Otherwise, a third expert annotator decides the answer after comparing and analyzing the two submissions.

Statistics and Analysis. After removing the

	PB	ZX
consensus ratio (sent)	35.88	46.20
consensus ratio (token)	69.38	79.21
OOV ratio	26.68	17.91

Table 1: Analysis of the annotated data.

sentences with wrong word segmentation or incomprehensible semantics, we obtain 9,040 PB sentences and 3,249 ZX sentences. We analyze the two datasets from three aspects, as shown in Table 1. The sentence-wise consensus ratio is the percent of sentences that receive completely the same submission from two annotators, which is only 35% for PB and 46% for ZX. This means that more than a half of all sentences need to be checked by expert annotators, showing the complexity of syntactic annotation and the necessity of double annotation for quality guarantee. The token-wise consensus ratio is the percent of tokens that receive the same heads and labels from two annotators, which is still lower than 70% for PB and 80% for ZX. These consensus ratios clearly show that PB is more difficult to annotate than ZX. As user generated content, PB is much more casual and contains a lot of word ellipsis phenomena, wrongly written characters, abbreviated words, ill-grammar expressions, and so on.

The OOV (out-of-vocabulary) ratio means the percent of tokens that do not occur in the source-domain BC data of Jiang et al. (2018). We can see that the OOV ratio is much higher in PB than ZX, which would certainly make PB more difficult to parse.

3 Approaches

This section presents several semi-supervised cross-domain parsing approaches.

3.1 Base Biaffine Parser

In this work, we build all the approaches over the state-of-the-art deep biaffine parser (Dozat and Manning, 2017). As a graph-based dependency parser, it employs a deep biaffine neural network to compute the scores of all dependencies, and uses viterbi decoding to find the highest-scoring tree. Figure 2 shows how to compute the score of an arc $\texttt{score}(i \leftarrow j)$.

First, the biaffine parser applies multi-layer bidirectional sequential LSTMs (BiLSTM) to encode the input sentence. The input of the i-th word is the concatenation of word/tag embeddings, i.e., $\mathbf{x}_i = \mathbf{e}^{w_i} \oplus \mathbf{e}^{t_i}$. The output vector of the top-layer BiLSTM for the i-th word is denoted as $\mathbf{h}_i$. It is fed into two separate MLPs to get two lower-dimensional representation vectors of the word, as a head and a dependent respectively.

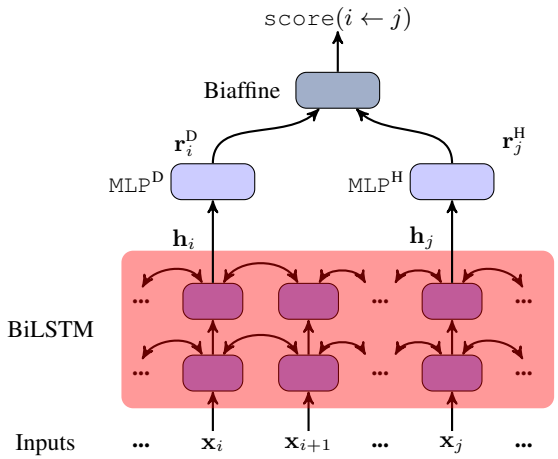

Figure 2: Computation of $\texttt{score}(i \leftarrow j)$ in the biaffine parser. For simplicity, we only draw two-layer BiLSTMs.

$$\mathbf{r}_i^{\mathrm{H}}, \mathbf{r}_i^{\mathrm{D}} = \mathrm{MLP}^{\mathrm{H}}\left(\mathbf{h}_i\right), \mathrm{MLP}^{\mathrm{D}}\left(\mathbf{h}_i\right) \quad (1)$$

Finally, the score of an arc is computed via a biaffine operation.

$$\texttt{score}(i \leftarrow j) = \begin{bmatrix} \mathbf{r}_i^{\mathrm{D}} \\ 1 \end{bmatrix}^{\mathrm{T}} \mathbf{W}^b \mathbf{r}_j^{\mathrm{H}} \quad (2)$$

Similarly, the parser uses extra MLPs and biaffines to compute label scores $\texttt{score}(i \overset{l}{\leftarrow} j)$, Due to space limitation, we refer readers to Dozat and Manning (2017) for more details.

Training loss. For each w_i and its gold-standard head w_j and label l, the parser adopts local cross-entropy losses.

$$\begin{aligned} \texttt{loss}(i \overset{l}{\leftarrow} j) = &- \log \frac{e^{\texttt{score}(i \leftarrow j)}}{\sum_{0 \le k \le n} e^{\texttt{score}(i \leftarrow k)}} \\ &- \log \frac{e^{\texttt{score}(i \overset{l}{\leftarrow} j)}}{\sum_{l' \in \mathcal{L}} e^{\texttt{score}(i \overset{l'}{\leftarrow} j)}} \end{aligned} \quad (3)$$

where $\mathcal{L}$ is the label set. Separate losses are computed for heads selection and labeling.

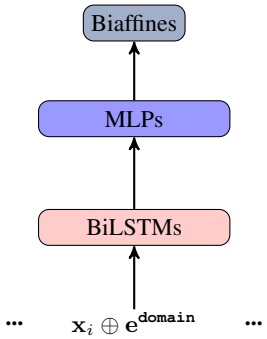

Figure 3: The framework of the DOEMB approach, where `domain` = "src" for source-domain sentences and "tgt" for target-domain ones.

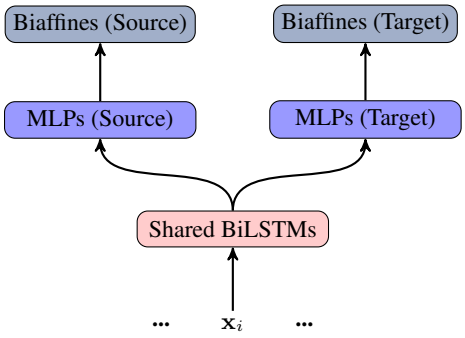

Figure 4: The framework of MTL.

3.2 Combining Two Training Datasets

In this subsection, we describe three simple approaches for combining the source- and target-domain training datasets.

(1) **Direct concatenation (CONCAT)**. The most straightforward way is to directly merge multiple training datasets into a larger one. This method treats the source- and target-domain training datasets equally. The basic parser can be directly used with little modification. The major drawback for this method is that the model uses the same parameters for both domains, and thus is unable to learn the domain-specific features.

(2) **Domain embedding (DOEMB)**. Stymne et al. (2018) propose a treebank embedding approach to improve parsing by utilizing multiple heterogeneous treebanks (following diverse annotation guideline) for a language. Inspired by their work, we propose to concatenate each word position with an extra domain embedding to indicate which domain this training sentence comes from, as illustrated in Figure 3. In this way, we expect the model can fully utilize both training datasets, since most parameters are shared except the two domain embedding vectors, and learn to distinguish the domain-specific and general features as well.

(3) **Multi-task learning (MTL)** aims to incorporate labeled data of multiple related tasks for improving performance (Collobert and Weston, 2008). Guo et al. (2016) first employ MTL to improve parsing performance by utilizing multiple heterogeneous treebanks and treating each treebank as a separate task. As shown in Figure 4, we make a straightforward extension to the biaffine parser to realize multi-task learning. The source-domain and target-domain parsing are treated as two individual tasks with shared parameters for word/tag embeddings and BiLSTMs. The main weakness of MTL is that the model cannot make full use of the source-domain labeled data, since the source-domain training data only contributes to the training of the shared parameters.

The corpus weighting strategy. For all above three approaches, the target-domain labeled data would be overwhelmed by the source-domain data during training if directly combined, since there usually exists a very big gap in their scale. Therefore, we employ the simple corpus weighting strategy (Li et al., 2014) as a useful trick. Before each iteration, we randomly sample training sentences separately from the target- and source-domain training data in the proportion of $1 : M$. Then we merge and randomly shuffle the sampled data for one-iteration training. We treat $M \geq 1$ as a hyper-parameter tuned on the dev data.

3.3 Utilizing Unlabeled Data

Besides labeled data, how to exploit unlabeled data, both target- and source-domain, has been an interesting and important direction for cross-domain parsing for a long time, as discussed in Section 5. Recently, Peters et al. (2018) introduce embeddings from language models (ELMo) to effectively utilize large amount of raw texts as a pretraining step. They use multiple BiLSTM layers as the sentence encoder and employ left-to-right sequential language model losses.

In this work, we propose a very simple two-step approach to apply ELMo to the cross-domain scenario.

Step 1: Training ELMo on a large-scale general-domain unlabeled data. We train ELMo on the Chinese Gigaword Third Edition, consisting of about 1.2 million sentences. It takes about 7 days using 6 GPU nodes (GTX 1080Ti).

Step 2: Fine-tuning ELMo on the target-domain unlabeled data. We then fine-tune ELMo on the target-domain unlabeled data using the parameters trained in the previous step as the start point. To save computation resource, we merge all train/dev/unlabeled data of all three domains as one unlabeled dataset for fine-tuning ELMo once, and use the same fine-tuned ELMo for all three domains.

For each word, the representations from the three BiLSTM layers of ELMo are averaged and used to replace the original word embeddings in the Biaffine Parser. We did not try to let the model automatically learn different weights for different layers, which may leads to slightly better performance. Since ELMo uses charLSTM to learn the first-layer word representations, we did try to expand the character dictionary with those that only occur in the target-domain unlabeled data, and randomly initialize their corresponding char embeddings before fine-tuning ELMo. However, this only produces slight and inconsistent performance gains.

4 Experiments

Data. We use the balanced corpus (BC) released by Jiang et al. (2018) as the source domain, following their train/dev/test split. We use our newly annotated PB/ZX datasets as two target domains, and split each into train/dev/test, with the consideration that the dev/test datasets are made as large as possible for the sake of more reliable evaluation. We also provide target-domain unlabeled data, as discussed in Section 2. Table 2 shows the data statistics.

Evaluation metrics. We use the standard labeled attachment score (LAS, percent of words that receives correct heads and labels) and unlabeled attachment score (UAS, ignoring labels).

Parser settings. We implement the basic biaffine parser and the proposed approaches with PyTorch. We follow the hyperparameter settings of Dozat and Manning (2017), such as learning rate and dropout ratios. Each parser is trained for at most $1,000$ 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 50 consecutive iterations.

	BC	PB	ZX
train	52,433	5,140	1,649
dev	998	1,300	500
test	1,995	2,600	1,100
unlabeled	-	326,981	32,492

Table 2: Data statistics in sentence number.

Trained on	BC		PB		ZX	
	UAS	LAS	UAS	LAS	UAS	LAS
BC	**82.77**	**77.66**	68.73	61.93	69.34	61.32
PB	62.10	55.20	**75.85**	**70.12**	51.50	41.92
ZX	56.15	48.34	52.56	43.76	**69.54**	**63.65**

Table 3: Performance on dev data of models trained on a single-domain training data.

4.1 Single-domain Training Results

Table 3 presents parsing accuracy on the dev data when training each parser on a single-domain training data. We can see that although PB-train is much smaller than BC-train, the PB-trained parser outperforms the BC-trained parser by about 8% on PB-dev, indicating the usefulness and importance of target-domain labeled data especially when two domains are very dissimilar.

However, the gap between the ZX-trained parser and the BC-trained is only about 2% in LAS, which we believe has a two-fold reason. First, the size of ZX-train is even smaller, and is only less than one third of that of PB-train. Second, the BC corpus are from the People Daily newspaper and probably contains novel articles, which are more similar to ZX. Overall, it is clear and reasonable that the parser achieves best performance on a given domain when the training data is from the same domain.

4.2 Combining Two Training Datasets

We combine the source- and target-domain training data using the three approaches described in Section 3.2. Due to the big gap between the size of the source- and target-domain training data, we employ the corpus weighting strategy to balance the effect of difference sources.

Figure 5 shows the results on the dev data with different weighting factor M. The curves on both PB and ZX clearly show that corpus weighting is extensively helpful, and the performance gap between a good weight factor and a bad one can

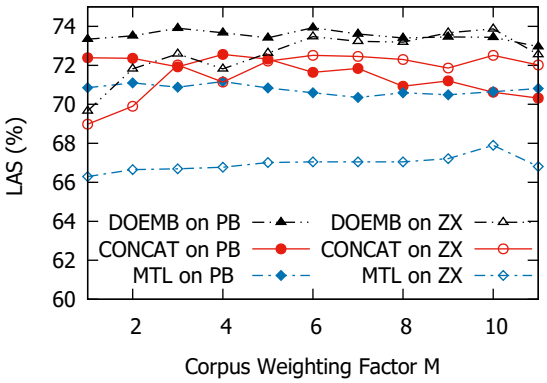

Figure 5: Effect of corpus weighting on different approaches on the dev data.

be large for certain target domains and methods. Specifically, for PB as the target domain, it seems sufficient to use the same weight for the source and target domains (i.e., $M = 1$), and choosing a proper larger M leads to less than 1% improvement. In contrast, corpus weighting is more important for the ZX domain, and leads to much better performance with a larger M. In addition to the very small size of ZX-train, another reason may be due to the large similarity between ZX and BC, as previously discussed.

From another aspect, we can see that the DOEMB approach always performs best among the three approaches on both target domains, and MTL is the most ineffective in making use of the source-domain training data.

Overall, the results are consistent with our discussions in Section 3.2. The key of the success of DOEMB over both CONCAT and MTL lies in the balance between merging the knowledge in both domains by sharing more parameters and distinguishing the two domains in order to learn domain-specific and general features.

For each method-domain pair, we select the best corpus weighting M according to their results on the dev data.

4.3 Utilization of Unlabeled Data

In this part, we enhance the most effective DOEMB approach with ELMo with the approach described in Section 3.3. Table 4 reports the results.

Surprisingly, using the ELMo trained on general-domain Chinese Gigaword corpus has opposite effect on the two target domains. LAS decreases by 0.99 on PB but increases by 1.16 on ZX. We suspect the reason may be that that

	PB		ZX	
	UAS	LAS	UAS	LAS
DOEMB	78.97	73.93	78.64	73.87
+ ELMo (Giga)	*78.49*	*72.94*	79.92	75.03
+ Fine-tuning	**83.08**	**78.37**	**81.48**	**76.51**

Table 4: Performance of the DOEMB approach enhanced with ELMo on the dev data.

Chinese gigaword corpus, like BC, contains many novel-related texts that are similar to ZX. In contrast, it is quite unlikely to have texts similar to PB, considering the PB texts are usually recent user-generated content. This finding is different from the in-domain parsing results, where ELMo is always helpful (Che et al., 2018; Clark et al., 2018)

Further fine-tuning ELMo on target-domain unlabeled data leads to consistent and large improvement on both domains. Compared with "ELMo (Giga)", LAS increases by 5.43 on PB and 1.48 on ZX. We believe the larger improvement on PB versus ZX is mainly due to the much larger scale of unlabeled PB data. The results demonstrate that through fine-tuning on target-domain unlabeled data, ELMo effectively learns domain-specific knowledge, and is able to produce more reliable contextualized word representations.

4.4 Final Results On Test Data

Table 5 shows the final results on the test data, which are consistent with the previous observations. First, when constrained on single-domain training data, using the target-domain data is the most effective. Second, using source-domain data as extra training data is helpful, and the DOEMB method performs the best. Third, it is extremely useful and efficient to first train ELMo on very large-scale general-purpose unlabeled data and then fine-tune it on relatively small-scale target-domain unlabeled data.

4.5 Analysis

The final performances on PB are consistently higher than those on ZX by about 2%, as shown in Table 5. We believe one major reason is PB-train is more than three times larger than ZX-train. This then raises an interesting and important question. When facing a new domain, how much data do we need to annotate to reach a certain performance given a certain amount of source-domain data?

	PB		ZX	
	UAS	LAS	UAS	LAS
Trained on single-domain data				
BC-train	67.55	61.01	**68.44**	59.55
PB-train	**74.52**	**69.02**	51.62	40.36
ZX-train	52.24	42.76	68.14	**61.71**
Trained on source- and target-domain data				
MTL	75.39	69.69	72.11	65.66
CONCAT	77.49	72.16	76.80	70.85
DOEMB	78.24	72.81	77.96	72.04
+ ELMo	*77.62*	*72.35*	78.50	72.49
+ Fine-tuning	**82.05**	**77.16**	**80.44**	**75.11**

Table 5: Final results on the test data.

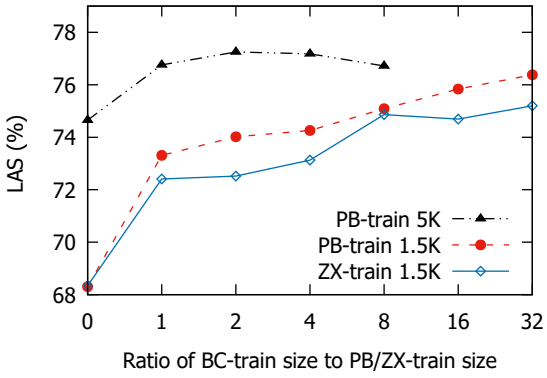

Figure 6: The effect of the *relative* size of the target-domain training data.

We try to give some clues through the following analysis.

Effect of the source-domain data size is shown in Figure 6. We fix the size of the target-domain data and increase the size of the source-domain data by using a random subset of BC-train. The "PB/ZX-train 1.5K" curves are based on random 1500 PB/ZX-train sentences in order to make fair comparison, and the "PB-train 5K" curve uses random 5000 PB-train sentences in order to understand the effect of larger target-domain data. For example, "4" at the x-axis means that the size of BC-train is four times as much as that of the target-domain data.

We can see that when the size of the target-domain data is small, i.e., "PB/ZX-train 1.5K", adding more source-domain BC-train data leads to consistent improvements. In split of the same data size, "PB-train 1.5K" and "ZX-train 1.5K" still have a large performance gap, which is probably caused by the effect of ELMo with the much larger scale of unlabeled PB data, although ZX is easier to parse as discussed in Section 2.

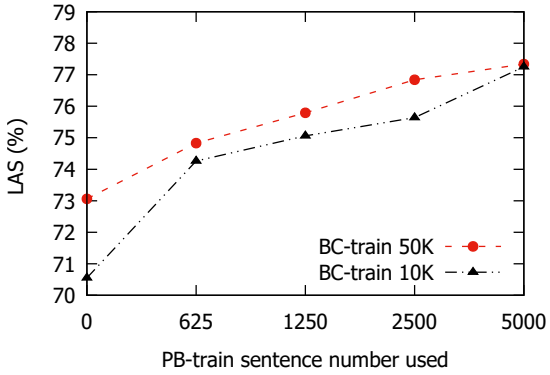

Figure 7: The effect of the size of the target-domain training data.

In contrast, for the larger "PB-train 5K", the peak LAS is obtained when 10K BC-train sentences are used, and using more BC-train data even slightly hurts performance. This shows that when the target-domain training data is large, the usefulness of the source-domain data becomes limited.

Effect of the target-domain data size is shown in Figure 7. Due to the small size of ZX-train, we only experiment with PB-train. We draw a "BC-train 10K" curve, since the previous analysis show that its combination with "PB-train 5K" already reaches peak performance.

We can see that exponentially enlarging the size of the target-domain data leads to nearly linearized improvement, indicating data annotation is the most direct and effective (or maybe necessary) way for improving cross-domain parsing performance.

On the other hand, we can see although the final performance is nearly the same for BC-train 50K and 10K, the 50K curve is obviously more steady and consistent, showing that it is usually a wise choice to use all available source-domain data.

5 Related Works

Domain adaptation has been a crucial and challenging research topic in both NLP and ML fields. Due to the vast scope of related research, we try to give a brief (and far from complete) review on some representative approaches of high relevance with syntactic parsing.

Unsupervised domain adaptation. Due to the lack of sufficient labeled data, most previous works focuses on unsupervised domain adapta-

tion, assuming there is only labeled data for the source domain. Researchers make great effort to learn useful features from large-scale unlabeled target-domain data, which is usually much easier to collect. As a typical semi-supervised approach, *self-training* is shown to be very useful for cross-domain constituent parsing (McClosky et al., 2006) and dependency parsing (Yu et al., 2015). There are also many failed works on applying self-training for in-domain and cross-domain dependency parsing.

Sagae and Tsujii (2007) apply *co-training* to the CoNLL-2007 cross-domain dependency parsing task and report positive gains (Nivre et al., 2007). In contrast, Dredze et al. (2007) experiment with many domain adaptation approaches with no success on the same datasets and suggest the major obstacle comes from the divergent annotation guideline adopted by the target-domain evaluation data.

Source-domain data selection is another interesting research direction. Given a target domain, the idea is to automatically select a most relevant subset from the source-domain training data to train the parsing model, instead of using all the labeled data (Plank and van Noord, 2011; Khan et al., 2013).

The *multi-source domain adaptation* problem assumes there are labeled datasets for multiple source domains. Given a target domain, the challenge is how to effectively combine knowledge in the source domains. McClosky et al. (2010) first raise this scenario for constituent parsing. They employ a regression model to predict cross-domain performance, and then use the values to combine parsing models independently trained on each source domain. Guo et al. (2018) employ a similar idea of *mixture of experts* under the neural MTL framework, and conduct experiments on sentiment classification and POS tagging tasks. They employ meta-training to learn to compute the point-to-set distance between a target-domain example and a source domain.

Semi-supervised domain adaptation assumes there exist some (usually very small-scale) labeled target-domain data, which can be used to directly learn the domain-specific distributions or features. Daumé III (2007) propose a simple yet effective *feature augmentation* approach that performs well on a number of sequence labeling tasks. The idea is to distinguish domain-specific and general features by making a copy of each feature for each domain plus a shared (general) pseudo domain. Finkel and Manning (2009) further propose a hierarchical Bayesian extension of this idea. As pointed by Finkel and Manning (2009), those two works can be understood as MTL under the traditional discrete-feature ML framework.

Kim et al. (2017) propose a *neural mixture of experts* approach for cross-domain intent classification and slot tagging. Different from the unsupervised method of Guo et al. (2018), they use a small amount of target-domain labeled data to train an attention module for the computation of example-to-domain distances.

In the parsing community, Flannery and Mori (2015) propose to annotate *partially labeled* target-domain data with active learning for cross-domain Japanese dependency parsing. Similarly, Joshi et al. (2018) annotate a few dozen partially labeled target-domain sentences with a few brackets for cross-domain constituent parsing. Both results report large improvement and show the usefulness of even small amount of target-domain annotation, showing the great potential of semi-supervised domain adaptation for parsing.

6 Conclusions

This work addresses the task of semi-supervised domain adaptation for Chinese dependency parsing, based on our two newly-annotated large-scale domain-aware data, i.e., PB and ZX. We propose a simple domain embedding approach with corpus weighting to effectively combine both the source- and target-domain training data. To utilize unlabeled target-domain data, We further propose an effective two-stage approach based on the recently proposed contextualized word representations (ELMo). Our proposed semi-supervised domain adaptation approach leads to absolute LAS improvement of 16.15% (77.16 vs. 61.01) and 15.56% (75.11 vs. 59.55) on PB/ZX-test respectively, over the non-adapted parser trained on the source BC-train.

Moreover, detailed analysis shows that enlarging the target-domain labeled data is most effective in boost cross-domain parsing performance. Meanwhile, more source-domain labeled data usually leads to higher and more consistent improvement, especially when the scale of the target-domain training data is small.

Acknowledgments

The authors would like to thank the anonymous reviewers for the helpful comments. We are greatly grateful to all participants in data annotation for their hard work. This work was supported by National Natural Science Foundation of China (Grant No. 61876116, 61525205, 61572338) and was also partially supported by the joint research project of Alibaba and Soochow University.

References

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.

Wanxiang Che, Yijia Liu, Yuxuan Wang, Bo Zheng, and Ting Liu. 2018. Towards better UD parsing: Deep contextualized word embeddings. In *Proceedings of CoNLL Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 2227–2237.

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

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.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*.

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*.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependecy parsing. In *Proceedings of ICLR*.

Mark Dredze, John Blitzer, Partha Pratim Talukdar, Kuzman Ganchev, João Graca, and Fernando Pereira. 2007. Frustratingly hard domain adaptation for dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 1051–1055.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A. Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of ACL*, pages 334–343.

Jenny Rose Finkel and Christopher D. Manning. 2009. Hierarchical bayesian domain adaptation. In *Proceedings of NAACL*, pages 602–610.

Daniel Flannery and Shinsuke Mori. 2015. Combining active learning and partial annotation for domain adaptation of a japanese dependency parser. In *Proceedings of the 14th International Conference on Parsing Technologies*, pages 11–19.

Jennifer Foster, Ozlem Cetinoglu, Joachim Wagner, Joseph Le Roux, Joakim Nivre, Deirdre Hogan, and Josef van Genabith. 2011. From news to comment: Resources and benchmarks for parsing the language of web 2.0. In *Proceedings of IJCNLP*, pages 893–901.

Jiang Guo, Wanxiang Che, Haifeng Wang, and Ting Liu. 2016. A universal framework for inductive transfer parsing across multi-typed treebanks. In *Proceedings of COLING*, pages 12–22.

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

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of CoNLL*.

Xinzhou Jiang, Zhenghua Li, Bo Zhang, Min Zhang, Sheng Li, and Luo Si. 2018. Supervised treebank conversion: Data and approaches. In *Proceedings of ACL*, pages 2706–2716.

Vidur Joshi, Matthew Peters, and Mark Hopkins. 2018. Extending a parser to distant domains using a few dozen partially annotated examples. In *Proceedings of ACL*, pages 1190–1199.

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

Young-Bum Kim, Karl Stratos, and Dongchan Kim. 2017. Domain attention with an ensemble of experts. In *Proceedings of ACL*, pages 643–653.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *CoRR*, abs/1603.04351.

Nikita Kitaev and Dan Klein. 2018. Constituency parsing with a self-attentive encoder. In *Proceedings of ACL*, pages 2675–2685.

Lingpeng Kong, Nathan Schneider, Swabha Swayamdipta, Archna Bhatia, Chris Dyer, and Noah A. Smith. 2014. A dependency parser for tweets. In *Proceedings of EMNLP*, pages 1001–1012.

Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. *Dependency Parsing (Synthesis Lectures On Human Language Technologies)*. Morgan and Claypool Publishers.

Zhenghua Li, Min Zhang, and Wenliang Chen. 2014. Ambiguity-aware ensemble training for semi-supervised dependency parsing. In *Proceedings of ACL*, pages 457–467.

Xuezhe Ma, Zecong Hu, Jingzhou Liu, Nanyun Peng, Graham Neubig, and Eduard H. Hovy. 2018. Stack-pointer networks for dependency parsing. In *Proceedings of ACL*, pages 1403–1414.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Reranking and self-training for parser adaptation. In *Proceedings of ACL*, pages 337–344.

David McClosky, Eugene Charniak, and Mark Johnson. 2010. Automatic domain adaptation for parsing. In *Proceedings of NAACL-HLT*, pages 28–36.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of ACL*, pages 92–97.

Joakim Nivre, Johan Hall, Sandra Kübler, Ryan McDonald, Jens Nilsson, Sebastian Riedel, and Deniz Yuret. 2007. The coNLL 2007 shared task on dependency parsing. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL 2007*, pages 915–932.

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.

Slav Petrov and Ryan McDonald. 2012. Overview of the 2012 shared task on parsing the web. In *Notes of the First Workshop on Syntactic Analysis of Non-Canonical Language (SANCL)*.

Barbara Plank and Gertjan van Noord. 2011. Effective measures of domain similarity for parsing. In *Proceedings of ACL*, pages 1566–1576.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. *OpenAI Technical Report*.

Kenji Sagae and Jun'ichi Tsujii. 2007. Dependency parsing and domain adaptation with LR models and parser ensembles. In *Proceedings of the CoNLL Shared Task Session of EMNLP-CoNLL*, pages 1044–1050.

William Yang Wang, Lingpeng Kong, Kathryn Mazaitis, and William W Cohen. 2014. Dependency parsing for weibo: An efficient probabilistic logic programming approach. In *Proceedings of EMNLP*, pages 1152–1158.

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*, pages 1–10.

Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2014. Type-supervised domain adaptation for joint segmentation and pos-tagging. In *Proceedings of EACL*, pages 588–597.

Hao Zhou, Yue Zhang, Shujian Huang, and Jiajun Chen. 2015. A neural probabilistic structured-prediction model for transition-based dependency parsing. In *Proceedings of ACL*, pages 1213–1222.