A Fine-Grained Domain Adaption Model for Joint Word Segmentation and POS Tagging

Peijie Jiang[1] **Dingkun Long** **Yueheng Sun**[2] **Meishan Zhang**[1*]
Guangwei Xu **Pengjun Xie**
[1]School of New Media and Communication, Tianjin University, China
[2]College of Intelligence and Computing, Tianjin University, China
{jzx555,yhs,zhangmeishan}@tju.edu.cn
{longdingkun1993,ahxgwOnePiece,xpjandy}@gmail.com

Abstract

Domain adaption for word segmentation and POS tagging is a challenging problem for Chinese lexical processing. Self-training is one promising solution for it, which struggles to construct a set of high-quality pseudo training instances for the target domain. Previous work usually assumes a universal source-to-target adaption to collect such pseudo corpus, ignoring the different gaps from the target sentences to the source domain. In this work, we start from joint word segmentation and POS tagging, presenting a fine-grained domain adaption method to model the gaps accurately. We measure the gaps by one simple and intuitive metric, and adopt it to develop a pseudo target domain corpus based on fine-grained subdomains incrementally. A novel domain-mixed representation learning model is proposed accordingly to encode the multiple subdomains effectively. The whole process is performed progressively for both corpus construction and model training. Experimental results on a benchmark dataset show that our method can gain significant improvements over a vary of baselines. Extensive analyses are performed to show the advantages of our final domain adaption model as well.

1 Introduction

Chinese Word Segmentation (CWS) and Part-Of-Speech (POS) tagging are two fundamental tasks for natural language processing (NLP) in Chinese (Emerson, 2005; Jin and Chen, 2008), serving as backbones for a number of downstream NLP tasks. The joint models of the two tasks can lead to better performance because they are closely-related and the pipeline models suffer from the error propagation problem (Ng and Low, 2004; Zhang and Clark, 2008; Wang et al., 2011; Zeng et al., 2013; Zhang et al., 2018; Tian et al., 2020a), which can be alleviated in the joint architecture.

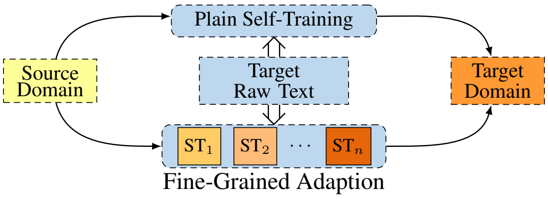


Figure 1: The idea of fine-grained domain adaption.

Currently, joint CWS and POS tagging has gained great achievements with BERT inputs (Tian et al., 2020a,b). Our preliminary results show that the F1-score of joint POS tagging can be close to 95% when the training and test corpus both belong to a standard newswire domain. Unfortunately, it is not always the case in real applications. The performance might be degraded dramatically when the source and target domains are highly different. Taken the ZhuXian (a novel from Internet) as an example (Zhang et al., 2014), the same model can only obtain an F1-score of 89% for POS tagging according to our results.

It is a typical domain adaption problem targeted to joint CWS and POS tagging. Self-training could be one promising solution (Inoue et al., 2018; Zou et al., 2019; Saito et al., 2020) which can accomplish the goal in a fully-automatic manner without any human intervention (Liu and Zhang, 2012). By using a source model to automatically label a large-scale raw corpus of the target domain, and then selecting a set of high-confidence pseudo-labeled instances as additional training data, we can obtain boosted performance on the target domain. The quality of pseudo corpus is the key to success. For the target sentences which are far from the source domain, the generated corpus based on them might be of extremely-low quality (Shu et al., 2018; Zhao et al., 2019). Thus, these sentences should be either filtered, resulting in a biased corpus to the target domain, or be kept with great noises to degrade the overall target performance.

In this work, we suggest a fine-grained domain

[*] Corresponding author.

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3587–3598
November 7–11, 2021. ©2021 Association for Computational Linguistics

adaption method to alleviate the above problem of self-training. We define a simple and intuitive metric to measure the distance (gap) of a target sentence to the source domain. Based on the metric, we create a set of high-quality training corpora incrementally according to the distances of the target sentences to the source domain. Figure 1 shows the main idea. The process is conducted by several iterations in a progressive manner, where at each new iteration, we add a small set of high-quality instances which are not as distant from the previous iteration. Finally, we arrive at a training corpus covering the target domain of various distances fully. At each iteration, we go only a little further by the distance, thus the quality of the pseudo corpus can be greatly ensured by the previous model.

By the fine-grained domain adaption, we can obtain a training corpus of multiple types from different iterations, where each type differs from the other in both quality and input distribution. During the early iterations, the produced instances are possible with higher quality and close to the source domain, while for the later iterations, the quality might be lower and the distance to the source domain is larger. To make full use of the corpus together with the source training set, we present a domain-mixed model for sophisticated representation learning to capture domain-aware and domain-invariant features (Daumé III, 2007; Ganin et al., 2016; Tzeng et al., 2017), which is also strengthened progressively by the incremental style of the fine-grained domain adaption.

We conduct experiments on a benchmark ZhuXian dataset (Zhang et al., 2014) to show the effectiveness of our method. In detail, the Penn Chinese Treebank version (Xue et al., 2005) 6.0 (CTB6) is used as the source corpus, belonging to the newswire domain, while the target ZhuXian corpus is from an Internet novel. Experimental results show that our fine-grained domain adaption is significantly better than previous self-training studies. Moreover, we find that our domain-mixed representation learning model suits the fine-grained framework perfectly. We also conduct extensive analyses to understand our model comprehensively. We will release our codes at github.com/JZX555/FGDA under Apache License 2.0 to help the reproduction.

2 Joint CWS and POS Tagging

This section describes the basic model of our joint CWS and POS tagging. Concretely, we regard our joint task as a character-level sequence labeling problem following Tian et al. (2020a). Given an input character sequence $X = [x_1, ..., x_n]$, the output labels $Y = [y_1, ..., y_n]$ are concatenations of word boundaries (i.e., BMES) and POS tags for all sentential characters. We exploit an ADBERT-BiLSTM-CRF model as our basic model, which is very strong in performance and highly parameter efficient. The model includes two parts sequentially: (1) ADBERT for character representation, (2) BiLSTM-CRF for feature extraction, label inference and training. Below, we introduce the ADBERT directly and the BiLSTM-CRF is exactly the same as Tian et al. (2020a) which can be referred to in their work for the details.

Adapter ∘ BERT We exploit BERT (Devlin et al., 2019) to derive character representations for a given sentence $X = [x_1, ..., x_n]$, as it brings state-of-the-art performances for a range of Chinese language processing tasks. In particular, we patch BERT with adapters (Houlsby et al., 2019) inside all the included transformer units. By this way, fine-tuning BERT parameters is no longer necessary across different tasks, and we only need to tune the adapter parameters. More particularly, we let all adapters across different transformer units use a shared set of parameters to reduce the scale of tunable model parameters of our joint task. Here we refer to this method as ADBERT:

$$\boldsymbol{e}_1, ..., \boldsymbol{e}_n = \text{ADBERT}(X = [x_1, ..., x_n]), \quad (1)$$

where the detailed network of transformer with adapters is illustrated in our Appendix A.

3 Our Method

The above joint CWS and POS tagging model can perform well on the standard setting when the test domain is similar to the training domain (Tian et al., 2020a,b). However, the performance might be degraded dramatically when the test (i.e., target) domain differs from the training (i.e., source) domain significantly. There have two studies for cross-domain of joint CWS and POS tagging (Liu and Zhang, 2012; Zhang et al., 2014), both of which have exploited self-training due to its effectiveness as well as simplicity for domain adaption. The self-training aims to produce a set of high-confidence training instances of the target domain which are used to train a target model. Here we follow this line of work, presenting a novel fine-grained domain adaption strategy.

The fine-grained domain adaption is an extension of the standard self-training, aiming to produce a helpful pseudo training corpus of the target domain. The line of work is essentially orthogonal to the representation learning methods which aim to learn sophisticated (e.g., domain-aware and domain-invariant) features for domain adaption. Thus, we also present a novel domain-mixed model based on the basic ADBERT-BiLSTM-CRF for effective exploration of our fine-grained domain adaption. In the following, we first describe the fine-grained domain adaption method in detail, and then introduce our representation learning model.

3.1 Fine-Grained Domain Adaption

The overall flow of self-training includes three steps: (1) first, we train an initial model by the source corpus; (2) second, we apply the source model onto a large-scale raw corpus, obtaining auto-labeled pseudo instances of the target domain; (3) finally, we select a set of high-confidence instances from the pseudo corpus which would be added to train the target model. The flow can be conducted repeatedly by several iterations, where the model in step 1 is trained by the progressively added step-3 instances. However, according to our preliminary results, the plain iterative self-training can only achieve very marginal improvement.

The reason may lie in that the above process is difficult to ensure the quality of the selected instances, especially when the input target sentences are very distant from the source domain (Sohn et al., 2020). The step-1 models do not perform well on these sentences without any specialization. If these sentences are excluded because of their low quality, the final target model would be trained on a biased corpus, while these sentences are added into the target training corpus, great noises are introduced which would degrade the overall performance. Aiming for the problem, we propose a fine-grained domain adaption strategy to alleviate the influence of the large gaps during the automatic corpus construction.

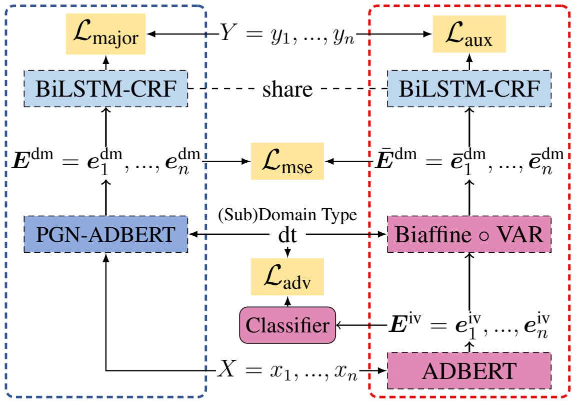
Concretely, we guide the iterative self-training by a specific explicit distance metric. At each iteration, we add a set of high-confidence pseudo instances whose distances are only a little larger than the previous iteration. The sentences during each selection can be regarded as from a special fine-grained subdomain of the target domain. By this way, the target model is gradually adapted to the distant sentences far away from the source domain, producing a higher-quality corpus of various distances. Compared with the direct source-to-target adaption, we adopt the OOV (i.e., the newly-generated words which are out of the training vocabulary) number as the distance measurement, which is highly simple and intuitive. We construct a set of high-quality automatic corpora by choosing from the zero/one-number-OOV target sentences to the large-number-OOV target sentences progressively.

Algorithm 1: Fine-Grained Adaption

```
   Data: Source domain training dataset $S$
         Target domain raw corpus $D_1$
   Output: Latest model $M$
1  Initial training dataset $\mathrm{T}_1 = S$
2  for i = 1, 2, 3... until converge do
3      Model training: $M_i = \mathrm{Train}(\mathrm{T}_i)$
4      Data auto-labeling: $\hat{D}_i = M_i(D_i)$
5      Lexicon: $L_{\text{tgt}} = L_{\text{tgt}} \cup L_{\text{top-K}}(\hat{D}_i)$
6      Progress $i$th auto instances: $\mathrm{ST}_i = \{\}$
7      foreach instance $(\hat{X}, \hat{Y})$ in $\hat{D}_i$ do
8          $\mathrm{C}_{\text{oov}}$: $\mathrm{num}_{\text{OOV}} \leq i$
9          $\mathrm{C}_{\text{lex}}$: all oov in $L_{\text{tgt}}$
10         $\mathrm{C}_{\text{conf}}$: $p(\hat{Y}|\hat{X}) \geq p_{\text{threshold}}$
11         if $C_{oov}$ && $C_{lex}$ && $C_{conf}$ then
12             $\mathrm{ST}_i = \mathrm{ST}_i + \{(\hat{X}, \hat{Y})\}$
13         end
14     end
15     $\mathrm{T}_{i+1} = \mathrm{T}_i + \mathrm{ST}_i$ ;
16     $D_{i+1} = D_i \setminus \mathrm{ST}_i.X$ ;
17 end
```

Algorithm 1 shows the pseudo codes of fine-grained domain adaption. Initially, we set the first-iteration training dataset by the source corpus S, and then execute the pseudo codes of lines 3-16 repeatedly. First, we train a model M_i by current-iteration training dataset T_i, and apply the model to the remaining raw corpus of the target domain, resulting in auto-labeled corpus $\hat{D}_i$, as shown by the codes at lines 3-4. Next, we conduct a lexicon building process at line 5 which would be used for quality assurance. At each iteration, we collect a set of top-K confident word-POS pairs $L_{\text{top-K}}$ by their weighted frequencies in $\hat{D}_i$,[1] which are added to the target lexicon L_{tgt}. Then, the key arrives

[1] The frequency is discounted by the sentence-level probability of the best output.


$\mathcal{L}_{\text{major}}$
$Y = y_1, ..., y_n$
$\mathcal{L}_{\text{aux}}$
BiLSTM-CRF
share
BiLSTM-CRF
$\boldsymbol{E}^{\text{dm}} = \boldsymbol{e}_1^{\text{dm}}, ..., \boldsymbol{e}_n^{\text{dm}}$
$\mathcal{L}_{\text{mse}}$
$\bar{\boldsymbol{E}}^{\text{dm}} = \bar{\boldsymbol{e}}_1^{\text{dm}}, ..., \bar{\boldsymbol{e}}_n^{\text{dm}}$
(Sub)Domain Type
dt
PGN-ADBERT
Biaffine ∘ VAR
$\mathcal{L}_{\text{adv}}$
Classifier
$\boldsymbol{E}^{\text{iv}} = \boldsymbol{e}_1^{\text{iv}}, ..., \boldsymbol{e}_n^{\text{iv}}$
$X = x_1, ..., x_n$
ADBERT

(a) (Sub)Domain-Aware (b) (Sub)Domain-Invariant

Figure 2: The structure of the domain-mixed model, where the four objectives are defined in Equation 4.

at lines 6-15 for new training dataset selection to obtain ST_i, which advances the training corpus to T_{i+1}. We traverse all instances in $\hat{D}_i$, and add the instances which satisfy C_{oov}, C_{lex} and C_{conf} together, where C_{oov} indicates the OOV number to control the distance to the source domain, and C_{lex} and C_{conf} ensure the instance quality. Finally, at line 16, we remove the selected instances from the target domain corpus and start the next iteration.

3.2 Our Domain-Mixed Model

By fine-grained domain adaptation, we can obtain a training corpus of multiple types (i.e., S, $\text{ST}_1, \cdots, \text{ST}_\text{n}$ (n denotes the last iteration) in Algorithm 1) where each type corresponds to a domain (i.e., S) or subdomain (i.e., ST_n). Thus, the exploration of the training corpus can be regarded as multi-source domain adaption (Zhang et al., 2015; Sun et al., 2015). To better explore the corpus, we propose a novel domain-mixed model to fully benefit from the fine-grained domain adaptation.

Our domain-mixed model follows a standard representation learning framework of domain adaption, which attempts to capture effective domain-aware and domain-invariant features. Figure 2 shows the overall architecture of the model, where two individual ADBERT-BiLSTM-CRF components are included, which are used for domain-aware and domain-invariant feature learning, respectively. The feature learning modules are both adapted at the ADBERT, and a shared BiLSTM-CRF is exploited across the two components. In the below, we introduce the (sub)domain-aware and (sub)domain-invariant components, respectively, and then describe the overall inference and training.

The (Sub)Domain-Aware Component A major problem of our basic ADBERT-BiLSTM-CRF model is that it treats all (sub)domain types of our final training corpus equally. Here we take the (sub)domain types as inputs along with the sentences deriving domain-aware features. Concretely, we follow Jia et al. (2019) and Üstün et al. (2020), exploiting Parameter Generator Network (PGN) on the adapter layers to achieve our goal, which generates (sub)domain-aware parameters for the adapters inside the ADBERT.

We pack all parameters of the adapter layers into a single vector $\boldsymbol{V}$ by reshaping and concatenation, which can be reverse unpacked perfectly for adapter calculation. As shown in Figure 2(a), we refer to ADBERT with PGN as PGN-ADBERT. Taken the input sentence and (sub)domain type pair by (X, dt), and the overall calculation of the (sub)domain-aware character representations is formalized as follow:

$$\begin{aligned} \boldsymbol{e}_1^{\text{dm}}, ..., \boldsymbol{e}_n^{\text{dm}} &= \text{PGN-ADBERT}(X, \text{dt}) \\ &= \text{ADBERT}(X, \boldsymbol{V} = \boldsymbol{\Theta}\boldsymbol{e}^{\text{dt}}), \end{aligned} \tag{2}$$

where $\boldsymbol{\Theta}$ is a learnable parameter in this component, $\boldsymbol{e}^{\text{dt}}$ is the (sub)domain type embedding, and PGN-ADBERT is a special case of ADBERT with specified module parameters $\boldsymbol{V}$. The resulted representations are then fed into BiLSTM-CRF for our joint task.

The (Sub)Domain-Invariant Component The domain-invariant features have been extensively investigated because of their generalization capability across different domains (Daumé III, 2007). Here we present a (sub)domain-invariant component to learn these general features across our source domain and fine-grained target subdomains, parallel to the (sub)domain-aware component. Figure 2(b) shows the architecture of this part. Firstly, the character inputs X go through ADBERT, deriving the domain-invariant features $\boldsymbol{e}_1^{\text{iv}}, ..., \boldsymbol{e}_n^{\text{iv}}$, and then we reconstruct the domain-aware features $\bar{\boldsymbol{e}}_1^{\text{dm}}, ..., \bar{\boldsymbol{e}}_n^{\text{dm}}$ by specifying the input (sub)domain type dt, which are then fed into BiLSTM-CRF for our joint task following our basic model.

The domain-invariant features $\boldsymbol{e}_1^{\text{iv}}, ..., \boldsymbol{e}_n^{\text{iv}}$, are learned in an adversarial manner (Ganin and Lempitsky, 2015; Ganin et al., 2016) for sentence-level (sub)domain type classification. We derive sentence-level representation $\boldsymbol{v}$ by averaged pooling over these features, and then determine the (sub)domain type of the input sentence by a simple linear classifier. Note that we will intentionally cheat the classifier to make the $\boldsymbol{v}$ domain irrelevant, aiming to obtain good domain-invariant features.

In natural, the domain-invariant component tries to reconstruct and approximate the domain-aware component since they share the same decoding part. We unite the domain-invariant features $\boldsymbol{e}_1^{\text{iv}}, ..., \boldsymbol{e}_n^{\text{iv}}$ and the (sub)domain type dt to reconstruct the domain-aware features, which are then used for our joint task. The advantages of this manner are that we can maximize the capacity of the domain-invariant features and further enhance the interaction between the domain-aware and domain-invariant features.

Concretely, the reconstruction is implemented by a variational module with reparameterization (Kingma and Welling, 2014). Given the (sub)domain type dt and the character representation $\boldsymbol{e}_i^{\text{iv}}(i \in [1, n])$, the domain-aware representation can be calculated by:

$$\begin{aligned} \boldsymbol{\mu}_i &= \text{BiAffine}_{\text{mean}}(\boldsymbol{e}_i^{\text{iv}}, \boldsymbol{e}^{\text{dt}}), \\ \log(\boldsymbol{\sigma}_i^2) &= \text{BiAffine}_{\text{var}}(\boldsymbol{e}_i^{\text{iv}}, \boldsymbol{e}^{\text{dt}}), \\ \bar{\boldsymbol{e}}_i^{\text{dm}} &\sim \mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i^2), \end{aligned} \tag{3}$$

where we use BiAffine operations to generate a Gaussian distribution and then sample the domain-aware features $\bar{\boldsymbol{e}}_i^{\text{dm}}$ from the distribution.

3.3 Inference and Training

We regard the (sub)domain-aware component as our major component, which outputs the final joint CWS and POS tagging results. The (sub)domain-invariant component is an auxiliary component to help the learning of the major one. Intuitively, through an alignment between the major and auxiliary components, the learned features of our major component can be naturally decomposed into domain-aware and domain-invariant features.

Inference For inference, we use the (sub)domain types of S and ST_{n} (i.e., the last fine-grained subdomain type) to perform decoding of the source and target domains, respectively.

Training We exploit four optimization objectives for training, as shown in Figure 2:

$$\begin{aligned} \mathcal{L}_{\text{major}}(X, Y, \text{dt}) &= -\log p_{\text{major}}(Y|X, \text{dt}), \\ \mathcal{L}_{\text{aux}}(X, Y, \text{dt}) &= -\log p_{\text{aux}}(Y|X, \text{dt}), \\ \mathcal{L}_{\text{adv}}(X, \text{dt}) &= \log p_{\text{adv}}(\text{dt}|X), \\ \mathcal{L}_{\text{mse}}(X) &= \parallel \boldsymbol{E}^{\text{dm}} - \bar{\boldsymbol{E}}^{\text{dm}} \parallel^2, \end{aligned} \tag{4}$$

where the first two are the losses of the two components of joint CWS and POS tagging, the third one is referred to as the adversarial loss to deceive the (sub)domain type classification, and the last is to minimize the distance of the domain-aware features between our two components leading to highly-resembled (aligned) character representations from variational reconstruction. Further, we sum the four objectives together:

$$\begin{aligned} \mathcal{L} = &\mathcal{L}_{\text{major}}(X, Y, \text{dt}) + \mathcal{L}_{\text{aux}}(X, Y, \text{dt}) \\ &+ \lambda_1 \mathcal{L}_{\text{adv}}(X, \text{dt}) + \lambda_2 \mathcal{L}_{\text{mse}}(X), \end{aligned} \tag{5}$$

resulting in the final objective of our domain-mixed model, where λ_1 and λ_2 are two hyperparameters.

Data Set		#sents	#words	#chars
CTB6	Train	23,401	641,372	1,055,586
	Devel	2,078	59,929	100,276
	Test	2,795	81,579	134,149
ZX	Test	1,394	34,355	48,075
	Raw	32,023	N/A	1,417,418

Table 1: Data statistics of CTB6 and ZhuXian.

4 Experiment

4.1 Datasets

We use the CTB6 dataset as the source domain (newswire), splitting the dataset into training, development and test sections following Tian et al. (2020a). To verify the effectiveness of our proposed domain adaption method, we exploit the ZhuXian dataset (Zhang et al., 2014) as the target domain, which belongs to a novel from Internet and is the only-one benchmark dataset for domain adaption of joint CWS and POS tagging. We strictly follow unsupervised domain adaptation where there is only a test corpus of the target domain. Table 1 shows the data statistics, where the detailed sentence, word as well as character numbers are reported. For the Zhuxian dataset, we use only the raw text and test corpus, which is available from Zhang et al. (2014).

4.2 Setting

Evaluation We adopt the standard word-level matching method to evaluate the performance of CWS and POS tagging. In particular, the joint strategy is used for POS tagging evaluation, considering word boundaries as well as POS tags as a whole. We calculate precision (P), recall (R) values, and use their F1-score as the major evaluation metric.

Model	CTB6						ZhuXian					
	CWS			POS			CWS			POS		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
(1) Baseline												
Vanilla	**97.29**	**96.85**	**97.07**	**94.73**	**94.30**	**94.51**	**94.12**	**93.61**	**93.87**	**89.19**	**88.70**	**88.94**
(2) Self-Training												
Vanilla	97.18	96.76	96.97	94.31	93.91	94.11	94.23	93.94	94.08	89.24	88.96	89.10
+Iterative	97.17	96.85	97.01	94.44	94.13	94.29	94.30	93.89	94.10	89.36	88.96	89.16
+Domain-PGN	97.21	96.84	97.03	94.40	94.04	94.22	94.25	94.03	94.14	89.27	89.06	89.17
+Domain-Mixed	**97.29**	**96.90**	**97.09**	**94.55**	**94.17**	**94.36**	**94.45**	**94.12**	**94.28**	**89.61**	**89.29**	**89.45**
(3) Fine-Grained Domain Adaption												
Vanilla	97.17	96.9	97.03	94.51	94.24	94.37	94.44	94.86	94.65	89.67	90.07	89.87
+Domain-PGN	97.33	97.04	97.19	94.57	94.29	94.43	94.74	94.71	94.72	90.07	90.04	90.06
+Domain-Mixed	**97.44**	**97.18**	**97.31**	**94.83**	**94.58**	**94.71**	**94.99**	**95.14**	**95.07**	**90.51**	**90.65**	**90.58**

Table 2: Main results, where the instance selection of self-training is simply implemented by ranking the auto-labeled sentences according to their output probabilities during the decoding, the Vanilla model refers to as the ADBERT-BiLSTM-CRF model, Iterative indicates the vanilla model with iterative self-training, and Domain-PGN indicates the model with only the (sub)domain-aware way.

Considering that there is no development corpus available for the target domain in a real scenario, we use the CTB6 development set to select the best-performing models.

Hyperparameters All hyperparameters are set empirically according to the previous studies as well as our preliminary findings (Tian et al., 2020a,b). Most importantly, our fine-grained domain adaption consumes 12 iterations to reach the peak, and the values for all other hyperparameters are described in our Appendix B.

4.3 Main Results

Table 2 shows the main results on the test datasets of both CTB6 and ZhuXian. The CTB6 results are reported to show whether the domain-adapted models could handle the source domain as well. First, we examine the F1 values of the baseline performances. Our vanilla (i.e., ADBERT-BiLSTM-CRF) model can obtain comparable performances on both CWS and POS tagging with state-of-the-art models such as Tian et al. (2020a) [2]. We can see that the model performances can drop significantly on the ZhuXian domain, resulting in decreases of $97.07 - 93.87 = 3.20$ and $94.51 - 88.94 = 5.57$ for CWS and POS tagging, respectively. The observation indicates that domain adaption is very important for our joint task.

Next, we compare fine-grained domain adaption with various self-training. Based on the vanilla model, the self-training obtains very small performance gains (including iterative self-training), i.e., only close to 0.2% which is very insignificant. The result is inconsistent with Zhang et al. (2014) which shows large improvements by simple self-training. The main reason might be due to the strong baseline with the BERT representations.

With fine-grained domain adaption, we can generate a higher quality pseudo corpus. Therefore, the gains by the vanilla model are very significant over the baseline,[3] where the improvements are 0.78 and 0.93 for CWS and POS tagging, respectively, significantly better than the vanilla self-training systems due to the quality differences of pseudo corpora. By using the final domain-mixed model, our fine-grained domain adaption can be improved further, leading to another improvement of 0.42 and 0.71 for CWS and POS tagging. The observations indicate that our method is highly effective for domain adaption of joint CWS and POS tagging.

We can see that our domain-mixed model can help the normal self-training as well, showing the effectiveness of the representation learning for domain adaption. We also compare our proposed domain-mixed model with the major component alone (Domain-PGN for short), where the latter has been demonstrated to be effective in a different scenario (Jia et al., 2019). According to the results, the Domain-PGN gives slightly better performances on CWS and POS tagging for both self-training and fine-grained domain adaption compared with the counterpart baseline. Our final domain-mixed

[2] Tian et al. (2020a) report F-score of 97.39 and 94.99 for CWS and POS tagging, respectively, by using various external information.

[3] We regard one model as significantly different from another if the p-value is below 10^{-4} by pair-wise t-test.

Model	CTB6		ZhuXian		Trainable
	CWS	POS	CWS	POS	Params Size
Finetuning	97.24	94.74	93.91	88.95	120M
Adapter	97.36	94.81	93.81	88.98	35M
Adapter (shared)	97.07	94.51	93.87	88.94	14M

Table 3: Comparisons between BERT fine-tuning and ADBERT.

Model	P	R	F1	Δ_{F1}
Final	90.51	90.65	90.58	–
$-\text{C}_{\text{oov}}$	90.20	90.16	90.18	-0.40
$-\text{C}_{\text{lex}}$	90.39	90.25	90.32	-0.26
$-\text{C}_{\text{conf}}$	90.28	90.38	90.33	-0.25
$-\text{C}_{\text{oov}} - \text{C}_{\text{lex}}$	90.02	89.99	90.00	-0.58
Self-Training	89.61	89.29	89.45	-1.13

Table 4: Ablation study of the instance selection strategies of our final model (F1 values of POS are reported).

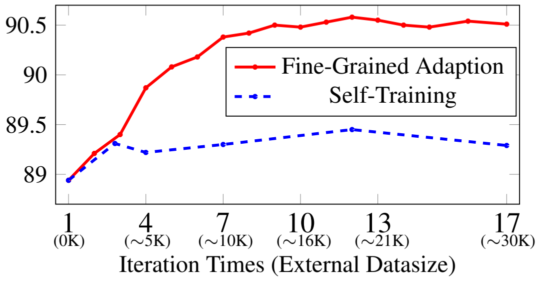


Figure 3: The POS tagging performance with respect to the number of the pseudo training instances.

model is much better, leading to significant performance increases on both tasks especially in fine-grained domain adaption.

Interestingly, we find that our final model is capable of bringing better performances on the source CTB6 test dataset as well, unlike the observations as shown in the self-training models which can hurt the source performance to a certain extent. The finding indicates that our final model is with strong practical values, since it enables one model to perform well on multiple domains.

4.4 Analysis

In this subsection, we conduct detailed experimental analyses for a comprehensive understanding of our method in-depth.

The Exploration of BERT Our work exploits ADBERT instead of the standard exploration of BERT finetuning. Here we examine the differences between them considering both performance and the size of trainable model parameters. Since ADBERT freezes all parameters of BERT, the number of trainable model parameters would be reduced greatly. Table 3 shows the comparison results, where Finetuning indicates the standard BERT-CRF model with BERT parameters tunable, Adapter denotes the ADBERT model that all adapters own separate parameters, and Adapter (shared) indicates our final ADBERT that all adapters across different transformer layers share the same parameters. As shown, we can see that our final choice can achieve comparable performance to the others with much fewer number of trainable parameters, thus our final ADBERT is highly parameter efficient.

The Instance Selection Strategy As mentioned in Algorithm 1, we include three conditions for instance selection at each iteration: C_{oov}, C_{lex} and C_{conf}. Here we conduct ablation experiments to check the necessity of them. Note that when C_{oov} is excluded, we select at most 2K instances at each iteration by the probabilities from high to low. Table 4 shows the results. As shown, we can see that all conditions are useful, and in addition, all results outperform the plain iterative self-training method. In particular, the model $-\text{C}_{\text{oov}} - \text{C}_{\text{lex}}$ is degraded into the self-training with iterative adaption combined with the domain-mixed model. The comparison further demonstrates the advantage of our domain-mixed model.

The Size of Pseudo Training Corpus It is very interesting to compare the fine-grained domain adaption with (one-iteration) self-training under the view of the pseudo training dataset size. We align the iteration of fine-grained domain adaption with self-training by the added training corpus size of the ZhuXian domain. Figure 3 shows the comparison results. As shown, the performance of self-training would be hardly increasing after 3K instances, while our fine-grained method can give significant improvements continually until iteration 12 (consuming 20K corpus). The comparison shows that our fine-grained domain adaption is much more effective than self-training. However, our iterative fine-grained domain adaption needs more time to training than non-iterative self-training [4].

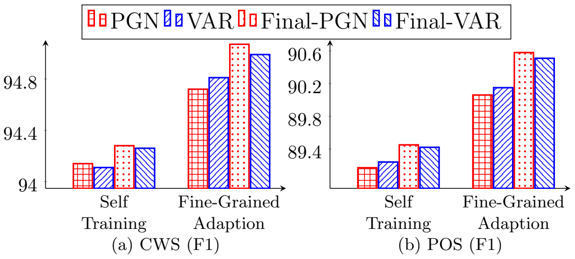
[4] The time cost of iterative fine-grained domain adaption is closely related to the number of iterations. Assuming the time cost of non-iterative training is C and our raw corpus is a closed set, the time cost of the iterative training is equal to $0.7 * N * C$, where N is the iteration number.

Model		P	R	F1
Baseline	Vanilla	**93.99**	**93.55**	**93.77**
Self-Training	Vanilla	94.01	94.08	94.04
	+Iterative	94.18	94.06	94.12
	+Domain-PGN	94.20	94.01	94.11
	+Domain-Mixed	**94.47**	**94.07**	**94.27**
Fine-Grained Adaption	Vanilla	94.65	94.47	94.51
	+Domain-PGN	94.70	94.51	94.60
	+Domain-Mixed	**95.27**	**94.64**	**94.86**

Table 5: The results of independent CWS task using our method on ZhuXian dataset.

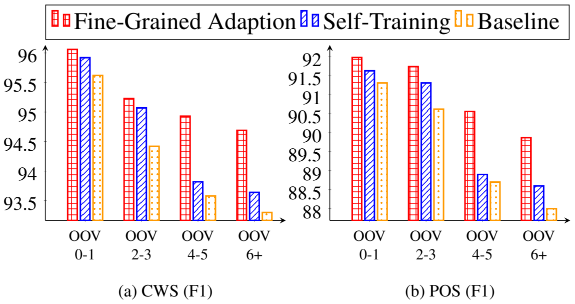
The Independent CWS Task Our major goal is for joint CWS and POS tagging, while it is expected to examine our method for the CWS task alone. Here we also use the CTB6 dataset as the source corpus and the ZhuXian dataset as the target domain. The basic model can be exactly the same. Table 5 shows the final results. Our method can achieve significant improvements on the CWS alone, resulting in increases of 94.86 - 93.77 = 1.09, which means that our fine-grained domain adaption method can be suitable for CWS as well. The other model tendencies are consistent with the joint task. Interestingly, we find that the independent CWS model has a lower improvement in recall. The reason may be that the POS tagging can provide several additional features, which let the joint model prefer more fine-grained segmentation, leading to a larger recall value.

Domain-Aware v.s. Domain-Invariant It is interesting to compare our (sub)domain-aware (PGN) and (sub)domain-invariant (VAR) components comprehensively. In fact, the two components alone can serve for domain adaption as well besides our integrated usage. The PGN can be used directly for inference, while for VAR, we can perform decoding by setting $\bar{\boldsymbol{e}}_i^{\text{dm}} = \boldsymbol{\mu}_i$ in Equation 3. Here we analyze four models, PGN and VAR alone, and the integrated model inferencing with PGN (Final-PGN) and VAR (Final-VAR), respectively. All four models are trained on the same and full training corpus (i.e., $S + \text{ST}_1, ..., S + \text{ST}_1 + ... + \text{ST}_n$, respectively and gradually). Figure 4 shows the results. As shown, we can see that PGN and VAR are actually comparable to each other, and in our final model, PGN is slightly better than VAR. We find that in our integrated model, both PGN and VAR are much better than using them alone, which shows the importance of the joint learning by the carefully-designed $\mathcal{L}_{\text{mse}}$.



(a) CWS (F1)
(b) POS (F1)

Figure 4: Comparisons between (sub)domain-aware (PGN) and (sub)domain-invariant (VAR) components, where PGN and VAR indicate that they are exploited separately for representation learning, and Final-PGN and Final-VAR denote our final model by using PGN/VAR for decoding, respectively.



(a) CWS (F1)
(b) POS (F1)

Figure 5: The results of Domain-Mixed model on different OOV distribution test datas use self-training and fine-grained adaption.

The Sentential OOV Number Our fine-grained domain adaption is mainly advanced by the sentential OOV numbers with respect to the source training dataset. Thus, it is meaningful to examine the model performance on sentences with different OOV numbers. We divide the ZhuXian test dataset by four categories according to the OOV number in sentence, which are respectively [0-1], [2-3], [4-5] and $\geq$6. All categories include a sufficient number of sentences for statistical comparisons. Based on the division, we compare the performance of the fine-grained adaption, self-training as well as baseline models. Figure 5 shows the results. We can see that with the increase of OOV number, the model performance can be decreased as a whole, which is reasonable. In addition, our final model can significantly improve the model performance with higher OOV numbers in sentence.

The Subdomain Type of Our Final Inference For the training of our final model, we have several fine-grained subdomain types of the target domain, and we select the last subdomain type for the final inference, which might be unmatched with the real subdomain type. Here we analyze the input domain

Domain	ZhuXian-CWS			ZhuXian-POS		
Type	P	R	F1	P	R	F1
ST_1	95.00	95.17	95.08	90.49	90.63	90.56
ST_6	94.98	95.16	95.07	90.49	90.65	90.57
ST_{11}	94.99	95.14	95.07	90.51	90.65	90.58

Table 6: The influence of using different domain types.

type selection in depth by comparing the model performance with the first (ST_1), median (ST_6) and last (ST_{11}) subdomain types. Table 6 shows the results. As shown, there is almost no difference between the three selections for the ZhuXian domain, indicating that the selection of fine-grained subdomain types is not important in our final model. The observation is reasonable since the test corpus cover a range of the specified subdomains and fixed selection can face the same issue, thus the final selection could be totally empirical.

5 Related Work

CWS and POS tagging are closely-related tasks for Chinese processing, which could be handled either jointly or in a pipeline way (Ng and Low, 2004; Shi and Wang, 2007; Zhang and Clark, 2008; Jiang et al., 2008; Kruengkrai et al., 2009; Jiang et al., 2009; Sun, 2011). The joint models are able to obtain better performances, as they can alleviate the error propagation problem between two tasks (Ng and Low, 2004; Zhang and Clark, 2008; Jiang et al., 2009; Wang et al., 2011). Recently, neural models lead to state-of-the-arts for joint CWS and POS tagging (Zheng et al., 2013; Shao et al., 2017; Zeng et al., 2013; Tian et al., 2020a). In particular, the BERT representations (Devlin et al., 2019) and the BiLSTM neural network (Graves et al., 2013; Huang et al., 2015) have shown impressive results for the joint task (Zhang et al., 2018; Diao et al., 2019; Tian et al., 2020a,b). In this work, we adopt both BERT and BiLSTM to reach a strong baseline for cross-domain adaption.

Domain adaptation has been extensively studied in both the machine learning and NLP communities (Daumé III, 2007; Ben-David et al., 2007; Chen et al., 2011; Søgaard, 2013; Zou et al., 2019; Saito et al., 2020). The typical methods of domain adaption can be divided into two categories mainly. The first category aims to create a set of pseudo training corpora for the target domain, while the second category attempts to learn transferable features from the source domain to the target. Self-training is one most representative methods of the first category (McClosky et al., 2006; Yu et al., 2015; Zou et al., 2019). For the second category, the representation learning of domain-specific and domain-invariant features has received the most attention recently (Glorot et al., 2011; Ganin et al., 2016; Tzeng et al., 2017; Long et al., 2017; Hoffman et al., 2018).

For the joint CWS and POS tagging task, Liu and Zhang (2012) and Zhang et al. (2014) investigate the task under the cross-domain adaption setting, both of which exploit self-training. In particular, Zhang et al. (2014) suggest a lexicon-based type-supervised model for further enhancement, and meanwhile publish a benchmark dataset which is publicly available for cross-domain adaption of joint CWS and POS tagging. Unfortunately, there is no future work for the joint task since then, while the majority of studies focus on the cross-domain of the two individual tasks (Liu et al., 2014; Schnabel and Schütze, 2014; Peng and Dredze, 2016; Huang et al., 2017; Zhou et al., 2017; Gui et al., 2017; Ding et al., 2020). We propose a novel fine-grained domain adaption method with a domain-mixed representation learning model for the joint task.

6 Conclusion

We suggested a novel fine-grained domain adaption method for joint word segmentation and POS tagging. We started from self-training strategy, which exploits various transfers to generate pseudo training instances for the target domain, and argued that the strategy might lead to low-quality of the auto-labeled instances when the target sentences are distant from the source domain. To address the problem, we proposed fine-grained domain adaption, regarding the OOV number to the source training corpus as the main advancing indicator to construct a higher quality corpus progressively. In addition, we combined our method with another line of representation learning of domain adaption, presenting a domain-mixed model for full exploration of the produced training instances. We evaluated our method on the benchmark ZhuXian dataset by using CTB6 as the source domain. The results showed that our method is highly effective, and our final model can achieve significant improvements on the joint task.

Acknowledgments

This work is supported by grants from the National Key Research and Development Program of China (No. 2018YFC0832101) and the National Natural Science Foundation of China (No. 62176180).

References

Shai Ben-David, John Blitzer, Koby Crammer, Fernando Pereira, et al. 2007. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19:137.

Samuel R. Bowman, Luke Vilnis, Oriol Vinyals, Andrew Dai, Rafal Jozefowicz, and Samy Bengio. 2016. Generating sentences from a continuous space. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 10–21.

Minmin Chen, Kilian Q Weinberger, and John Blitzer. 2011. Co-training for domain adaptation. In *Proceedings of NeurIPS*.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th ACL*, pages 256–263.

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

Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. 2019. Zen: pre-training chinese text encoder enhanced by n-gram representations. *arXiv preprint arXiv:1911.00720*.

Ning Ding, Dingkun Long, Guangwei Xu, Muhua Zhu, Pengjun Xie, Xiaobin Wang, and Haitao Zheng. 2020. Coupling distant annotation and adversarial training for cross-domain Chinese word segmentation. In *Proceedings of the ACL*, pages 6662–6671.

Thomas Emerson. 2005. The second international chinese word segmentation bakeoff. In *Proceedings of the fourth SIGHAN workshop on Chinese language Processing*.

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

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. 2016. Domain-adversarial training of neural networks. *JMLR*, 17(1):2096–2030.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the ICML*.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. 2013. Speech recognition with deep recurrent neural networks. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6645–6649. Ieee.

Tao Gui, Qi Zhang, Haoran Huang, Minlong Peng, and Xuan-Jing Huang. 2017. Part-of-speech tagging for twitter with adversarial neural networks. In *Proceedings of the EMNLP*, pages 2411–2420.

Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei Efros, and Trevor Darrell. 2018. Cycada: Cycle-consistent adversarial domain adaptation. In *Proceedings of the ICML*, pages 1989–1998.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *Proceedings of the ICML*, pages 2790–2799.

Shen Huang, Xu Sun, and Houfeng Wang. 2017. Addressing domain adaptation for chinese word segmentation with global recurrent structure. In *Proceedings of the IJCNLP*, pages 184–193.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, and Kiyoharu Aizawa. 2018. Cross-domain weakly-supervised object detection through progressive domain adaptation. In *Proceedings of the CVPR*, pages 5001–5009.

Chen Jia, Xiaobo Liang, and Yue Zhang. 2019. Cross-domain ner using cross-domain language modeling. In *Proceedings of the ACL*, pages 2464–2474.

Wenbin Jiang, Liang Huang, and Qun Liu. 2009. Automatic adaptation of annotation standards: Chinese word segmentation and pos tagging–a case study. In *Proceedings of the ACL-IJCNLP*, pages 522–530.

Wenbin Jiang, Liang Huang, Qun Liu, and Yajuan Lü. 2008. A cascaded linear model for joint chinese word segmentation and part-of-speech tagging. In *Proceedings of ACL*, pages 897–904.

Guangjin Jin and Xiao Chen. 2008. The fourth international chinese language processing bakeoff: Chinese word segmentation, named entity recognition and chinese pos tagging. In *Proceedings of the sixth SIGHAN workshop on Chinese language processing*.

Diederik P. Kingma and Max Welling. 2014. Auto-encoding variational bayes. In *Proceedings of the ICLR*.

Canasai Kruengkrai, Kiyotaka Uchimoto, Yiou Wang, Kentaro Torisawa, Hitoshi Isahara, et al. 2009. An error-driven word-character hybrid model for joint chinese word segmentation and pos tagging. In *Proceedings of the ACL-IJCNLP*, pages 513–521.

Yang Liu and Yue Zhang. 2012. Unsupervised domain adaptation for joint segmentation and POS-tagging. In *Proceedings of COLING 2012: Posters*, pages 745–754.

Yijia Liu, Yue Zhang, Wanxiang Che, Ting Liu, and Fan Wu. 2014. Domain adaptation for crf-based chinese word segmentation using free annotations. In *Proceedings of the EMNLP*, pages 864–874.

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I Jordan. 2017. Deep transfer learning with joint adaptation networks. In *Proceedings of the ICML*, pages 2208–2217.

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

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese part-of-speech tagging: One-at-a-time or all-at-once? word-based or character-based? In *Proceedings of the EMNLP*, pages 277–284.

Nanyun Peng and Mark Dredze. 2016. Multi-task domain adaptation for sequence tagging. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*.

Kuniaki Saito, Donghyun Kim, Stan Sclaroff, and Kate Saenko. 2020. Universal domain adaptation through self supervision. *arXiv preprint arXiv:2002.07953*.

Tobias Schnabel and Hinrich Schütze. 2014. Flors: Fast and simple domain adaptation for part-of-speech tagging. *TACL*, 2:15–26.

Yan Shao, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2017. Character-based joint segmentation and POS tagging for Chinese using bidirectional RNN-CRF. In *Proceedings of the IJCNLP*, pages 173–183.

Yanxin Shi and Mengqiu Wang. 2007. A dual-layer crfs based joint decoding method for cascaded segmentation and labeling tasks. In *Proceedings of the IJCAI*, pages 1707–1712.

Rui Shu, Hung H Bui, Hirokazu Narui, and Stefano Ermon. 2018. A dirt-t approach to unsupervised domain adaptation.

Anders Søgaard. 2013. Semi-supervised learning and domain adaptation in natural language processing. *Synthesis Lectures on Human Language Technologies*, 6(2):1–103.

Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Han Zhang, and Colin Raffel. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *arXiv preprint arXiv:2001.07685*.

Shiliang Sun, Honglei Shi, and Yuanbin Wu. 2015. A survey of multi-source domain adaptation. *Information Fusion*, 24:84–92.

Weiwei Sun. 2011. A stacked sub-word model for joint chinese word segmentation and part-of-speech tagging. In *Proceedings of the ACL*, pages 1385–1394.

Yuanhe Tian, Yan Song, Xiang Ao, Fei Xia, Xiaojun Quan, Tong Zhang, and Yonggang Wang. 2020a. Joint chinese word segmentation and part-of-speech tagging via two-way attentions of auto-analyzed knowledge. In *Proceedings of the ACL*, pages 8286–8296.

Yuanhe Tian, Yan Song, and Fei Xia. 2020b. Joint Chinese word segmentation and part-of-speech tagging via multi-channel attention of character n-grams. In *Proceedings of the 28th COLING*, pages 2073–2084.

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. 2017. Adversarial discriminative domain adaptation. In *Proceedings of the CVPR*, pages 7167–7176.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proceedings of the EMNLP*, pages 2302–2315.

Yiou Wang, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, Kentaro Torisawa, et al. 2011. Improving chinese word segmentation and pos tagging with semi-supervised methods using large auto-analyzed data. In *Proceedings of 5th IJCNLP*, pages 309–317.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2):207.

Juntao Yu, Mohab El-karef, 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.

Xiaodong Zeng, Derek F. Wong, Lidia S. Chao, and Isabel Trancoso. 2013. Graph-based semi-supervised model for joint Chinese word segmentation and part-of-speech tagging. In *Proceedings of the ACL*, pages 770–779.

Kun Zhang, Mingming Gong, and Bernhard Schölkopf. 2015. Multi-source domain adaptation: A causal view. In *Proceedings of the AAAI*, volume 29.

Meishan Zhang, Nan Yu, and Guohong Fu. 2018. A simple and effective neural model for joint word segmentation and pos tagging. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(9):1528–1538.

Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2014. Type-supervised domain adaptation for joint segmentation and POS-tagging. In *Proceedings of the 14th EACL*, pages 588–597.

Yue Zhang and Stephen Clark. 2008. Joint word segmentation and pos tagging using a single perceptron. In *Proceedings of the ACL-08: HLT*, pages 888–896.

Han Zhao, Remi Tachet Des Combes, Kun Zhang, and Geoffrey Gordon. 2019. On learning invariant representations for domain adaptation. In *Proceedings of the ICML*, pages 7523–7532.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for Chinese word segmentation and POS tagging. In *Proceedings of the EMNLP*, pages 647–657.

Hao Zhou, Zhenting Yu, Yue Zhang, Shujian Huang, Xinyu Dai, and Jiajun Chen. 2017. Word-context character embeddings for chinese word segmentation. In *Proceedings of the EMNLP*, pages 760–766.

Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. 2019. Confidence regularized self-training. In *Proceedings of the ICCV*, pages 5982–5991.

A Transformer with Adapters

Figure 6 illustrates the internal network structure of the transformer unit in ADBERT. As shown, we can see that two adapter layers are inserted inside each transformer unit:

$$\begin{aligned} \boldsymbol{h}_{\text{mid}} &= \text{GELU}(\boldsymbol{W}_1^{\text{share}}\boldsymbol{h}_{\text{in}} + \boldsymbol{b}_1^{\text{share}}), \\ \boldsymbol{h}_{\text{out}} &= \boldsymbol{W}_2^{\text{share}}\boldsymbol{h}_{\text{mid}} + \boldsymbol{b}_2^{\text{share}} + \boldsymbol{h}_{\text{in}}, \end{aligned} \tag{6}$$

where $\boldsymbol{W}_1^{\text{share}}$, $\boldsymbol{W}_2^{\text{share}}$, $\boldsymbol{b}_1^{\text{share}}$, $\boldsymbol{b}_2^{\text{share}}$ are adapter parameters, which are much smaller than those of BERT in scale.

Here we further emphasize that when BERT is powered with adapters, BERT can be regarded as a static knowledge by freezing all the pretrained parameters for downstream tasks, since the BERT parameter values can be shared across these tasks.

B Hyperparameters

For the model part, we set all the hidden sizes of BiLSTM to 400, and set the hidden sizes of all shared adapters to 192. We exploit the pretrained BERT-base-Chinese model for the character representations,[5] thus the output dimensional size of character representation is 768. The embedding of domain type is with a dimensional size of 50. For fine-grained domain adaption, the number of high-confidence word-tag pairs in Top-K is set by 1000, the probability threshold $p_{\text{threshold}}$ is 0.8.

For training, we exploit online learning with a batch size of 16 to update the model parameters, and use the Adam algorithm with a constant learning rate 2×10^{-5} to optimize the parameters. The gradient clipping mechanism by a maximum value of 5.0 is adopted to avoid gradient explosion. We use sequential-level dropout to the character representations to avoid overfitting, where the sequential hidden vectors are randomly set to zeros with a probability of 0.2. In particular, we have two hyperparameters λ_1 and λ_2 in our overall training objective, which is auto-adjust during the training from 0 to 1 by exponential annealing in the first 5,000 steps (Bowman et al., 2016).

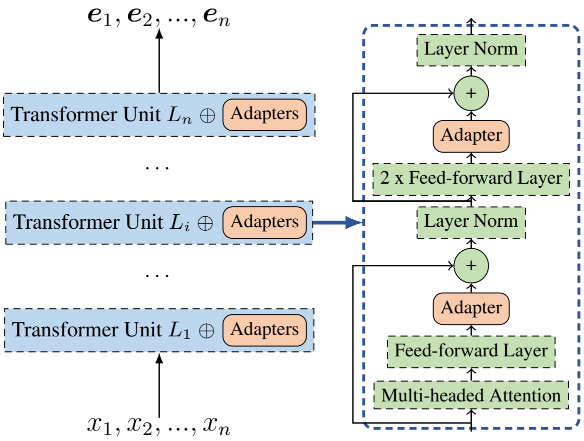

$\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_n$
Transformer Unit L_n ⊕ Adapters
Transformer Unit L_i ⊕ Adapters
Transformer Unit L_1 ⊕ Adapters
$x_1, x_2, ..., x_n$
Layer Norm
+
Adapter
2 x Feed-forward Layer
Layer Norm
+
Adapter
Feed-forward Layer
Multi-headed Attention

Figure 6: The structure of ADBERT.

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