Coupling Distant Annotation and Adversarial Training for Cross-Domain Chinese Word Segmentation

Ning Ding[1,2] **, Dingkun Long**[2]**, Guangwei Xu**[2]**,**
Muhua Zhu[2]**, Pengjun Xie**[2]**, Xiaobin Wang**[2]**, Hai-Tao Zheng**[1*]
[1]Tsinghua University, China [2]Alibaba Group
{dingn18}@mails.tsinghua.edu.cn, {zhumuhua}@gmail.com,
{dingkun.ldk,kunka.xgw,chengchen.xpj,xuanjie.wxb}@alibaba-inc.com,
{zheng.haitao}@sz.tsinghua.edu.cn,

Abstract

Fully supervised neural approaches have achieved significant progress in the task of Chinese word segmentation (CWS). Nevertheless, the performance of supervised models tends to drop dramatically when they are applied to out-of-domain data. Performance degradation is caused by the distribution gap across domains and the out of vocabulary (OOV) problem. In order to simultaneously alleviate these two issues, this paper proposes to couple distant annotation and adversarial training for cross-domain CWS. For distant annotation, we rethink the essence of "Chinese words" and design an automatic distant annotation mechanism that does not need any supervision or pre-defined dictionaries from the target domain. The approach could effectively explore domain-specific words and distantly annotate the raw texts for the target domain. For adversarial training, we develop a sentence-level training procedure to perform noise reduction and maximum utilization of the source domain information. Experiments on multiple real-world datasets across various domains show the superiority and robustness of our model, significantly outperforming previous state-of-the-art cross-domain CWS methods.

1 Introduction

Chinese is an ideographic language and lacks word delimiters between words in written sentences. Therefore, Chinese word segmentation (CWS) is often regarded as a prerequisite to downstream tasks in Chinese natural language processing. This task is conventionally formalized as a character-based sequence tagging problem (Peng et al., 2004), where each character is assigned a specific label to denote the position of the character in a word. With the development of deep learning techniques, recent years have also seen increasing interest in applying neural network models onto CWS (Cai and Zhao, 2016; Liu et al., 2016; Cai et al., 2017; Ma et al., 2018). These approaches have achieved significant progress on in-domain CWS tasks, but they still suffer from the cross-domain issue when they come to processing of out-of-domain data.

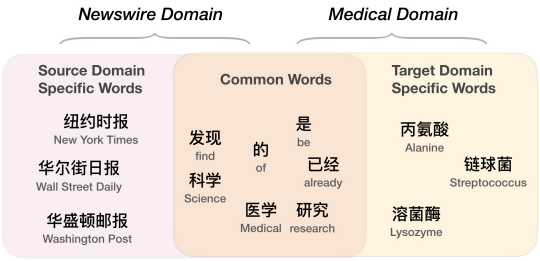


Figure 1: Different word distributions for the newswire domain and the medical domain.

Cross-domain CWS is exposed to two major challenges: 1) **Gap of domain distributions**. This is a common issue existing in all domain adaptation tasks. Source domain data and target domain data generally have different distributions. As a result, models built on source domain data tend to degrade performance when they are applied to target domain data. Generally, we need some labeled target domain data to adapt source domain models, but it is expensive and time consuming to manually craft such data. 2) **Out of vocabulary (OOV) problem**, which means there exist some words in the testing data that never occur in the training data. Source domain models have difficulties in recognizing OOV words since source domain data contains no information on the OOVs. Figure 1 presents examples to illustrate the difference between the word distributions of the newswire domain and the medical domain. Segmenters built on the newswire domain have very limited information to segment domain-specific words like "溶菌酶 (Lysozyme)".

Previous approaches to cross-domain CWS mainly fall into two groups. The first group aims to attack the OOV issue by utilizing predefined dictionaries from the target domain to facilitate cross-domain CWS (Liu et al., 2014; Zhao et al.,

* Corresponding author

2018; Zhang et al., 2018), which are apt to suffer from scalability since not all domains possess predefined dictionaries. In other words, these methods are directly restricted by external resources that are available in a target domain. Studies in the second group (Ye et al., 2019) attend to learn target domain distributions like word embeddings from unlabeled target domain data. In this approach, source domain data is not fully utilized since the information from source domain data is transferred solely through the segmenter built on the data.

In this paper, we propose to attack the aforementioned challenges simultaneously by coupling the techniques of *distant annotation* and *adversarial training*. The goal of distant annotation is to automatically construct labeled target domain data with no requirement for human-curated domain-specific dictionaries. To this end, we rethink the definition and essence of "Chinese words" and develop a word miner to obtain domain-specific words from unlabeled target domain data. Moreover, a segmenter is trained on the source domain data to recognize the common words in unlabeled target data. This way, sentences from the target domain are assigned automatic annotations that can be used as target domain training data.

Although distant annotation could provide satisfactory labeled target domain data, there still exist annotation errors that affect the final performance. To reduce the effect of noisy data in automatic annotations in target domain data and make better use of source domain data, we propose to apply adversarial training jointly on the source domain dataset and the distantly constructed target domain dataset. And the adversarial training module can capture deeper domain-specific and domain-agnostic features.

To show the effectiveness and robustness of our approach, we conduct extensive experiments on five real-world datasets across various domains. Experimental results show that our approach achieves state-of-the-art results on all datasets, significantly outperforming representative previous works. Further, we design sufficient subsidiary experiments to prove the alleviation of the aforementioned problems in cross-domain CWS.

2 Related Work

Chinese Word Segmentation Chinese word segmentation is typically formalized as a sequence tagging problem. Thus, traditional machine learning models such as Hidden Markov Models (HMMs) and Conditional Random Fields (CRFs) are widely employed for CWS in the early stage (Wong and Chan, 1996; Gao et al., 2005; Zhao et al., 2010). With the development of deep learning methods, research focus has been shifting towards deep neural networks that require little feature engineering. Chen et al. (2015) are the first that use LSTM (Hochreiter and Schmidhuber, 1997) to resolve long dependencies in word segmentation problems. Since then, the majority of efforts is building end-to-end sequence tagging architectures, which significantly outperform the traditional approaches on CWS task (Wang and Xu, 2017; Zhou et al., 2017; Yang et al., 2017; Cai et al., 2017; Chen et al., 2017; Huang et al., 2019b; Gan and Zhang, 2019; Yang et al., 2019).

Cross-domain CWS As a more challenging task, cross-domain CWS has attracted increasing attention. Liu and Zhang (2012) propose an unsupervised model, in which they use a character clustering method and the self-training algorithm to jointly model CWS and POS-tagging. Liu et al. (2014) apply partial CRF for cross-domain CWS via obtaining a partial annotation dataset from freely available data. Similarly, Zhao et al. (2018) build partially labeled data by combining unlabeled data and lexicons. Zhang et al. (2018) propose to incorporate the predefined domain dictionary into the training process via predefined handcrafted rules. Ye et al. (2019) propose a semi-supervised approach that leverages word embeddings trained on the segmented text in the target domain.

Adversarial Learning Adversarial learning is derived from the Generative Adversarial Nets (GAN) (Goodfellow et al., 2014), which has achieved huge success in the computer vision field. Recently, many works have tried to apply adversarial learning to NLP tasks. (Jia and Liang, 2017; Li et al., 2018; Farag et al., 2018) focus on learning or creating adversarial rules or examples for improving the robustness of the NLP systems. For cross-domain or cross-lingual sequence tagging, the adversarial discriminator is widely used to extract domain or language invariant features (Kim et al., 2017; Huang et al., 2019a; Zhou et al., 2019).

3 Our Approach

Figure 2 shows the framework of our approach to cross-domain CWS, which is mainly composed of two components: 1) **Distant Annotation (DA)**,

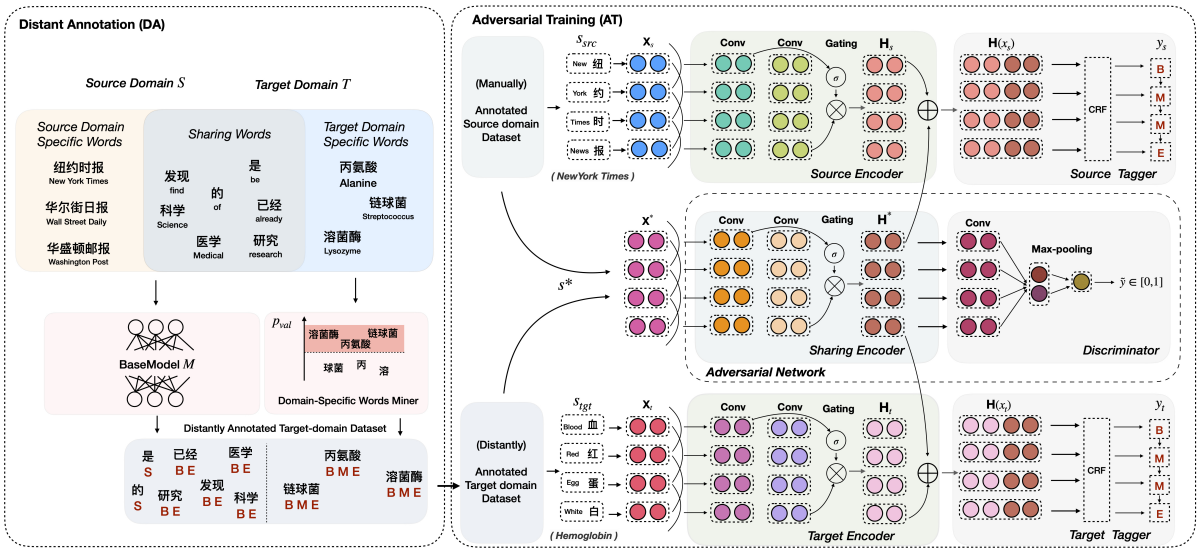


Figure 2: Detailed architecture of DAAT, the left part is the structure of the Distant Annotation (DA) module. The annotated dataset on target domain will be sent to the Adversarial Training (AT) module on the right part.

and 2) **Adversarial Training (AT)**. In the following, we will describe details of the framework (DAAT) from the left to right in Figure 2.

In this paper, bold-face letters (e.g. $\boldsymbol{W}$) are used to denote vectors, matrices and tensors. We use numerical subscripts to indicate the indices of a sequence or vector. We use the subscript of src to indicate the source domain and tgt to denote the target domain.

3.1 Distant Annotation

As illustrated in Figure 2, given a labeled source domain dataset and an unlabeled target domain dataset, distant annotation (DA) aims to automatically generate word segmentation results for sentences in the target domain. DA has two main modules, including a base segmenter and a *Domain-specific Words Miner*. Specifically, the base segmenter is a GCNN-CRF (Wang and Xu, 2017) model trained solely on the labeled source domain data and is used to recognize words that are common among the source and target domains. *Domain-specific Words Miner* is designed to explore the target domain-specific words.

Base Segmenter In the CWS task, given a sentence $s = \{c_1, c_2, ..., c_n\}$, following the *BMES* tagging scheme, each character c_i is assigned one of the labels in $\{B, M, E, S\}$, indicating whether the character is in the beginning, middle, end of a word, or the character is merely a single-character word.

For a sentence s, we first use an embedding layer to obtain the embedding representation $\boldsymbol{e}_i$ for each character c_i. Then, the sentence s can be represented as $\boldsymbol{e} = \{\boldsymbol{e}_1, \boldsymbol{e}_2, ..., \boldsymbol{e}_n\} \in \mathbb{R}^{n \times d}$, where d denotes the embedding dimension. $\boldsymbol{e}$ will be fed into the GCNN model (Dauphin et al., 2017; Gehring et al., 2017), which computes the output as:

$$\boldsymbol{H}_s = (\boldsymbol{e} * \boldsymbol{W} + b) \odot \sigma(\boldsymbol{e} * \boldsymbol{V} + c), \quad (1)$$

here, $\boldsymbol{W} \in \mathbb{R}^{k \times d \times l}$, $b \in \mathbb{R}^l$, $\boldsymbol{V} \in \mathbb{R}^{k \times d \times l}$, $c \in \mathbb{R}^l$. d and l are the input and output dimensions respectively, and k is the window size of the convolution operator. σ is the sigmoid function and $\odot$ represents element-wise product. We adopt a stacking convolution architecture to capture long distance information, the output of the previous layers will be treated as input of the next layer. The final representation of sentence s is $\boldsymbol{H}_s = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_n\}$.

Correlations among labels are crucial factors in sequence tagging. Particularly, for an input sequence $s_{src} = \{c_1, c_2, ..., c_n\}$ (take source domain data as example), the corresponding label sequence is $L = \{y_1, y_2, ..., y_n\}$. The goal of CRF is to compute the conditional probability distribution:

$$P(L|s_{src}) = \frac{exp(\sum_{i=1}^{n}(S(y_i) + T(y_{i-1}, y_i)))}{\sum_{L' \in \mathbb{C}} exp(\sum_{i=1}^{n}(S(y'_i) + T(y'_{i-1}, y'_i)))}, \quad (2)$$

where T denotes the transition function to calculate the transition scores from y_{i-1} to y_i. $\mathbb{C}$ contains all the possible label sequences on sequence s and L' is a random label sequence in $\mathbb{C}$. And S represents the score function to compute the emission score from the hidden feature vector $\boldsymbol{h}_i$ to the cor-

responding label y_i, which is defined as:

$$S(y_i) = \boldsymbol{W}^{y_i}\boldsymbol{h}_i + b^{y_i}, \quad (3)$$

$\boldsymbol{W}^{y_i}$ and b^{y_i} are learned parameters specific to the label y_i.

To decode the highest scored label sequence, a classic Viterbi (Viterbi, 1967) algorithm is utilized as the decoder. The loss function of the sequence tagger is defined as the sentence-level negative log-likelihood:

$$\mathcal{L}_{src} = -\sum \log P(L|s_{src}). \quad (4)$$

The loss of the target tagger $\mathcal{L}_{tgt}$ could be computed similarly.

Domain-specific Words Miner As mentioned in section 1, previous works usually use existing domain dictionaries to solve the domain-specific noun entities segmentation problem in cross-domain CWS. But this strategy does not consider that it is properly difficult to acquire a dictionary with high quality for a brand new domain. In contrast, we develop a simple and efficient strategy to perform domain-specific words mining without any predefined dictionaries.

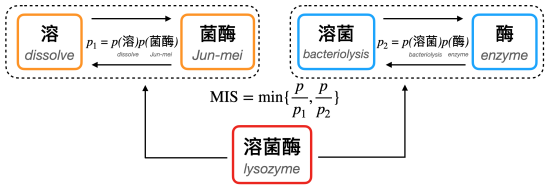
Given large raw text on target domain and a base segmenter, we can obtain a set of segmented texts $\Gamma = \{T_1, T_2, ..., T_N\}$, where stop-words are removed. Then let $\gamma = \{t_1, t_2, ..., t_m\}$ denote all the n-gram sequences extracted from Γ. For each sequence t_i, we need to calculate the possibility that it is a valid word. In this procedure, four factors are mainly considered.

1) *Mutual Information* (MI). MI (Kraskov et al., 2004) is widely used to estimate the correlation of two random variables. Here, we use mutual information between different sub-strings to measure the internal tightness for a text segment, as shown in Figure 3(a). Further, in order to exclude extreme cases, it is necessary to enumerate all the sub-string candidates. The final MI score for one sequence t_i consists of n characters $t_i = \{c_1...c_n\}$ is defined as:

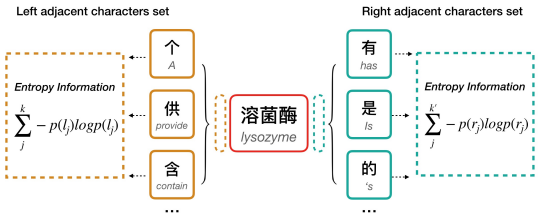
$$\mathrm{MIS}(t_i) = \min_{j\in[1:n]}\{\frac{p(t_i)}{p(c_1...c_j)\cdot p(c_{j+1}...c_n)}\}, \quad (5)$$

where $p(\cdot)$ denotes the probability given the whole corpus Γ.

2) *Entropy Score* (ES). Entropy is a crucial concept aiming at measuring the uncertainty of random variables in information theory (Jaynes, 1957).



(a) Mutual Information to measure the internal tightness.



(b) Entropy Score to measure the external flexibility.

Figure 3: Examples of Mutual Score and Entropy Information factors.

Thus, we can use ES to measure the uncertainty of candidate text fragment, since higher uncertainty means a richer neighboring context. Let $N_l(t_i) = \{l_1, ..., l_k\}$ and $N_r(t_i) = \{r_1, ..., r_{k'}\}$ be the set of left and right adjacent characters for t_i. The left entropy score ES_l and right entropy ES_r of t_i can be formulated as $\mathrm{ES}_l(t_i) = \sum_j^k -p(l_j)\log p(l_j)$ and $\mathrm{ES}_r(t_i) = \sum_j^{k'} -p(r_j)\log p(r_j)$ respectively. We choose $\min(\mathrm{ES}_l(t_i), \mathrm{ES}_r(t_i))$ as the final score for t_i. Hence, $\mathrm{ES}(t_i)$ could explicitly represent the external flexibility for a text segment (as shown in Figure 3(b)), and further serve as an important indicator to judge whether the segment is an independent word.

3) *tf-idf*. tf-idf is a widely used numerical statistic that can reflect how important a word is to a document in a collection or corpus. As illustrated in Figure 1, most of the domain-specific words are noun entities, which share a large weighting factor in general.

In this work, we define a word probability score $p_{val}(t_i)$ to indicate how likely t_i can be defined as a valid word.

$$p_{val}(t_i) = \sigma(\mathrm{N}[\mathrm{MIS}(t_i)] + \mathrm{N}[\mathrm{ES}(t_i)] + \mathrm{N}[\mathrm{tfidf}(t_i)]), \quad (6)$$

where σ denotes the sigmoid function and N denotes normalization operation with the max-min method.

4) *Word frequency*. If t_i is a valid word, it should appear repeatedly in Γ.

Finally, by setting an appropriate threshold for $p_{val}(t_i)$ and word frequence, the *Domain-Specific*

Words Miner could effectively explore domain-specific words, then construct the domain-specific word collection $\mathcal{C}$ for the target domain. In this work, we only consider words t_i with $p_{val}(t_i) \geq 0.95$ and frequency larger than 10.

The left part of Figure 2 illustrates the data construction process of *DA*. First, we utilize the *Domain-specific Words Miner* to build the collection $\mathcal{C}$ for the target domain. Take sentence "溶酶菌的科学研究 (Scientific research on lysozyme)" as an example, we use the forward maximizing match algorithm based on $\mathcal{C}$, which shows that "溶酶菌 (lysozyme)" is a valid word. Hence, the labels of characters "溶", "酶", "菌" are "B", "M", "E". For the left part of the sentence, we adopt the baseline segmenter to perform the labelling process. "的科学研究" will be assigned with {"S", "B"."E", "B", "E"}. To this end, we are able to automatically build annotated dataset on the target domain.

3.2 Adversarial Training

The structure of the Adversarial Training module is illustrated as the right part of Figure 2. As mentioned in 3.1, we construct an annotated dataset for the target domain. Accordingly, the inputs of the network are two labeled datasets from source domain $\mathcal{S}$ and target domain $\mathcal{T}$. There are three encoders to extract features with different emphases, and all the encoders are based on GCNN as introduced in section 3.1. For domain-specific features, we adopt two independent encoders E_{src} and E_{tgt} for source domain and target domain. For domain-agnostic features, we adopt a sharing encoder E_{shr} and a discriminator G_d, which will be both trained as adversarial players.

For the two domain-specific encoders, the input sentence is s_{src}=$\{c_1^s, c_2^s, ..., c_n^s\}$ from source domain, or sentence s_{tgt}=$\{c_1^t, c_2^t, ..., c_m^t\}$ from the target domain. The sequence representation of s_{src} and s_{tgt} can be obtained by E_{src} and E_{tgt}. Thus, the domain independent representations of s_{src} and s_{tgt} are $\boldsymbol{H}_s \in \mathbb{R}^{n \times l}$ and $\boldsymbol{H}_t \in \mathbb{R}^{m \times l}$, where n and m denote the sequence lengths of s_{src} and s_{tgt} respectively, l is the output dimension of GCNN encoder.

For the sharing encoder, we hope that E_{shr} is able to generate representations that could fool the sentence level discriminator to correctly predict the domain of each sentence, such that E_{shr} finally extracts domain-agnostic features. Formally, given sentences s_{src} and s_{tgt} from source domain and target domain, E_{shr} will produce sequence features $\boldsymbol{H}_s^*$ and $\boldsymbol{H}_t^*$ for s_{src} and s_{tgt} respectively.

The discriminator G_d of the network aims to distinguish the domain of each sentence. Specifically, we will feed the final representation $\boldsymbol{H}^*$ of every sentence s to a binary classifier G_y where we adopt the text CNN network (Kim, 2014). G_y will produce a probability that the input sentence s is from the source domain or target domain. Thus, the loss function of the discriminator is:

$$\begin{aligned}\mathcal{L}_d = &- \mathbb{E}_{s \sim p_{\mathcal{S}}(s)}[\log G_y(E_{shr}(s)] \\ &- \mathbb{E}_{s \sim p_{\mathcal{T}}(s)}[\log (1 - G_y(E_{shr}(s))],\end{aligned} \tag{7}$$

Features generated by the sharing encoder E_{shr} should be able to fool the discriminator to correctly predict the domain of s. Thus, the loss function for the sharing encoder $\mathcal{L}_c$ is a flipped version of $\mathcal{L}_d$:

$$\begin{aligned}\mathcal{L}_c = &- \mathbb{E}_{s \sim p_{\mathcal{S}}(s)}[\log (1 - G_y(E_{shr}(s)])) \\ &- \mathbb{E}_{s \sim p_{\mathcal{T}}(s)}[\log G_y(E_{shr}(s)],\end{aligned} \tag{8}$$

Finally, we concatenate $\boldsymbol{H}$ and $\boldsymbol{H}^*$ as the final sequence representation of the input sentence. For s_{src} from source domain, $\boldsymbol{H}(s_{src}) = [\boldsymbol{H}_s \oplus \boldsymbol{H}_s^*]$, while for s_{tgt} from the target domain, $\boldsymbol{H}(s_{tgt}) = [\boldsymbol{H}_t \oplus \boldsymbol{H}_t^*]$. The final representation will be fed into the CRF tagger.

So far, our model can be jointly trained in an end-to-end manner with the standard back-propagation algorithm. More details about the adversarial training process are described in Algorithm 1. When there is no annotated dataset on the target domain, we could remove $\mathcal{L}_{tgt}$ during the adversarial training process and use the segmenter on source domain for evaluation.

Algorithm 1 Adversarial training algorithm.

Input: Manually annotated dataset $\mathcal{D}_s$ for source domain $\mathcal{S}$, and distantly annotated dataset $\mathcal{D}_t$ for target domain $\mathcal{T}$

```
for i ← 1 to epochs do
    for j ← 1 to num_of_steps per epoch do
        Sample mini-batches X_s ~ D_s, X_t ~ D_t
        if j%2 = 1 then
            loss = L_src + L_tgt + L_d
            Update θ w.r.t loss
        else
            loss= L_src + L_tgt + L_c
            Update θ w.r.t loss
        end
    end
end
```

Dataset			Sents	Words	Chars	Domain
SRC	PKU	Train	47.3K	1.1M	1.8M	News
		Test	6.4K	0.2M	0.3M	
TGT	DL	Full	40.0K	2.0M	2.9M	Novel
		Test	1.0K	32.0K	47.0K	
	FR	Full	148K	5.0M	7.1M	Novel
		Test	1.0K	17.0K	25.0K	
	ZX	Full	59.0K	2.1M	3.0M	Novel
		Test	1.0K	21K	31.0K	
	DM	Full	32.0K	0.7M	1.2M	Medical
		Test	1.0K	17K	30K	
	PT	Full	17.0K	0.6M	0.9M	Patent
		Test	1.0K	34.0K	57.0K	

Table 1: Statistics of datasets. The datasets of the target domain (TGT) are originally raw texts without golden segmentation, and the statistics are obtained by the baseline segmenter. The *DA* module will distantly annotate the datasets as mentioned in 3.1.

4 Experiments

In this section, we conduct extensive cross-domain CWS experiments on multiple real-world datasets with different domains, then comprehensively evaluate our method and other approaches.

4.1 Datasets and Experimental Settings

Datasets Six datasets across various domains are used in our work. The statistics of all datasets are shown in Table 1. In this paper, we use PKU dataset (Emerson, 2005) as the source domain data, which is a benchmark CWS dataset on the newswire domain. In addition, the other five datasets in other domains will be utilized as the target domain datasets. Among the five target domain datasets there are three Chinese fantasy novel datasets, including DL *(DoLuoDaLu)*, FR *(FanRenXiuXianZhuan)* and ZX *(ZhuXian)* (Qiu and Zhang, 2015). An obvious advantage for fantasy novel datasets is that there are a large number of proper words originated by the author for each fiction, which could explicitly reflect the alleviation of the OOV problem for an approach. Besides the fiction datasets, we also use DM (dermatology) and PT (patent) datasets (Ye et al., 2019), which are from *dermatology* domain and *patent* domain respectively. All the domains of the target datasets are very different from the source dataset (newswire). To perform a fair and comprehensive evaluation, the full/test settings of the datasets follow Ye et al. (2019).

Hyper-Parameters Table 2 shows the hyper-parameters used in our method. All the models are implemented with Tensorflow (Abadi et al., 2016) and trained using mini-batched back-propagation. Adam optimizer (Kingma and Ba, 2015) is used for optimization. The models are trained on NVIDIA Tesla V100 GPUs with CUDA[1].

Evaluation Metrics We use standard micro-averaged precision (P), recall (R) and F-measure as our evaluation metrics. We also compute OOV rates to reflect the degree of the OOV issue.

4.2 Compared Methods

We make comprehensive experiments with selective previous proposed methods, which are: **Partial CRF** (Liu et al., 2014) builds partially annotated data using raw text and lexicons via handcrafted rules, then trains the CWS model based on both labeled dataset (PKU) and partially annotated data using CRF. **CWS-DICT** (Zhang et al., 2018) trains the CWS model with a BiLSTM-CRF architecture, which incorporates lexicon into a neural network by designing handcrafted feature templates. For fair comparison, we use the same domain dictionaries produced by the *Domain-specific Words Miner* for **Partial CRF** and **CWS-DICT** methods. **WEB-CWS** (Ye et al., 2019) is a semi-supervised word-based approach using word embeddings trained with segmented text on target domain to improve cross-domain CWS.

Besides, we implement strong baselines to perform a comprehensive evaluation, which are: **GCNN (PKU)** uses the PKU dataset only, and we adopt the GCNN-CRF sequence tagging architecture (Wang and Xu, 2017). **GCNN (Target)** uses the distantly annotated dataset built on the target domain only. **GCNN (Mix)** uses the mixture dataset with both the PKU dataset and the distantly annotated target domain dataset. **DA** is a combination of GCNN (PKU) and domain-specific words. Details are introduced in 3.1. **AT** denotes the setting that we adopt adversarial training when no distantly annotated dataset on the target domain is provided, but the raw text is available.

4.3 Overall Results

The final results are reported in Table 3, from which we can observe that:

(1) Our DAAT model significantly outperforms previously proposed methods on all datasets, yielding the state-of-the-art results. Particularly, DAAT improves the F1-score on the five datasets from 93.5 to 94.1, 90.2 to 93.1, 89.6 to 90.9, 82.8 to 85.0 and 85.9 to 89.6 respectively. The results demon-

[1] source code and dataset will be available at https://github.com/Alibaba-NLP/DAAT-CWS

Hyper-parameter Name	Value
Threshold for p_{val}	0.95
Char emb size	200
GCNN output dim	200
Text CNN num of filters	200
Text CNN filter size	[3,4,5]
GCNN layers	5
Dropout Rate	0.3
Batch size	128
Learning rate	0.001
Epochs	30

Table 2: Hyper-parameters.

strate that the unified framework is empirically effective, for the alleviation of the OOV problem and the full utilization of source domain information.

(2) As mentioned in section 3, the AT model uses the same adversarial training network as the DAAT, yet without annotation on the target domain dataset. Results on the AT setting could explicitly reflect the necessity to construct the annotated target domain dataset. Specifically, without the constructed dataset, the AT method only yields 90.7, 86.8, 85.0, 81.0 and 85.1 F1-scores on five datasets respectively. But when use the annotated target domain dataset, we can get the DAAT with the best performance.

(3) WEB-CWS was the state-of-the-art approach that utilizes word embeddings trained on the segmented target text. Yet it is worth noticing that our model that only combines the base segmenter trained on PKU and domain-specific words (*DA*) could outperform WEB-CWS, which indicates that the distant annotation method could exploit more and deeper semantic features from the raw text. For the CWS-DICT method, which requires an external dictionary, we use the word collection (built by the *Domain-specific Words Miner*) to guarantee the fairness of the experiments. We can observe that our framework could yield significantly better results than CWS-DICT. Moreover, CWS-DICT needs existing dictionaries as external information, which is difficult for the model to transfer to brand new domains without specific dictionaries. In contrast, our framework utilizes the *Domain-specific Words Miner* to construct the word collection with high flexibility across domains.

4.4 Effect of Distant Annotation

In this section, we focus on exploring the ability to tackle the **OOV problem** for the *DA* method, which could distantly construct an annotated dataset from the raw text on the target domain. As illustrated in Table 4, the cross-domain CWS task suffers from a surprisingly serious OOV problem. All OOV rates (source) are above 10%, which will definitely degrade model performance. Nevertheless, after constructing an annotated dataset on the target domain, the OOV rate (target) drops significantly. Specifically, the *DA* method yields 9.92%, 13.1%, 14.09% 20.51% and 14.94% absolute OOV rate drop on the five out-domain datasets. The statistical result reveals that the *Domain-specific Words Miner* could accurately explore specific domain words for any domains from raw texts. Therefore, the *DA* of our framework could efficaciously tackle the OOV problem. Moreover, the module does not need any specific domain dictionaries, which means it can be transferred to new domains without limitations.

4.5 Impact of the Threshold p_{val}

Obviously, the setting of the hyper-parameter p_{val} will directly affect the scale and quality of the domain-specific word collection. To analyze how p_{val} affects the model performance, we conduct experiments with different setting p_{val} in $\{0.7, 0.8, 0.9, 0.95, 0.99\}$, and the size of word collection and model performance on DL and DM datasets are shown in Figure 4. Constant with intuition, the collection size will decrease as the increase of p_{val} because the filter criterion for words will get more strict, which is also a process of noise reduction. However, the F1-score curves are not incremental or descending. When $p_{val} <= 0.95$, the F1-scores on two datasets will increase because the eliminated words of this stage are mostly wrong. While the F1-scores will maintain or decrease when $p_{val} > 0.95$, because in this case, some correct words will be eliminated. We set $p_{val} = 0.95$ to guarantee the quality and quantity of the word collection simultaneously, so as to guarantee the model performance. And in this setting, the collection sizes are 0.7k words for DL, 1.7k for FR, 3.3k for ZX, 1.5k for DM and 2.2k for PT respectively.

4.6 Effect of Adversarial Learning

We develop an adversarial training procedure to reduce the noise in the annotated dataset produced by *DA*. In Table 3, we find that GCNN (Target) method trained on the annotated target dataset constructed by *DA* achieves impressive performance on all the five datasets, outperforming the WEB-CWS method. In addition, with the adversarial training module, the model further yields the remark-

Dataset	Previous Methods (F1-score)			Ours (F1-score)					
	Partial CRF	CWS-DICT	WEB-CWS	AT	GCNN (PKU)	DA	GCNN(Mix)	GCNN (Target)	**DAAT**
DL	92.5	92.0	**93.5**	90.7	90.0	93.6	93.9	93.9	**94.1 (+0.6)**
FR	**90.2**	89.1	89.6	86.8	86.0	92.4	92.6	92.6	**93.1 (+2.9)**
ZX	83.9	88.8	**89.6**	85.0	85.4	90.4	90.6	90.7	**90.9 (+1.3)**
DM	**82.8**	81.2	82.2	81.0	82.4	83.8	83.9	84.3	**85.0 (+2.2)**
PT	85.0	**85.9**	85.1	85.1	87.6	89.1	89.3	89.3	**89.6 (+3.7)**

Table 3: The overall results on five datasets. The first block contains the latest cross-domain methods. And the second block reports the results for our implemented methods and DAAT. Numbers in the parentheses indicate absolute improvement than previous SOTA results.

Dataset		OOV rate (source)	OOV rate (target)
Source	PKU	3.70%	-
Target	DL	11.15%	**1.23%**
	FR	14.08%	**0.98%**
	ZX	15.52%	**1.43%**
	DM	25.93%	**5.42%**
	PT	18.39%	**3.45%**

Table 4: OOV rates on five datasets. OOV rate (source) means the OOV rate test dataset and PKU dataset. OOV rate (target) means the OOV rate between the test dataset and the constructed annotated target dataset.

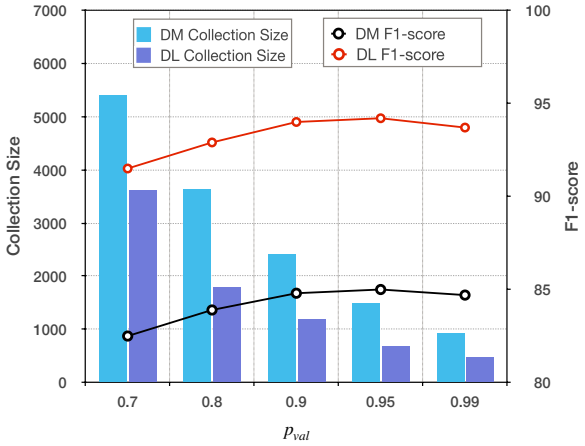
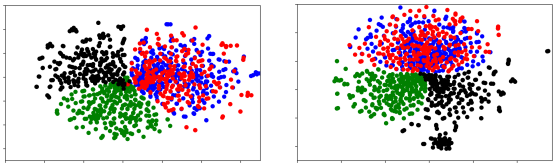


Figure 4: The impact of different p_{val} on mined collection size and model performance.

able improvements of the F1-scores. The results demonstrate that the adversarial network could capture deeper semantic features than simply using the GCNN-CRF model, via better making use of the information from both source and target domains.

4.7 Analysis of Feature Distribution

As introduced in 3.2, in the process of adversarial learning, domain-independent encoders could learn domain-specific features $\boldsymbol{H}_s$ and $\boldsymbol{H}_t$, and the sharing encoder could learn domain-agnostic features $\boldsymbol{H}_s^*$ and $\boldsymbol{H}_t^*$. We use t-SNE (Maaten and Hinton, 2008) algorithm to project these feature representations into planar points for visualization to further analyze the feature learning condition. As illustrated in Figure 5, domain-independent features $\boldsymbol{H}_s$



(a) Features on DM. (b) Features on DL.

Figure 5: t-SNE visualisation of $\boldsymbol{H}$ and $\boldsymbol{H}^*$ produced by the domain independent encoder and sharing encoder. Where green points $\rightarrow \boldsymbol{H}_s$. black points $\rightarrow \boldsymbol{H}_t$, blue points $\rightarrow \boldsymbol{H}_s^*$, red points $\rightarrow \boldsymbol{H}_t^*$

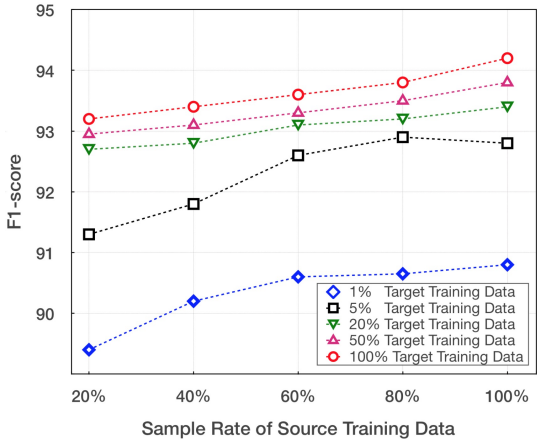


Figure 6: The impact of data amount for the source and target data on PKU (source, 47.3k sentences) and DL (target, 40.0k sentences).

(green) and $\boldsymbol{H}_t$ (black) have little overlap, indicating the distribution gap between different domains. However, the domain-agnostic feature distributions $\boldsymbol{H}_s^*$ (red) and $\boldsymbol{H}_t^*$ (blue) are very similar, implying that the learned feature representation can be well shared by both domains.

4.8 Impact of Amount from Source and Target data

In this subsection, we analyze the impact of the data usage for both source and target domain, the experiment is conducted on the PKU (source) and DL (target) datasets. In Figure 6, we respectively select 20%, 40%, 60%, 80% and 100% of the source do-

main data and 1%, 5%, 20%, 50%, 100% of the target domain data to perform the training procedure. The result demonstrates that increasing source and target data will both lead to an increase F1-score. Generally, the amount of the target data gives more impact on the whole performance, which conforms to the intuition. The " 1% Target Training Data" line indicates that the performance of the model will be strictly limited if the target data is severely missing. But when the amount of the target data increase to 5%, the performance will be improved significantly, which shows the ability to explore domain-specific information for our method.

5 Conclusion

In this paper, we intuitively propose a unified framework via coupling distant annotation and adversarial training for the cross-domain CWS task. In our method, we investigate an automatic distant annotator to build the labeled target domain dataset, effectively address the OOV issue. Further, an adversarial training procedure is designed to capture information from both the source and target domains. Empirical results show that our framework significantly outperforms other proposed methods, achieving the state-of-the-art result on all five datasets across different domains.

6 Acknowledgment

We sincerely thank all the reviewers for their insightful comments and suggestions. This research is partially supported by National Natural Science Foundation of China (Grant No. 61773229 and 61972219), the Basic Research Fund of Shenzhen City (Grand No. JCYJ20190813165003837), and Overseas Cooperation Research Fund of Graduate School at Shenzhen, Tsinghua University (Grant No. HW2018002).

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *Proceedings of OSDI*, pages 265–283.

Deng Cai and Hai Zhao. 2016. Neural word segmentation learning for Chinese. In *Proceedings of ACL*, pages 409–420.

Deng Cai, Hai Zhao, Zhisong Zhang, Yuan Xin, Yongjian Wu, and Feiyue Huang. 2017. Fast and accurate neural word segmentation for Chinese. In *Proceedings of ACL*, volume 2, pages 608–615.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015. Long short-term memory neural networks for Chinese word segmentation. In *Proceedings of EMNLP*, pages 1197–1206.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-criteria learning for Chinese word segmentation. In *Proceedings of ACL*, volume 1, pages 1193–1203.

Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier. 2017. Language modeling with gated convolutional networks. In *Proceedings of ICML*, pages 933–941.

Thomas Emerson. 2005. The second international Chinese word segmentation bakeoff. In *Proceedings of SIGHAN workshop*.

Youmna Farag, Helen Yannakoudakis, and Ted Briscoe. 2018. Neural automated essay scoring and coherence modeling for adversarially crafted input. In *Proceedings of NAACL*, pages 263–271.

Leilei Gan and Yue Zhang. 2019. Investigating self-attention network for Chinese word segmentation. *arXiv preprint arXiv:1907.11512*.

Jianfeng Gao, Mu Li, Chang-Ning Huang, and Andi Wu. 2005. Chinese word segmentation and named entity recognition: A pragmatic approach. *Computational Linguistics*, 31(4):531–574.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of ICML*, pages 1243–1252.

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

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Lifu Huang, Heng Ji, and Jonathan May. 2019a. Cross-lingual multi-level adversarial transfer to enhance low-resource name tagging. In *Proceedings of NAACL*, pages 3823–3833.

Weipeng Huang, Xingyi Cheng, Kunlong Chen, Taifeng Wang, and Wei Chu. 2019b. Toward fast and accurate neural Chinese word segmentation with multi-criteria learning. *arXiv preprint arXiv:1903.04190*.

Edwin T Jaynes. 1957. Information theory and statistical mechanics. *Physical review*, 106(4):620.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. In *Proceedings of EMNLP*, pages 2021–2031.

Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. 2017. Cross-lingual transfer learning for pos tagging without cross-lingual resources. In *Proceedings of EMNLP*, pages 2832–2838.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*, pages 1746–1751.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. 2004. Estimating mutual information. *Physical review E*, 69(6):066138.

Zhongyang Li, Xiao Ding, and Ting Liu. 2018. Generating reasonable and diversified story ending using sequence to sequence model with adversarial training. In *Proceedings of COLING*, pages 1033–1043.

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

Yijia Liu, Wanxiang Che, Jiang Guo, Bing Qin, and Ting Liu. 2016. Exploring segment representations for neural segmentation models. In *Proceedings of IJCAI*, pages 2880–2886.

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 EMNLP*, pages 864–874.

Ji Ma, Kuzman Ganchev, and David Weiss. 2018. State-of-the-art Chinese word segmentation with Bi-LSTMs. In *Proceedings of EMNLP*, pages 4902–4908.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of CICLING*, page 562.

Likun Qiu and Yue Zhang. 2015. Word segmentation for Chinese novels. In *Proceedings of AAAI*, pages 2440–2446.

Andrew Viterbi. 1967. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.

Chunqi Wang and Bo Xu. 2017. Convolutional neural network with word embeddings for Chinese word segmentation. In *Proceedings of IJCNLP*, volume 1, pages 163–172.

Pak-kwong Wong and Chorkin Chan. 1996. Chinese word segmentation based on maximum matching and word binding force. In *Proceedings of COLING*, pages 200–203.

Jie Yang, Yue Zhang, and Fei Dong. 2017. Neural word segmentation with rich pretraining. In *Proceedings of ACL*, volume 1, pages 839–849.

Jie Yang, Yue Zhang, and Shuailong Liang. 2019. Subword encoding in lattice lstm for Chinese word segmentation. In *Proceedings of NAACL*, pages 2720–2725.

Yuxiao Ye, Weikang Li, Yue Zhang, Likun Qiu, and Jian Sun. 2019. Improving cross-domain Chinese word segmentation with word embeddings. In *Proceedings of NACCL*, pages 2726–2735.

Qi Zhang, Xiaoyu Liu, and Jinlan Fu. 2018. Neural networks incorporating dictionaries for Chinese word segmentation. In *Proceedings of AAAI*, pages 5682–5689.

Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2010. A unified character-based tagging framework for Chinese word segmentation. *TALIP*, 9(2):1–32.

Lujun Zhao, Qi Zhang, Peng Wang, and Xiaoyu Liu. 2018. Neural networks incorporating unlabeled and partially-labeled data for cross-domain Chinese word segmentation. In *Proceedings of IJCAI*, pages 4602–4608.

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 EMNLP*, pages 771–777.

Joey Tianyi Zhou, Hao Zhang, Di Jin, Hongyuan Zhu, Meng Fang, Rick Siow Mong Goh, and Kenneth Kwok. 2019. Dual adversarial neural transfer for low-resource named entity recognition. In *Proceedings of ACL*, pages 3461–3471.