Pre-training with Meta Learning for Chinese Word Segmentation

Zhen Ke[1], Liang Shi[1], Songtao Sun[1], Erli Meng[1], Bin Wang[1], Xipeng Qiu[2]
Xiaomi AI Lab, Xiaomi Inc., Beijing, China[1]
Shanghai Key Laboratory of Intelligent Information Processing, Fudan University[2]
{kezhen,shiliang1,sunsongtao,mengerli,wangbin11}@xiaomi.com
{xpqiu}@fudan.edu.cn

Abstract

Recent researches show that pre-trained models (PTMs) are beneficial to Chinese Word Segmentation (CWS). However, PTMs used in previous works usually adopt language modeling as pre-training tasks, lacking task-specific prior segmentation knowledge and ignoring the discrepancy between pre-training tasks and downstream CWS tasks. In this paper, we propose a CWS-specific pre-trained model METASEG, which employs a unified architecture and incorporates meta learning algorithm into a multi-criteria pre-training task. Empirical results show that METASEG could utilize common prior segmentation knowledge from different existing criteria and alleviate the discrepancy between pre-trained models and downstream CWS tasks. Besides, METASEG can achieve new state-of-the-art performance on twelve widely-used CWS datasets and significantly improve model performance in low-resource settings.

1 Introduction

Chinese Word Segmentation (CWS) is a fundamental task for Chinese natural language processing (NLP), which aims at identifying word boundaries in a sentence composed of continuous Chinese characters. It provides a basic component for other NLP tasks like named entity recognition (Li et al., 2020), dependency parsing (Yan et al., 2020), and semantic role labeling (Xia et al., 2019), etc.

Generally, most previous studies model the CWS task as a character-based sequence labeling task (Xue, 2003; Zheng et al., 2013; Chen et al., 2015; Ma et al., 2018; Qiu et al., 2020). Recently, pre-trained models (PTMs) such as BERT (Devlin et al., 2019) have been introduced into CWS tasks, which could provide prior semantic knowledge and boost the performance of CWS systems. Yang (2019) directly fine-tunes BERT on several CWS benchmark datasets. Huang et al. (2020) fine-tunes BERT in a multi-criteria learning framework, where each criterion shares a common BERT-based feature extraction layer and has separate projection layer. Meng et al. (2019) combines Chinese character glyph features with pre-trained BERT representations. Tian et al. (2020) proposes a neural CWS framework WMSEG, which utilizes memory networks to incorporate wordhood information into the pre-trained model ZEN (Diao et al., 2019).

Criteria	Li	Na	entered	the semi-final	
CTB6	李娜		进入	半决赛	
PKU	李	娜	进入	半	决赛
MSRA	李娜		进入	半	决赛

Table 1: An example of CWS on different criteria.

PTMs have been proved quite effective by fine-tuning on downstream CWS tasks. However, PTMs used in previous works usually adopt language modeling as pre-training tasks. Thus, they usually lack task-specific prior knowledge for CWS and ignore the discrepancy between pre-training tasks and downstream CWS tasks.

To deal with aforementioned problems of PTMs, we consider introducing a CWS-specific pre-trained model based on existing CWS corpora, to leverage the prior segmentation knowledge. However, there are multiple inconsistent segmentation criteria for CWS, where each criterion represents a unique style of segmenting Chinese sentence into words, as shown in Table 1. Meanwhile, we can easily observe that different segmentation criteria could share a large proportion of word boundaries between them, such as the boundaries between word units "李娜(Li Na)", "进入(entered)" and "半决赛(the semi-final)", which are the same for all segmentation criteria. It shows that the common prior segmentation knowledge is shared by different criteria.

In this paper, we propose a CWS-specific pre-trained model METASEG. To leverage shared

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 5514–5523
June 6–11, 2021. ©2021 Association for Computational Linguistics

segmentation knowledge of different criteria, METASEG utilizes a unified architecture and introduces a multi-criteria pre-training task. Moreover, to alleviate the discrepancy between pre-trained models and downstream unseen criteria, meta learning algorithm (Finn et al., 2017) is incorporated into the multi-criteria pre-training task of METASEG.

Experiments show that METASEG could outperform previous works significantly, and achieve new state-of-the-art results on twelve CWS datasets. Further experiments show that METASEG has better generalization performance on downstream unseen CWS tasks in low-resource settings, and improve recalls for Out-Of-Vocabulary (OOV) words. To the best of our knowledge, METASEG is the first task-specific pre-trained model especially designed for CWS.

2 Related Work

Recently, PTMs have been used for CWS and achieve good performance (Devlin et al., 2019). These PTMs usually exploit fine-tuning as the main way of transferring prior knowledge to downstream CWS tasks. Specifically, some methods directly fine-tune PTMs on CWS tasks (Yang, 2019), while others fine-tune them in a multi-task framework (Huang et al., 2020). Besides, other features are also incorporated into PTMs and fine-tuned jointly, including Chinese glyph features (Meng et al., 2019), wordhood features (Tian et al., 2020), and so on. Although PTMs improve CWS systems significantly, their pre-training tasks like language modeling still have a wide discrepancy with downstream CWS tasks and lack CWS-specific prior knowledge.

Task-specific pre-trained models are lately studied to introduce task-specific prior knowledge into multiple NLP tasks. Specifically designed pre-training tasks are introduced to obtain the task-specific pre-trained models, and then these models are fine-tuned on corresponding downstream NLP tasks, such as named entity recognition (Xue et al., 2020), sentiment analysis (Ke et al., 2020) and text summarization (Zhang et al., 2020). In this paper, we propose a CWS-specific pre-trained model METASEG.

3 Approach

As other task-specific pre-trained models (Ke et al., 2020), the pipeline of METASEG is divided into two phases: pre-training phase and fine-tuning phase. In pre-training phase, we design a unified architecture and incorporate meta learning algorithm into a multi-criteria pre-training task, to obtain the CWS-specific pre-trained model which has less discrepancy with downstream CWS tasks. In fine-tuning phase, we fine-tune the pre-trained model on downstream CWS tasks, to leverage the prior knowledge learned in pre-training phase.

In this section, we will describe METASEG in three parts. First, we introduce the Transformer-based unified architecture. Second, we elaborate on the multi-criteria pre-training task with meta learning algorithm. Finally, we give a brief description of the downstream fine-tuning phase.

3.1 The Unified Architecture

In traditional CWS systems (Chen et al., 2015; Ma et al., 2018), CWS model usually adopts a separate architecture for each segmentation criterion. An instance of the CWS model is created for each criterion and trained on the corresponding dataset independently. Thus, a model instance can only serve one criterion, without sharing any segmentation knowledge with other different criteria.

To better leverage the common segmentation knowledge shared by multiple criteria, METASEG employs a unified architecture based on the widely-used Transformer network (Vaswani et al., 2017) with shared encoder and decoder for all different criteria, as illustrated in Figure 1.

The input for the unified architecture is an augmented sentence, which is composed of a specific criterion token plus the original sentence to represent both criterion and text information. In embedding layer, the augmented sentence is transformed into input representations by summing the token, segment and position embeddings. The Transformer network is used as the shared encoder layer, encoding the input representations into hidden representations through blocks of multi-head attention and position-wise feed-forward modules (Vaswani et al., 2017). Then a shared linear decoder with softmax is followed to map hidden representations to the probability distribution of segmentation labels. The segmentation labels consist of four CWS labels $\{B, M, E, S\}$, denoting the word beginning, middle, ending and single word respectively.

Formally, the unified architecture can be concluded as a probabilistic model $P_\theta(Y|X)$, which represents the probability of the segmentation label

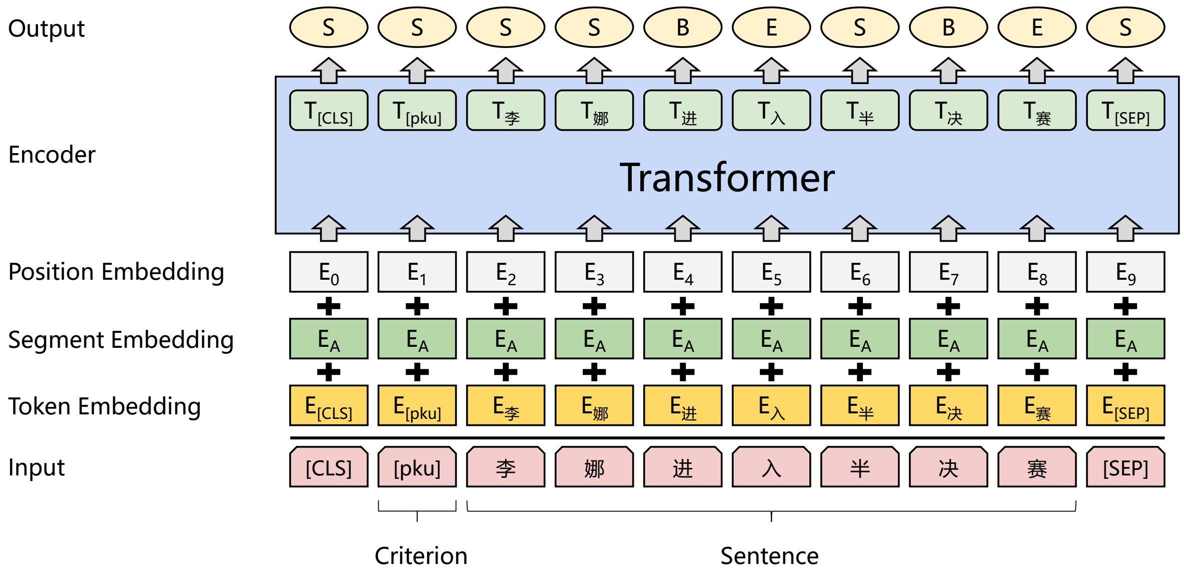


Figure 1: The unified framework of our proposed model, with shared encoder and decoder for different criteria. The input is composed of criterion and sentence, where the criterion can vary with the same sentence. The output is a corresponding sequence of segmentation labels of given criterion.

sequence Y given the augmented input sentence X. The model parameters θ are invariant of any criterion c, and would capture the common segmentation knowledge shared by different criteria.

3.2 Multi-Criteria Pre-training with Meta Learning

In this part, we describe multi-criteria pre-training with meta learning for METASEG. We construct a multi-criteria pre-training task, to fully mine the shared prior segmentation knowledge of different criteria. Meanwhile, to alleviate the discrepancy between pre-trained models and downstream CWS tasks, meta learning algorithm (Finn et al., 2017) is used for pre-training optimization of METASEG.

Multi-Criteria Pre-training Task As mentioned in Section 1, there are already a variety of existing CWS corpora (Emerson, 2005; Jin and Chen, 2008). These CWS corpora usually have inconsistent segmentation criteria, where human-annotated data is insufficient for each criterion. Each criterion is usually used to fine-tune a CWS model separately on a relatively small dataset and ignores the shared knowledge of different criteria. But in our multi-criteria pre-training task, multiple criteria are jointly used for pre-training to capture the common segmentation knowledge shared by different existing criteria.

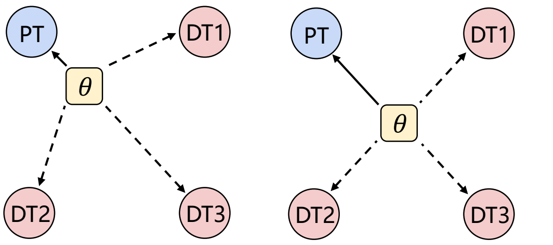
First, nine public CWS corpora (see Section 4.1) of diverse segmentation criteria are merged as a joint multi-criteria pre-training corpus D_T. Every sentence under each criterion is augmented with the corresponding criterion, and then incorporated into the joint multi-criteria pre-training corpus. To represent criterion information, we add a specific criterion token in front of the input sentence, such as [pku] for PKU criterion (Emerson, 2005). We also add [CLS] and [SEP] token to sentence beginning and ending respectively like Devlin et al. (2019). This augmented input sentence represents both criterion and text information, as shown in Figure 1.

Then, we randomly pick 10% sentences from the joint multi-criteria pre-training corpus D_T and replace their criterion tokens with a special token [unc], which means undefined criterion. With this design, the undefined criterion token [unc] would learn criterion-independent segmentation knowledge and help to transfer such knowledge to downstream CWS tasks.

Finally, given a pair of augmented sentence X and segmentation labels Y from the joint multi-criteria pre-training corpus D_T, our unified architecture (Section 3.1) predicts the the probability of segmentation labels $P_\theta(Y|X)$. We use the normal negative log-likelihood (NLL) loss as objective function for this multi-criteria pre-training task:

$$L(\theta; D_T) = -\sum_{X,Y\in D_T} \log P_\theta(Y|X) \qquad (1)$$

Meta Learning Algorithm The objective of most PTMs is to maximize its performance on pre-training tasks (Devlin et al., 2019), which would lead to the discrepancy between pre-trained models and downstream tasks. Besides, pre-trained CWS model from multi-criteria pre-training task could still have discrepancy with downstream unseen criteria, because downstream criteria may not exist in pre-training. To alleviate the above discrepancy, we utilize meta learning algorithm (Lv et al., 2020) for pre-training optimization of METASEG. The main objective of meta learning is to maximize generalization performance on potential downstream tasks, which prevents pre-trained models from over-fitting on pre-training tasks. As shown in Figure 2, by introducing meta learning algorithm, pre-trained models would have less discrepancy with downstream tasks instead of inclining towards pre-training tasks.



(a) Pre-training without meta learning

(b) Pre-training with meta learning

Figure 2: Pre-training with and without meta learning. PT represents the multi-criteria pre-training task, while solid line represents the pre-training phase. DT represents the downstream CWS task, while dashed line represents the fine-tuning phase. θ represents pre-trained model parameters.

The meta learning algorithm treats pre-training task T as one of the downstream tasks. It tries to optimize meta parameters θ_0, from which we can get the task-specific model parameters θ_k by k gradient descent steps over the training data D_T^{train} on task T,

$$\begin{aligned}
&\theta_1 = \theta_0 - \alpha \nabla_{\theta_0} L_T(\theta_0; D_{T,1}^{train}),\\
&\dots,\\
&\theta_k = \theta_{k-1} - \alpha \nabla_{\theta_{k-1}} L_T(\theta_{k-1}; D_{T,k}^{train}),
\end{aligned} \tag{2}$$

where α is learning rate, $D_{T,i}^{train}$ is the i-th batch of training data. Formally, task-specific parameters θ_k can be denoted as a function of meta parameters θ_0 as follows: $\theta_k = f_k(\theta_0)$.

To maximize the generalization performance on task T, we should optimize meta parameters θ_0 on the batch of test data D_T^{test},

$$\begin{aligned}
\theta_0^* &= \arg\min_{\theta_0} L_T(\theta_k; D_T^{test})\\
&= \arg\min_{\theta_0} L_T(f_k(\theta_0); D_T^{test}).
\end{aligned} \tag{3}$$

The above meta optimization could be achieved by gradient descent, so the update rule for meta parameters θ_0 is as follows:

$$\theta_0' = \theta_0 - \beta \nabla_{\theta_0} L_T(\theta_k; D_T^{test}), \tag{4}$$

where β is the meta learning rate. The gradient in Equation 4 can be rewritten as:

$$\begin{aligned}
&\nabla_{\theta_0} L_T(\theta_k; D_T^{test})\\
&= \nabla_{\theta_k} L_T(\theta_k; D_T^{test}) \times \nabla_{\theta_{k-1}} \theta_k \times \cdots \nabla_{\theta_0} \theta_1\\
&= \nabla_{\theta_k} L_T(\theta_k; D_T^{test}) \prod_{j=1}^{k} (I - \alpha \nabla^2_{\theta_{j-1}} L_T(\theta_{j-1}; D_{T,j}^{train}))\\
&\approx \nabla_{\theta_k} L_T(\theta_k; D_T^{test}),
\end{aligned} \tag{5}$$

where the last step in Equation 5 adopts first-order approximation for computational simplification (Finn et al., 2017).

Specifically, the meta learning algorithm for pre-training optimization is described in Algorithm 1. It can be divided into two stages: i) meta train stage, which updates task-specific parameters by k gradient descent steps over training data; ii) meta test stage, which updates meta parameters by one gradient descent step over test data. Hyper-parameter k is the number of gradient descent steps in meta train stage. The meta learning algorithm degrades to normal gradient descent algorithm when $k = 0$. The returned meta parameters θ_0 are used as the pre-trained model parameters for METASEG.

3.3 Downstream Fine-tuning

After pre-training phase mentioned in Section 3.2, we obtain the pre-trained model parameters θ_0, which capture prior segmentation knowledge and have less discrepancy with downstream CWS tasks. We fine-tune these pre-trained parameters θ_0 on downstream CWS corpus, to transfer the prior segmentation knowledge.

For format consistency, we process the sentence from the given downstream corpus in the same way as Section 3.2, by adding the criterion token `[unc]`, beginning token `[CLS]` and ending token beginning token `[SEP]`. The undefined criterion token `[unc]` is used in fine-tuning phase

Algorithm 1 Meta Learning for Pre-training Optimization

```
Require: Distribution over pre-training task $p(T)$, initial meta parameters $\theta_0$, objective function $L$
Require: Learning rate $\alpha$, meta learning rate $\beta$, meta train steps $k$
for $epoch = 1, 2, ...$ do
    Sample k training data batches $D_T^{train}$ from $p(T)$
    for $j = 1, 2, ..., k$ do
        $\theta_j \leftarrow \theta_{j-1} - \alpha \nabla_{\theta_{j-1}} L_T(\theta_{j-1}; D_{T,j}^{train})$
    end for
    Sample test data batch $D_T^{test}$ from $p(T)$
    $\theta_0 \leftarrow \theta_0 - \beta \nabla_{\theta_k} L_T(\theta_k; D_T^{test})$
end for
return Meta parameters $\theta_0$
```

instead of the downstream criterion itself, because the downstream criterion usually doesn't exist in pre-training phase and the pre-trained model has no information about it.

4 Experiment

4.1 Experimental Settings

Datasets We collect twelve publicly available CWS datasets, with each dataset representing a unique segmentation criterion. Among all datasets, we have PKU, MSRA, CITYU, AS from SIGHAN2005 (Emerson, 2005), CKIP, NCC, SXU from SIGHAN2008 (Jin and Chen, 2008), CTB6 from Xue et al. (2005), WTB from Wang et al. (2014), UD from Zeman et al. (2017), ZX from Zhang et al. (2014) and CNC [1].

WTB, UD, ZX datasets are kept for downstream fine-tuning phase, while the other nine datasets are combined into the joint multi-criteria pre-training corpus (Section 3.2), which amounts to nearly 18M words.

For CTB6, WTB, UD, ZX and CNC datasets, we use the official data split of training, development, and test sets. For the rest, we use the official test set and randomly pick 10% samples from the training data as the development set. We pre-process all these datasets following four procedures:

1. Convert traditional Chinese datasets into simplified, such as CITYU, AS and CKIP;
2. Convert full-width tokens into half-width;
3. Replace continuous English letters and digits with unique tokens;
4. Split sentences into shorter clauses by punctuation.

Table 2 presents the statistics of processed datasets.

Hyper-Parameters We employ METASEG with the same architecture as BERT-Base (Devlin et al., 2019), which has 12 transformer layers, 768 hidden sizes and 12 attention heads.

In pre-training phase, METASEG is initialized with released parameters of Chinese BERT-Base model [2] and then pre-trained with the multi-criteria pre-training task. Maximum input length is 64, with batch size 64, and dropout rate 0.1. We adopt AdamW optimizer (Loshchilov and Hutter, 2019) with $\beta_1 = 0.9, \beta_2 = 0.999$ and weight decay rate of 0.01. The optimizer is implemented by meta learning algorithm, where both learning rate α and meta learning rate β are set to 2e-5 with a linear warm-up proportion of 0.1. The meta train steps are selected to $k = 1$ according to downstream performance. Pre-training process runs for nearly 127,000 meta test steps, amounting to $(k + 1) * 127,000$ gradient descent steps, which takes about 21 hours on one NVIDIA Tesla V100 32GB GPU card.

In fine-tuning phase, we set maximum input length to 64 for all criteria but 128 for WTB, with batch size 64. We fine-tune METASEG with AdamW optimizer of the same settings as pre-training phase without meta learning. METASEG is fine-tuned for 5 epochs on each downstream dataset.

In low-resource settings, experiments are performed on WTB dataset, with maximum input length 128. We evaluate METASEG at sampling rates of 1%, 5%, 10%, 20%, 50%, 80%. Batch size is 1 for 1% sampling and 8 for the rest. We keep other hyper-parameters the same as those of fine-tuning phase.

The standard F1 score is used to evaluate the performance of all models. We report F1 score of each model on the test set according to its best checkpoint on the development set as Qiu et al. (2020).

[1] http://corpus.zhonghuayuwen.org/

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

Corpus	#Train Words	#Dev Words	#Test Words	OOV Rate	Avg. Length
PKU	999,823	110,124	104,372	3.30%	10.6
MSRA	2,133,674	234,717	106,873	2.11%	11.3
CITYU	1,308,774	146,856	40,936	6.36%	11.0
AS	4,902,887	546,694	122,610	3.75%	9.7
CKIP	649,215	72,334	90,678	7.12%	10.5
NCC	823,948	89,898	152,367	4.82%	10.0
SXU	475,489	52,749	113,527	4.81%	11.1
CTB6	678,811	51,229	52,861	5.17%	12.5
CNC	5,841,239	727,765	726,029	0.75%	9.8
WTB	14,774	1,843	1,860	15.05%	28.2
UD	98,607	12,663	12,012	11.04%	11.4
ZX	67,648	20,393	67,648	6.48%	8.2

Table 2: Statistics of datasets. The first block corresponds to the pre-training criteria. The second block corresponds to downstream criteria, which are unseen in pre-training phase.

4.2 Overall Results

4.2.1 Results on Pre-training Criteria

After pre-training, we fine-tune METASEG on each pre-training criterion. Table 3 shows F1 scores on test sets of nine pre-training criteria in two blocks. The first block displays the performance of previous works. The second block displays three models implemented by us: **BERT-Base** is the fine-tuned model initialized with official BERT-Base parameters. **METASEG (w/o fine-tune)** is our proposed pre-trained model directly used for inference without fine-tuning. **METASEG** is the fine-tuned model initialized with pre-trained METASEG parameters.

From the second block, we observe that fine-tuned METASEG could outperform fine-tuned BERT-Base on each criterion, with 0.26% improvement on average. It shows that METASEG is more effective when fine-tuned for CWS. Even without fine-tuning, METASEG (w/o fine-tune) still behaves better than fine-tuned BERT-Base model, indicating that our proposed pre-training approach is the key factor for the effectiveness of METASEG. Fine-tuned METASEG performs better than that of no fine-tuning, showing that downstream fine-tuning is still necessary for the specific criterion. Furthermore, METASEG can achieve state-of-the-art results on eight of nine pre-training criteria, demonstrating the effectiveness of our proposed methods.

4.2.2 Results on Downstream Criteria

To evaluate the knowledge transfer ability of METASEG, we perform experiments on three unseen downstream criteria which are absent in pre-training phase. Table 4 shows F1 scores on test sets of three downstream criteria. The first block displays previous works on these downstream criteria, while the second block displays three models implemented by us (see Section 4.2.1 for details).

Results show that METASEG outperforms the previous best model by 0.56% on average, achieving new state-of-the-art performance on three downstream criteria. Moreover, METASEG (w/o fine-tune) actually preforms zero-shot inference on downstream criteria and still achieves 87.28% average F1 score. This shows that METASEG does learn some common prior segmentation knowledge in pre-training phase, even if it doesn't see these downstream criteria before.

Compared with BERT-Base, METASEG has the same architecture but different pre-training tasks. It can be easily observed that METASEG with fine-tuning outperforms BERT-Base by 0.46% on average. This indicates that METASEG could indeed alleviate the discrepancy between pre-trained models and downstream CWS tasks than BERT-Base.

4.2.3 Ablation Studies

We perform further ablation studies on the effects of meta learning (ML) and multi-criteria pre-training (MP), by removing them consecutively from the complete METASEG model. After removing both of them, METASEG degrades into the normal BERT-Base model. F1 scores for ablation studies on three downstream criteria are illustrated in Table 5.

We observe that the average F1 score drops by 0.12% when removing the meta learning algorithm (-ML), and continues to drop by 0.34% when removing the multi-criteria pre-training task (-ML-MP). It demonstrates that meta learning and multi-criteria pre-training are both significant for the effectiveness of METASEG.

Models	PKU	MSRA	CITYU	AS	CKIP	NCC	SXU	CTB6	CNC	Avg.
Chen et al. (2017)	94.32	96.04	95.55	94.64	94.26	92.83	96.04	-	-	-
Ma et al. (2018)	96.10	97.40	97.20	96.20	-	-	-	96.70	-	-
He et al. (2019)	95.78	97.35	95.60	95.47	95.73	94.34	96.49	-	-	-
Gong et al. (2019)	96.15	97.78	96.22	95.22	94.99	94.12	97.25	-	-	-
Yang et al. (2019)	95.80	97.80	-	-	-	-	-	96.10	-	-
Meng et al. (2019)	96.70	98.30	97.90	96.70	-	-	-	-	-	-
Yang (2019)	96.50	98.40	-	-	-	-	-	-	-	-
Duan and Zhao (2020)	95.50	97.70	96.40	95.70	-	-	-	-	-	-
Huang et al. (2020)	**97.30**	98.50	97.80	97.00	-	-	97.50	97.80	97.30	-
Qiu et al. (2020)	96.41	98.05	96.91	96.44	96.51	96.04	97.61	-	-	-
Tian et al. (2020)	96.53	98.40	97.93	96.62	-	-	-	97.25	-	-
BERT-Base (ours)	96.72	98.25	98.19	96.93	96.49	96.13	97.61	97.85	97.45	97.29
MetaSeg (w/o fine-tune)	96.76	98.02	98.12	**97.04**	**96.81**	97.21	97.51	97.87	97.25	97.40
MetaSeg	96.92	**98.50**	**98.20**	97.01	96.72	**97.24**	**97.88**	**97.89**	**97.55**	**97.55**

Table 3: F1 scores on test sets of pre-training criteria. The first block displays results from previous works. The second block displays three models implemented by us.

Models	WTB	UD	ZX	Avg.
Ma et al. (2018)	-	96.90	-	-
Huang et al. (2020)	93.20	97.80	97.10	96.03
BERT-Base (ours)	93.00	98.32	97.06	96.13
MetaSeg (w/o fine-tune)	89.53	83.84	88.48	87.28
MetaSeg	**93.97**	**98.57**	**97.22**	**96.59**

Table 4: F1 scores on test sets of downstream criteria.

Models	WTB	UD	ZX	Avg.
MetaSeg	**93.97**	**98.57**	**97.22**	**96.59**
-ML	93.71	98.49	97.22	96.47
-ML-MP	93.00	98.32	97.06	96.13

Table 5: F1 scores for ablation studies on downstream criteria. -ML indicates MetaSeg without meta learning. -ML-MP indicates MetaSeg without meta learning and multi-criteria pre-training.

4.3 Discussion

4.3.1 Low-Resource Settings

To better explore the downstream generalization ability of MetaSeg, we perform experiments on the downstream WTB criterion in low-resource settings. Specifically, we randomly sample a given rate of instances from the training set and fine-tune the pre-trained MetaSeg model on down-sampling training sets. These settings imitate the realistic low-resource circumstance where human-annotated data is insufficient.

The performance at different sampling rates is evaluated on the same WTB test set and reported in Table 6. Results show that MetaSeg outperforms BERT-Base at every sampling rate. The margin is larger when the sampling rate is lower, and reaches 6.20% at 1% sampling rate. This demonstrates that MetaSeg could generalize better on the downstream criterion in low-resource settings.

When the sampling rate drops from 100% to 1%, F1 score of BERT-Base decreases by 7.60% while that of MetaSeg only decreases by 2.37%. The performance of MetaSeg at 1% sampling rate still reaches 91.60% with only 8 instances, comparable with performance of BERT-Base at 20% sampling rate. This indicates that MetaSeg can make better use of prior segmentation knowledge and learn from less amount of data. It shows that MetaSeg would reduce the need of human annotation significantly.

4.3.2 Out-of-Vocabulary Words

Out-of-Vocabulary (OOV) words denote the words which exist in inference phase but don't exist in training phase. OOV words are a critical cause of errors on CWS tasks. We evaluate recalls for OOV words on test sets of all twelve criteria in Table 7.

Results show that MetaSeg outperforms BERT-Base on ten of twelve criteria and improves recalls for OOV words by 0.99% on average. This indicates that MetaSeg could benefit from our proposed pre-training methodology and recognize more OOV words in inference phase.

4.3.3 Non-Pretraining Setup

To investigate the contribution of multi-criteria pre-training towards performance of MetaSeg, we perform experiments on a non-pretraining baseline **Transformer**. Transformer has the same architecture and is directly trained from scratch on the same nine datasets (Section 4.2.1), but doesn't have any pre-training phase as MetaSeg. Comparison of

Sampling Rates	1%	5%	10%	20%	50%	80%	100%
#Instances	8	40	81	162	406	650	813
BERT-Base (ours)	85.40	87.83	90.46	91.15	92.80	93.14	93.00
METASEG	**91.60**	**92.29**	**92.54**	**92.63**	**93.45**	**94.11**	**93.97**

Table 6: F1 scores on WTB test set in low-resource settings.

Models	**PKU**	**MSRA**	**CITYU**	**AS**	**CKIP**	**NCC**	**SXU**	**CTB6**	**CNC**	**WTB**	**UD**	**ZX**	**Avg.**
BERT-Base	80.15	81.03	90.62	79.60	**84.48**	79.64	84.75	89.10	61.18	83.57	93.36	**87.69**	82.93
METASEG	**80.90**	**83.03**	**90.66**	**80.89**	84.42	**84.14**	**85.98**	**89.21**	**61.90**	**85.00**	**93.59**	87.33	**83.92**

Table 7: Recalls for OOV words on test sets of all twelve criteria.

F1 scores between Transformer and METASEG is displayed in Table 8.

Results show that METASEG outperforms the non-pretraining Transformer on each criterion and achieves a 2.40% gain on average, even with the same datasets and architecture. It demonstrates that multi-criteria pre-training is vital for the effectiveness of METASEG and the performance gain is not merely from the large dataset size.

Moreover, METASEG has the generalization ability to transfer prior knowledge to downstream unseen criteria, which could not be achieved by the non-pretraining counterpart Transformer.

4.3.4 Visualization

To visualize the discrepancy between pre-trained models and downstream criteria, we plot similarities of three downstream criteria with METASEG and BERT.

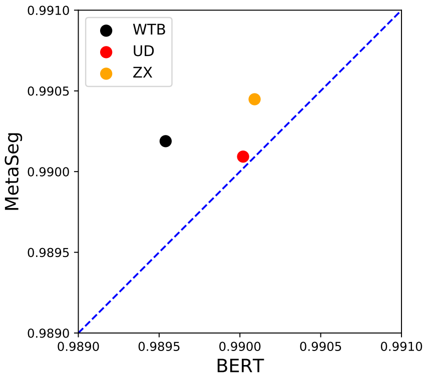


Figure 3: Cosine similarities between three downstream criteria and two pre-trained models. The dashed line indicates the positions where one criterion has equal similarities with two pre-trained models.

Specifically, we extract the criterion token embeddings of three downstream criteria WTB, UD and ZX. We also extract the undefined criterion token embeddings of METASEG and BERT as representations of these two pre-trained models. We compute cosine similarities between three criteria embeddings and two pre-trained model embeddings, and illustrate them in Figure 3.

We can observe that similarities of all three downstream criteria lie above the dashed line, indicating that all three downstream criteria are more similar to METASEG than BERT. The closer one criterion is to the upper left corner, the more similar it is to METASEG. Therefore, we can conclude that WTB is the most similar criterion to METASEG among all these criteria, which qualitatively corresponds to the phenomenon that WTB criterion has the largest performance gain in Table 4. The above visualization results show that our proposed approach could solidly alleviate the discrepancy between pre-trained models and downstream CWS tasks. Thus METASEG is more similar to downstream criteria.

5 Conclusion

In this paper, we propose a CWS-specific pre-trained model METASEG, which employs a unified architecture and incorporates meta learning algorithm into a multi-criteria pre-training task. Experiments show that METASEG could make good use of common prior segmentation knowledge from different existing criteria, and alleviate the discrepancy between pre-trained models and downstream CWS tasks. METASEG also gives better generalization ability in low-resource settings, and achieves new state-of-the-art performance on twelve CWS datasets.

Since the discrepancy between pre-training tasks and downstream tasks also exists in other NLP tasks and other languages, in the future we will explore whether the approach of pre-training with

Models	PKU	MSRA	CITYU	AS	CKIP	NCC	SXU	CTB6	CNC	Avg.
Transformer	95.33	94.79	95.36	95.22	95.17	93.90	95.66	96.45	94.51	95.15
METASEG	**96.92**	**98.50**	**98.20**	**97.01**	**96.72**	**97.24**	**97.88**	**97.89**	**97.55**	**97.55**

Table 8: Comparison of F1 scores on test sets of nine criteria between non-pretraining baseline Transformer and METASEG.

meta-learning in this paper could be applied to other tasks and languages apart from Chinese word segmentation.

References

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 the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1197–1206, Lisbon, Portugal. Association for Computational Linguistics.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial Multi-Criteria Learning for Chinese Word Segmentation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1193–1203, Vancouver, Canada. Association for Computational Linguistics.

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 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. 2019. ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations. *ArXiv*, abs/1911.00720.

Sufeng Duan and Hai Zhao. 2020. Attention Is All You Need for Chinese Word Segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Thomas Emerson. 2005. The Second International Chinese Word Segmentation Bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1126–1135, International Convention Centre, Sydney, Australia. PMLR.

Jingjing Gong, Xinchi Chen, Tao Gui, and Xipeng Qiu. 2019. Switch-LSTMs for Multi-Criteria Chinese Word Segmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*.

Han He, Lei Wu, Hua Yan, Zhimin Gao, Yi Feng, and George Townsend. 2019. Effective Neural Solution for Multi-criteria Word Segmentation. In *Smart Intelligent Computing and Applications*, pages 133–142, Singapore. Springer Singapore.

Weipeng Huang, Xingyi Cheng, Kunlong Chen, Taifeng Wang, and Wei Chu. 2020. Towards Fast and Accurate Neural Chinese Word Segmentation with Multi-Criteria Learning. In *Proceedings of the 38th International Conference on Computational Linguistics*.

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*.

Pei Ke, Haozhe Ji, Siyang Liu, Xiaoyan Zhu, and Minlie Huang. 2020. SentiLARE: Sentiment-Aware Language Representation Learning with Linguistic Knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. FLAT: Chinese NER Using Flat-Lattice Transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6836–6842. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Shangwen Lv, Yuechen Wang, Daya Guo, Duyu Tang, Nan Duan, Fuqing Zhu, Ming Gong, Linjun Shou, Ryan Ma, Daxin Jiang, Guihong Cao, Ming Zhou, and Songlin Hu. 2020. Pre-training Text Representations as Meta Learning. *ArXiv*, abs/2004.05568.

Ji Ma, Kuzman Ganchev, and David Weiss. 2018. State-of-the-art Chinese Word Segmentation with Bi-LSTMs. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4902–4908, Brussels, Belgium. Association for Computational Linguistics.

Yuxian Meng, Wei Wu, Fei Wang, Xiaoya Li, Ping Nie, Fan Yin, Muyu Li, Qinghong Han, Xiaofei Sun, and Jiwei Li. 2019. Glyce: Glyph-vectors for Chinese Character Representations. In *Advances in Neural Information Processing Systems 32*, pages 2746–2757. Curran Associates, Inc.

Xipeng Qiu, Hengzhi Pei, Hang Yan, and Xuanjing Huang. 2020. A Concise Model for Multi-Criteria Chinese Word Segmentation with Transformer Encoder. *EMNLP Findings*.

Yuanhe Tian, Yan Song, Fei Xia, Tong Zhang, and Yonggang Wang. 2020. Improving Chinese Word Segmentation with Wordhood Memory Networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8274–8285, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

William Yang Wang, Lingpeng Kong, Kathryn Mazaitis, and William W. Cohen. 2014. Dependency Parsing for Weibo: An Efficient Probabilistic Logic Programming Approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1152–1158, Doha, Qatar. Association for Computational Linguistics.

Qingrong Xia, Zhenghua Li, and Min Zhang. 2019. A Syntax-aware Multi-task Learning Framework for Chinese Semantic Role Labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5382–5392, Hong Kong, China. Association for Computational Linguistics.

Mengge Xue, Bowen Yu, Zhenyu Zhang, Tingwen Liu, Yue Zhang, and Bin Wang. 2020. Coarse-to-Fine Pre-training for Named Entity Recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*.

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*, page 207–238.

Nianwen Xue. 2003. Chinese Word Segmentation as Character Tagging. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 8, Number 1, February 2003: Special Issue on Word Formation and Chinese Language Processing*, pages 29–48.

Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020. A Graph-based Model for Joint Chinese Word Segmentation and Dependency Parsing. *Transactions of the Association for Computational Linguistics*, 8:78–92.

Haiqin Yang. 2019. BERT Meets Chinese Word Segmentation. *ArXiv*, abs/1909.09292.

Jie Yang, Yue Zhang, and Shuailong Liang. 2019. Subword Encoding in Lattice LSTM for Chinese Word Segmentation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2720–2725, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajič, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinková, Jan Hajič jr., Jaroslava Hlaváčová, Václava Kettnerová, Zdeňka Urešová, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria de Paiva, Kira Droganova, Héctor Martínez Alonso, Çağrı Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonça, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19, Vancouver, Canada. Association for Computational Linguistics.

Jingqing Zhang, Yao Zhao, Mohammad Saleh, and Peter J. Liu. 2020. PEGASUS: Pre-training with Extracted Gap-sentences for Abstractive Summarization. In *Proceedings of the 37th International Conference on Machine Learning*.

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 Conference of the European Chapter of the Association for Computational Linguistics*, pages 588–597, Gothenburg, Sweden. Association for Computational Linguistics.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep Learning for Chinese Word Segmentation and POS Tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 647–657, Seattle, Washington, USA. Association for Computational Linguistics.