Segment, Mask, and Predict: Augmenting Chinese Word Segmentation with Self-Supervision

Mieradilijiang Maimaiti[1,3,4], **Yang Liu**[*][1,2,3,4,5], **Yuanhang Zheng**[1,3,4], **Gang Chen**[1,3,4],
Kaiyu Huang[6], **Ji Zhang**[7], **Huanbo Luan**[1,3,4], **Maosong Sun**[1,3,4,5],
[1]Department of Computer Science and Technology, Tsinghua University, Beijing, China
[2]Institute for AI Industry Research, Tsinghua University, Beijing, China
[3]Institute for Artificial Intelligence, Tsinghua University, Beijing, China
[4]Beijing National Research Center for Information Science and Technology
[5]Beijing Academy of Artificial Intelligence
[6] School of Computer Science, Dalian University of Technology
[7]Alibaba DAMO Academy
{meadljmm15,zheng-yh19,cg18}@mails.tsinghua.edu.cn
{liuyang2011,sms}@tsinghua.edu.cn; luanhuanbo@gmail.com
kaiyuhuang@mail.dlut.edu.cn; zj122146@alibaba-inc.com

Abstract

Recent state-of-the-art (SOTA) effective neural network methods and fine-tuning methods based on pre-trained models (PTM) have been used in Chinese word segmentation (CWS), and they achieve great results. However, previous works focus on training the models with the fixed corpus at every iteration. The intermediate generated information is also valuable. Besides, the robustness of the previous neural methods is limited by the large-scale annotated data. There are a few noises in the annotated corpus. Limited efforts have been made by previous studies to deal with such problems. In this work, we propose a self-supervised CWS approach with a straightforward and effective architecture. First, we train a word segmentation model and use it to generate the segmentation results. Then, we use a revised masked language model (MLM) to evaluate the quality of the segmentation results based on the predictions of the MLM. Finally, we leverage the evaluations to aid the training of the segmenter by improved minimum risk training. Experimental results show that our approach outperforms previous methods on 9 different CWS datasets with single criterion training and multiple criteria training and achieves better robustness[1].

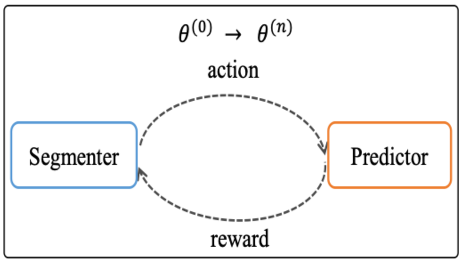


Figure 1: Word segmentation with self-supervision. Our work leverages the reward generated by the predictor to assist the training of the segmenter.

1 Introduction

In extensive natural language processing (NLP) scenarios, most of the tasks are based on word-level methods. When we deal with the Chinese language, there is no specific boundary between two Chinese words. The situation is different in western languages. For instance, there is a space between two words. Thus, Chinese word segmentation (CWS) is considered an essential task, which will accurately represent semantic information of Chinese NLP tasks. Besides, the length of the sentence is shortened by word segmentation. The shorter length of a sentence is effective for the deep learning method in some cases.

Recently, good performance for CWS has already been achieved in large-scale annotated corpora as reported by related research (Huang and Zhao, 2007; Zhao et al., 2019). Most methods start with data-driven to improve the performance for CWS. For instance, some neural methods try to incorporate external resources to achieve good performance for in-domain and cross-domain CWS (Zhou et al., 2017; Zhang et al., 2018). The previous methods fall into two categories: (1) the statistical machine learning methods and (2) neural network methods. In statistical machine learning meth-

[*]Corresponding author

[1]Code and dataset can be found at https://github.com/miradel51/Self_Supervised_CWS

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 2068–2077
November 7–11, 2021. ©2021 Association for Computational Linguistics

ods, Conditional Random Fields (CRF) is the most effective model for the sequence labeling problem (Zhao and Kit, 2008; Zhao et al., 2010). However, the performance of the CRF model depends on the quality of the hand-crafted features. To minimize the effects of different hand-crafted features, neural network methods (Chen et al., 2015b; Cai et al., 2017; Ma et al., 2018) have been widely used.

On the other hand, these supervised learning methods are usually limited by the training data. Recent SOTA approaches utilize the pre-trained models (PTM) to improve the quality of CWS (Tian et al., 2020; Huang et al., 2020). However, the CWS methods based on the PTM only utilize the large-scale annotated data to finetune the parameters. It omits much-generated information of the training step. Besides, the annotated data has some incorrect labels due to lexical diversity in Chinese, therefore the robustness of methods is quite important for the CWS.

In this work, we propose a self-supervised CWS approach to enhance the performance of CWS model. In addition, we also investigate on the cross-domain and low-quality datasets to analyze the robustness of CWS models. As depicted in Figure 1, our model consists of two parts: segmenter and predictor. We leverage the Transformer encoder as a word segmenter. We exploit the revised masked language model (MLM) as a predictor to improve the segmentation model. We generate masked sequences with respect to the segmentation results. Then we exploit MLM to predict the masked part and evaluate the quality of the segmentation based on the quality of the predictions. We leverage an improved version of minimum risk training (MRT) (Shen et al., 2016) to enhance the segmentation.

Our contributions are as follows:

- We propose a self-supervised method for CWS, which uses the predictions of revised MLM to assist the word segmentation model.
- We present an improved version of MRT by adding regularization terms to boost the performance of the word segmentation model.
- Experimental results show that our approach outperforms previous methods with different criteria training, and our proposed method also improves the robustness of the model.

2 Related Work

Chinese word segmentation (CWS) has been studied for several years as an essential Chinese NLP task. CWS methods are divided into two streams of approaches: word-based methods and character-based methods. Since Xue (2003) first formalizes the CWS task as a sequence labeling problem, almost all methods transfer the CWS results into the sequence labels. As a sequence labeling task, the CRF-based model can achieve a competitive performance with multiple features (Peng et al., 2004; Tseng et al., 2005; Zhao and Kit, 2008; Zhao et al., 2010). However, the effect of each method is determined by the quality of manual features. To reduce the influence of feature engineering, neural CWS methods have been studied and further progress has been made (Zheng et al., 2013; Pei et al., 2014; Chen et al., 2015a,b; Cai and Zhao, 2016; Chen et al., 2017; Cai et al., 2017). Neural methods gradually replaces traditional machine learning methods. Ma et al. (2018) propose the basic LSTM model that is the same with Chen et al. (2015b). But the former study could achieve SOTA performance through tuning the hyper-parameters. Some studies leverage the rich pre-trained embeddings to improve the performance for neural CWS methods (Zhou et al., 2017; Yang et al., 2017, 2019). To alleviate the issue of OOV words for CWS, some researches have been studied for cross-domain CWS. Zhang et al. (2018) incorporate the domain dictionary into the neural network, and Zhao et al. (2018) utilize the unlabeled data to enhance the ability to recognize OOV words. With the development of pre-trained language models (PLM) (Devlin et al., 2019), CWS methods also make further progress. Previous SOTA methods effectively achieve good performance for CWS (Meng et al., 2019; Huang et al., 2020; Duan and Zhao, 2020), and they take the advantages of PLMs rather than the pure models themselves. The redundant components get slight improvements that are not as much as the PLMs learning paradigm.

3 Method

The overall process of our method is shown in algorithm 1: First, we train a word segmentation model and use it to generate segmentation results. Then, according to the segmentation results, the masked sentence is generated based on certain strategies, and an MLM is trained with the masked sentence. Afterward, we mask the sentences in the training

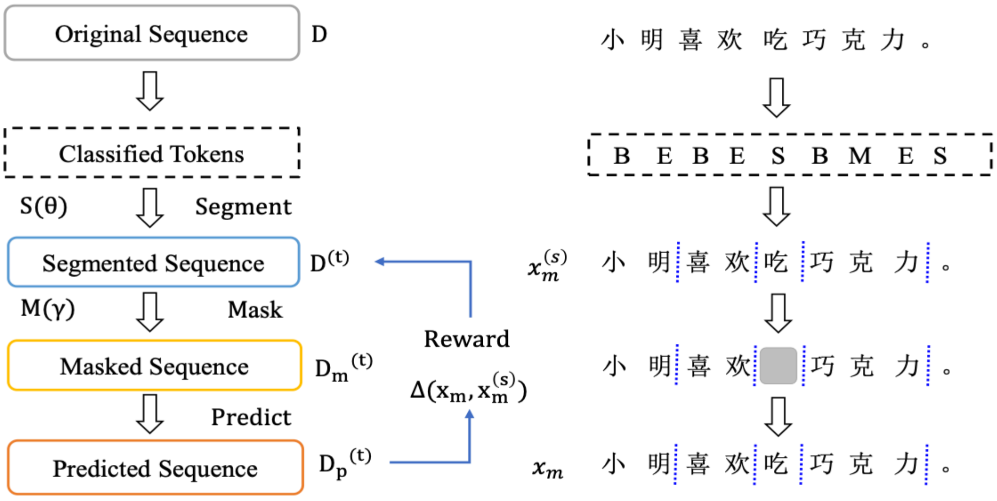


Figure 2: The architecture of our model. D, $D^{(t)}$ and $D_p^{(t)}$ represent the original sequence, segmented sequence and predicted tokens, while $S(\theta)$ and $M(\gamma)$ stand for segmentation model and revised mask prediction model respectively. $\Delta(x_m, x_m^{(s)})$ denotes the loss as a reward during predicting the masked tokens.

Algorithm 1 Self-supervised Word Segmentation

```
Input: Original sequence $D = \{\mathbf{x}^{(s)}\}_{s=1}^{S}$.
Output: Predicted sequence $D_p^{(t)}$.
Train Mask-Predictor $M(\gamma)$ based on $D$.
Train Segmenter $S(\theta^{(o)})$ based on $D$.
Employ $S(\theta^{(o)})$ to segment $D$ and achieve segmented sequence $D^{(t)}$.
Mask $D^{(t)}$ to obtain the masked sequence $D_m^{(t)}$ with the strategy.
Exploit $M(\gamma)$ to achieve predicted sequence $D_p^{(t)}$ based on $D^{(t)}$.
Calculate the accuracy by comparing $D_p^{(t)}$ and $D^{(t)}$ as a reward.
Update the $S(\theta^{(o)})$ to $S(\theta^{(n)})$.
```

set and predict the masked part using the MLM to evaluate the quality of the segmentation results. Finally, we use the results to aid the training of the segmentation model.

3.1 Segmentation Model

The model architecture is shown in Figure 2. Similar to the architecture of Huang et al. (2020) our segmentation model architecture is also based on BERT (Devlin et al., 2019). The input is a sentence with character-based tokenization and the output is generated by a BERT model and a CRF layer sequentially. The segmentation results are represented by four tags B, M, E, and S. B and E denote the beginning and end of a multi-character word, respectively. M denotes the middle part of a multi-character word, and S represents a single-character word. Our segmentation model is initialized with PTM (i.e. BERT) and trained with negative log-likelihood (NLL) loss.

3.2 Revised MLM as Predictor

In this work, we use a revised MLM similar to BERT (Devlin et al., 2019) to evaluate the quality of segmentations. However, the masking strategy adopted in the training of the Chinese BERT PTM makes the character a unit. This masking strategy cannot reflect the segmentation information, thus we design a new masking strategy that can reflect the segmentation information:

1. Only one character or multiple consecutive characters within a word can be masked simultaneously.

2. We set a threshold $mask\_count$. If the length of a word is less than or equal to $mask\_count$, the entire word will be masked. Otherwise, we randomly choose consecutive $mask\_count$ words and mask them.

3. From all possible maskings, we randomly select one with equal probability and apply it to the input.

Table 1 shows an example of the masking strategy we introduce above.

When evaluating the quality of segmentation results, we first find all the legal masked sequences

Segged Seq.	小明 喜欢 吃 巧克力 。
Masked Input	[M] [M] 喜 欢 吃 巧 克 力 。 小 明 [M] [M] 吃 巧 克 力 。 小 明 喜 欢 [M] 巧 克 力 。 小 明 喜 欢 吃 [M] [M] 力 。 小 明 喜 欢 吃 巧 [M] [M] 。 小 明 喜 欢 吃 巧 克 力 [M]

Table 1: Possible masked input examples of our masking strategy when $mask\_count = 2$. "Segged Seq." and "[M]" represent Segmented Sequence and Masked Token, respectively.

based on the segmentation result. Then, we use the revised MLM to evaluate the prediction quality of all masked words in these inputs. We take the average of all the quality scores as the quality of the segmentation result:

$$\begin{aligned} q(\mathbf{y},\mathbf{x}) &= \mathbb{E}_{\mathbf{x}_m|\mathbf{x}_o^{(s)},y;\gamma}\left[\Delta\left(\mathbf{x}_m,\mathbf{x}_m^{(s)}\right)\right] \\ &= \sum_{\mathbf{x}_o^{(s)}\in M(\mathbf{x},\mathbf{y})} P(\mathbf{x}_m|\mathbf{x}_o^{(s)};\gamma)\Delta\left(\mathbf{x}_m,\mathbf{x}_m^{(s)}\right), \end{aligned} \tag{1}$$

where $\mathbf{x}$ and $\mathbf{y}$ represent the input sequence and tag sequence, respectively, $M(\mathbf{x},\mathbf{y})$ denotes the set of all the legal maskings of $\mathbf{x}$ when the segmentation result is $\mathbf{y}$, and $\mathbf{x}_m$ denotes the results of the prediction from MLM. $\mathbf{x}_m^{(s)}$ and $\mathbf{x}_o^{(s)}$ respectively represent the ground-truth of the masked part and the observed part. γ indicates the parameter of the MLM. $\Delta\left(\mathbf{x}_m,\mathbf{x}_m^{(s)}\right) = 1 - sim\left(\mathbf{x}_m,\mathbf{x}_m^{(s)}\right)$ represents the difference between $\mathbf{x}_m$ and $\mathbf{x}_m^{(s)}$, where $sim\left(\mathbf{x}_m,\mathbf{x}_m^{(s)}\right)$ is the cosine similarity between $\mathbf{x}_m$ and $\mathbf{x}_m^{(s)}$, which can be obtained from BERT embeddings.

According to Equation (1), a larger value of $q(\mathbf{y},\mathbf{x})$ indicates a larger gap between the prediction result and ground-truth, i.e, a worse quality of prediction results.

3.3 Training Procedure with Improved MRT

After we train the segmentation model with NLL loss, we further train it using MRT (Shen et al., 2016). Specifically, on the training data $\mathbf{X}$, we optimize

$$\begin{aligned} J(\theta) &= \sum_{\mathbf{x}\in\mathbf{X}} \mathbb{E}_{\mathbf{y}|\mathbf{x};\theta}\left[q(\mathbf{y},\mathbf{x})\right] \\ &= \sum_{\mathbf{x}\in\mathbf{X}} \sum_{\mathbf{y}\in Y(\mathbf{x})} P(\mathbf{y}|\mathbf{x};\theta)q(\mathbf{y},\mathbf{x}), \end{aligned} \tag{2}$$

Cand.	$q(\cdot,\mathbf{x})$	Model			
		θ_1		θ_2	
		P	Q	P	Q
$\mathbf{y}_1$	-1.0	0.81	0.90	0.099	0.99
$\mathbf{y}_2$	0.0	0.09	0.10	0.001	0.01
$\mathbb{E}_{\mathbf{y}\vert\mathbf{x};\cdot,\alpha}\left[q(\mathbf{y},\mathbf{x})\right]$		-0.90		-0.99	

Table 2: Example of an abnormal phenomenon in MRT loss without regularization. "Cand.", P and Q denote "Candidate", $P(\cdot|\mathbf{x};\cdot,\alpha)$ and $Q(\cdot|\mathbf{x};\cdot)$, respectively.

where θ is the parameter of the segmentation model, and $Y(\mathbf{x})$ is the set of all possible word segmentation results of $\mathbf{x}$.

However, due to the large number of possible segmentation results, the computational cost of Equation (2) is unacceptably large. Therefore, we sample a subset of $S(\mathbf{x})$ from $Y(\mathbf{x})$, and define a new probability distribution Q on $S(\mathbf{x})$:

$$Q(\mathbf{y}|\mathbf{x};\theta,\alpha) = \frac{P(\mathbf{y}|\mathbf{x};\theta)^{\alpha}}{\sum_{\mathbf{y}'\in S(\mathbf{x})} P(\mathbf{y}'|\mathbf{x};\theta)^{\alpha}}, \tag{3}$$

where α is a parameter that controls the sharpness of Q. We calculate the approximation of Equation (2) on Q:

$$\begin{aligned} J(\theta) &\approx \sum_{\mathbf{x}\in\mathbf{X}} \mathbb{E}_{\mathbf{y}|\mathbf{x};\theta,\alpha}\left[q(\mathbf{y},\mathbf{x})\right] \\ &= \sum_{\mathbf{x}\in\mathbf{X}} \sum_{\mathbf{y}\in S(\mathbf{x})} Q(\mathbf{y}|\mathbf{x};\theta,\alpha)q(\mathbf{y},\mathbf{x}), \end{aligned} \tag{4}$$

Additionally, the loss defined in Equation (4) can only provide a weak supervision signal, because when the denominator of Equation (3) becomes smaller, the loss can be rather low even if the value of $P(\mathbf{y}|\mathbf{x};\theta)$ is very small (see Table 2). This may decrease the probability of some good segmentation results, thereby reducing the performance of the segmentation model. Therefore, we modify the loss defined in Equation (4) by adding a regularization term to mitigate the impact of getting the denominator of $Q(\mathbf{y}|\mathbf{x};\theta,\alpha)$ smaller:

$$\begin{aligned} J(\theta) = \sum_{\mathbf{x}\in\mathbf{X}} \Bigg(&\sum_{\mathbf{y}\in S(\mathbf{x})} Q(\mathbf{y}|\mathbf{x};\theta,\alpha)q(\mathbf{y},\mathbf{x}) \\ &- \lambda \sum_{\mathbf{y}'\in S(\mathbf{x})} P(\mathbf{y}'|\mathbf{x};\theta)^{\alpha}\Bigg), \end{aligned} \tag{5}$$

where the hyper-parameter λ is used to adjust the weight of the regularization term.

Corpora	Train	Dev.	Test	Word			Char		
				Type	Token.	Avglen.	Type	Token.	Avglen.
MSRA	84.80K	2.0K	4.0K	90.10K	2.50M	27.24	5.20K	4.01M	46.62
PKU	19.06K	2.0K	1.9K	58.20K	1.21M	57.82	4.70K	1.83M	95.85
AS	0.7M	2.0K	14.4K	0.14M	5.60M	7.7	6.11K	8.37M	11.80
CITYU	53.02K	2.0K	1.5K	70.76K	1.50M	27.45	4.92K	2.40M	45.33
CTB	24.42K	1.9K	2.0K	47.60K	0.80M	27.67	4.44K	1.30M	45.50
SXU	15.62K	1.5K	3.7K	35.92K	0.64M	30.90	4.28K	1.04M	50.50
CNC	0.21M	25.9K	25.9K	0.14M	7.30M	28.19	6.86K	10.08M	43.28
UDC	4.0K	0.5K	0.5K	20.13K	0.12M	24.67	3.60K	0.20M	39.14
ZX	2.37K	0.8K	1.4K	9.14K	0.12M	26.87	2.61K	0.17M	38.05

Table 3: Statistics of our corpora. "Dev." indicates the validation set, "Type" and "Token" denote non-repeated tokens and all tokens, respectively. "Avglen" represents the average length of sentences.

Parameter	BERT
Hidden Layer	768
Number of Layers	12
Number of Heads	12
Learning Rate	$2e-5$
Batch Size	64
Dropout	0.1
Epochs	10

Table 4: Hyper-parameter settings.

4 Experiments

4.1 Setup

Data Preparation

All the corpora used in our experiment are from SIGHAN05 (Emerson, 2005), SIGHAN08 (Jin and Chen, 2008), SIGHAN10 (mei Zhao and Liu, 2010) and some OTHER open datasets (Zhang et al., 2014) respectively. The statistics of our corpora are shown in Table 3 and some hyper-parameters also given in Table 4. The datasets MSRA, PKU, AS and CITYU are from the corpora SIGHAN05[2], while the datasets CTB and SXU are from SIGHAN08 and CNC, UDC and ZX are from OTHER open datasets. Both the corpora SIGHAN08 and OTHER datasets are also openly available[3]. SIGHAN10 contains data in different domains, and we choose "Finance", "Literature" and "Medicine" for our cross-domain experiment. Besides, we take CTB6 as CTB dataset in our whole experiment. We use the original format of AS and CITYU instead of using their corresponding simplified versions. Furthermore, we use the same data pre-processing as used in Huang et al. (2020) for whole experiment.

Both in the single criterion and multiple criteria experiments, the majority of results are originated from their corresponding papers. For the multiple criteria experiment (Chen et al., 2017), we follow He et al. (2018) and prepare the training data by combining all the datasets. For the noisy-labeled experiment, we convert the input sequence into character and randomly generate four tags (e.g. B, M, E, and S) for each position of the characters among the input sequence. We use the identical pre-processed data for all architectures and we only build 10% noisy-labeled data of each corpus and use 90% real data. For the revised masking strategy, we explore the best accuracy of the predictor by training and testing the MLM on SIGHAN05.

Baselines

We compare our method with the following strong baselines in the field of CWS:

1. LSTM+Beam: Cai et al. (2017) present a greedy neural word segmenter. The model is based on LSTM and modifies beam search for decoding.

2. LSTM+CRF: Ma et al. (2018) find that a bidirectional LSTM, when tuning the parameters carefully, can achieve better accuracy on many of the benchmark datasets.

[2] http://sighan.cs.uchicago.edu/bakeoff2005/

[3] https://github.com/hankcs/multi-criteria-cws/tree/master/data/other

Methods	SIGHAN05				SIGHAN08		OTHER		
	MSRA	PKU	AS	CITYU	CTB	SXU	CNC	UDC	ZX
Chen et al. (2017)	95.84	93.30	94.20	94.07	95.30	95.17	–	–	–
Zhou et al. (2017)	97.80	96.00	–	–	96.20	–	–	–	–
Yang et al. (2017)	97.50	96.30	95.70	96.90	96.20	–	–	–	–
He et al. (2018)	97.29	95.22	94.90	94.51	95.21	95.78	97.11	93.98	95.57
Gong et al. (2019)	96.46	95.74	94.51	93.71	97.09	95.57	–	–	–
LSTM+BEAM	97.10	95.80	95.30	95.60	96.10	95.95	96.10	96.20	96.30
LSTM+CRF	98.10	96.10	96.00	96.80	96.30	96.55	96.61	96.00	96.40
BERT	96.91	95.34	96.47	97.10	97.27	96.40	96.66	97.23	96.49
SELFATT+SOFT	97.60	95.50	95.70	96.40	97.28	96.60	96.88	97.12	96.50
BERT+LTL	97.53	96.23	97.03	97.63	97.34	96.65	96.89	97.51	96.72
Ours	**98.12**	**96.24**	**97.30**	**97.83**	**97.45**	**96.97**	**97.25**	**97.74**	**96.82**

Table 5: Comparison among the SOTA performance (F1-score, %) on the test datasets of 9 standard CWS datasets using **single criterion learning**. "BERT" denotes we take BERT as our PTM in the training. Any underlined result represents that re-implemented scores.

Methods	SIGHAN05				SIGHAN08		OTHER		
	MSRA	PKU	AS	CITYU	CTB	SXU	CNC	UDC	ZX
Chen et al. (2017)	96.04	94.32	94.64	95.55	96.18	96.04	–	–	–
He et al. (2018)	97.35	95.78	95.47	95.60	95.84	96.49	97.00	94.44	95.72
Gong et al. (2019)	97.78	96.15	95.22	96.22	97.26	97.25	–	–	–
BERT	97.22	96.06	97.07	97.39	97.36	96.81	96.71	97.48	96.60
BERT+LTL	96.67	96.30	97.16	97.72	97.38	96.90	97.10	97.61	96.81
Ours	**98.19**	**96.32**	**97.43**	**97.80**	**97.66**	**97.03**	**97.34**	**98.25**	**97.08**

Table 6: Comparison among the SOTA performance (F1-score, %) on the test datasets of 9 standard CWS datasets using **multiple criteria learning**. "BERT" represents that we regard BERT as our PTM in the training. The underlined results represent that we re-implement the existing methods for a fair comparison.

3. BERT: Devlin et al. (2019) present an effective pre-trained language model based on Transformer. It can achieve good performance via fine-tuning.

4. SELFATT+SOFT: Duan and Zhao (2020) modify a Gaussian-masked Directional Transformer without bi-gram features, and a biaffine attention scorer.

5. BERT+LTL: Huang et al. (2020) present a linear transfer layer to incorporate multiple-criteria segmentation data into one model.

4.2 Main Experiments

Results of Single Criterion Learning

As shown in Table 5, our proposed method obtains better results on different standard datasets with single criterion learning. Different segmentation criteria are used in the popular datasets. Especially, the segmentation rules of PKU, MSRA and ZX are different from each other (Huang et al., 2020). Therefore, to investigate the quality of our segmentation model, we compare our approach with the previous SOTA methods on the 9 benchmark datasets of CWS. We refer to the reported results in their corresponding papers, except the baselines BERT and BERT+LTL on SIGHAN05 corpora. However, for the other two corpora (i.e., SIGHAN08 and OTHER) we almost re-run the not reported results in their papers. Due to the low GPU memory, we re-implement the BERT version of BERT+LTL rather than using ROBERTA. We report all the results with single criterion learning.

Results of Multiple Criteria Learning

As given in Table 6, to further validate the quality of our method, we also conduct the multiple cri-

Methods	SIGHAN05				SIGHAN08		OTHER		
	MSRA	PKU	AS	CITYU	CTB	SXU	CNC	UDC	ZX
LSTM+BEAM	96.86	95.70	95.17	95.35	95.89	95.83	95.89	96.07	96.18
LSTM+CRF	97.89	95.89	95.88	96.67	96.19	96.47	96.49	95.85	96.25
BERT	96.78	95.20	96.28	97.01	97.14	96.24	96.51	97.11	96.30
SELFATT+SOFT	97.47	95.40	95.57	96.29	97.16	96.49	96.61	97.08	96.33
BERT+LTL	97.42	96.15	96.76	97.52	97.27	96.55	96.69	97.40	96.53
Ours	**97.93**	**96.18**	**97.12**	**97.68**	**97.32**	**96.83**	**97.12**	**97.63**	**96.67**

Table 7: Comparison among the strong baselines (F1-score, %) with noisy-labeled training on 9 CWS datasets using **single criterion learning**. "BERT" denotes that we take BERT as our PTM in the training.

Methods	Fin.	Lit.	Med.
Chen et al. (2015b)	95.20	92.89	92.16
Cai et al. (2017)	95.38	92.90	92.10
Huang et al. (2017)	95.81	94.33	92.26
Zhao et al. (2018)	95.84	93.23	93.73
Zhang et al. (2018)	96.06	94.76	94.18
BERT	95.87	95.57	94.66
BERT+LTL	95.96	95.88	94.87
Ours	95.93	**95.96**	**95.08**

Table 8: Comparison among the SOTA performance (F1-score, %) with supervised training on different domains. "Fin.", "Lit." and "Med." represent different domains (i.e., "Finance", "Literature" and "Medicine"). The underlined results represent that we re-implement the existing methods for a fair comparison.

teria experiment which is proposed by Chen et al. (2017) and we compare the performance of our model with other methods on the same corpora as single criterion training. Our proposed approach consistently outperforms previous SOTA methods. Although we remarkably outperform all baselines on the majority of datasets, we find that some results in multiple criteria learning are highly close to, and sometimes lower than the results of single criterion training. We also directly refer to the results of their papers except BERT and BERT+LTL. We only compare with a few baselines which also explore multiple criteria learning. The effectiveness of multiple criteria learning does not improve the performance of our model on CITYU corpus. However, on the other datasets, we obtain higher results than single criterion learning.

Comparison on Low-quality Datasets

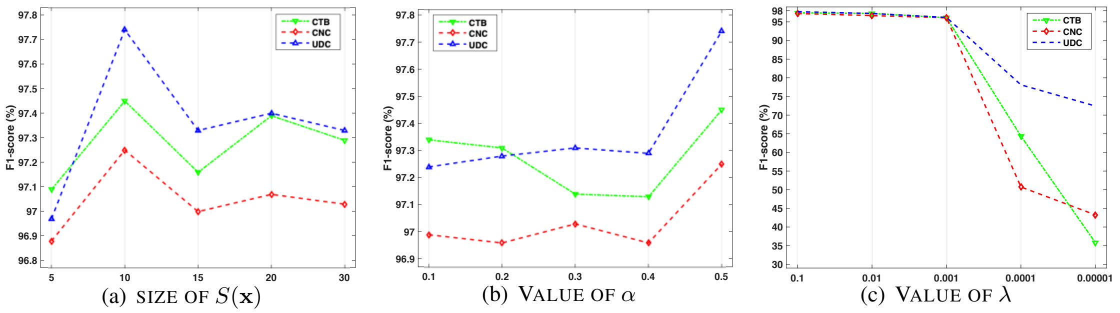
As Table 7 shows, to analyze the robustness of our proposed method with respect to the revised MLM, we prepare noisy-labeled datasets which contain 90% real data and 10% randomly shuffled data (see Section 4.1). In this experiment, we exploit the single criterion training on the noisy-labeled data rather than using multiple criteria training. We run all the models on the same noisy-labeled datasets with their corresponding architectures. Obviously, all the results are almost lower than the results from single criterion training. However, our proposed method still gains better results than SOTA baselines with noisy-labeled datasets rather than the standard labeled data. Not only with the single criterion training and multiple criteria training but also with the noisy-labeled data training, we constantly obtain improvements over highly similar previous work.

Comparison on Different Domains

In Table 8, to further validate the effectiveness of our model, we choose some datasets in different domains from SIGHAN10 corpora and compare the segmentation quality of highly similar previous works which also used cross-domain datasets. In this experiment, we also refer to the reported results from their papers, except the baseline systems BERT and BERT+LTL. We use the model trained on PKU corpus and test the different domain test datasets. The presented approach also gains better performance on the "Literature" and "Medicine" domain compared to other approaches but obtains worse results than BERT+LTL on the domain of "Finance".

4.3 Effect of Masked Count in MLM

As shown in Table 9, to explore the influences of the value of $mask\_count$ for the quality of MLM, we train MLM with different values of $mask\_count$. We find that the accuracy of the predictor achieves the highest score when



(a) SIZE OF $S(\mathbf{x})$ (b) VALUE OF α (c) VALUE OF λ

Figure 3: The effect of different values of the hyper-parameters $S(\mathbf{x})$, α and λ in our model from single criterion learning. (a) denotes the F1-score (%) with different size of $S(\mathbf{x})$ (default value is 10); (b) and (c) also represent the F1-score (%) with different values of α (default value is 0.5) and λ (default value is 0.1), respectively.

$mask\_count$	Accuracy (%)
2	33.60
3	21.93
4	22.12
5	22.19

Table 9: Comparison of the accuracy of predictor between different $mask\_count$.

Corpora	PTM	P.	R.	F.
MSRA	×	97.06	97.61	97.34
	√	98.18	98.06	**98.12**
AS	×	96.05	96.78	96.41
	√	96.30	98.33	**97.30**
CTB	×	95.97	96.23	96.10
	√	97.49	97.41	**97.45**
CNC	×	96.08	95.42	95.75
	√	97.41	97.08	**97.25**

Table 10: The effect of the PTM on our model with single criterion learning. “P., R. and F." denote the evaluation methods of precision, recall and F1-score (%). “√" and “×" represent with or without PTM, respectively.

$mask\_count = 2$. Note that if $mask\_count = 1$, only one character can be masked. In this case, masking any character is legal, regardless of the segmentation result. Therefore, we analyze the case where the $mask\_count$ is greater than or equal to 2 and choose the $mask\_count$ number that makes the accuracy of the MLM highest.

4.4 Ablation Study

Effect of Pre-Trained Model

As shown in Table 10, we explore the influences of the PTM on the segmentation model with single criterion training. We take BERT as PTM and explore the effect of PTM to the quality of our word segmentation model on different datasets (i.e, MSRA, AS, CTB and CNC) with different segmentation criteria. Intuitively, the performance of our segmentation approach with PTM obtains remarkably better results than without using PTM.

Effect of Hyper-Parameters

We regard improved MRT as a crucial part of our self-supervised word segmentation architecture. To choose the best values of the hyper-parameters, we explore the different values of α, λ and the size of $S(\mathbf{x})$ on the datasets CTB, CNC and UDC with single criterion training.

Effect of size of $S(\mathbf{x})$ $S(\mathbf{x})$ is a subset of all word segmentation results $Y(\mathbf{x})$ corresponding to the sentence $\mathbf{x}$, which is used to generate the distribution Q defined in Equation (3). As shown in Figure 3(a), when the size of $S(\mathbf{x}) = 10$, improved MRT enhances the quality of segmentation model remarkably better than other values on the different corpora.

Effect of α α is used to control the sharpness of the distribution Q defined in Equation (3). As depicted in Figure 3(b), when $\alpha = 0.5$ improved MRT increases the quality of our segmentation model outstandingly on the different corpora.

Effect of λ λ is the regularization term for improved MRT, which appears in Equation (5). As illustrated in Figure 3(c), when $\lambda = 0.1$ our model achieves the best segmentation performance compared to other values.

5 Conclusion and Future Work

In this work, we propose a self-supervised method for CWS. We first generate masked sequences based on the segmentation results and then use revised MLM to evaluate the quality of segmentation and enhance the segmentation by improved MRT. Experimental results show that our approach outperforms previous methods on both popular and cross-domain CWS datasets, and has better robustness on noised-labeled data. In the future, we can also extend our work to tasks of morphological word segmentation (e.g., morphological analysis).

Acknowledgments

This work was supported by the National Key R&D Program of China (No. 2018YFB1005103), National Natural Science Foundation of China (No. 62006138, No. 61925601, No. 61772302) and Huawei Noah's Ark Lab. We thank all anonymous reviewers for their valuable comments and suggestions on this work.

References

Deng Cai and Hai Zhao. 2016. Neural word segmentation learning for chinese. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 409–420, Berlin, Germany. Association for Computational Linguistics.

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 the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 608–615, Vancouver, Canada. Association for Computational Linguistics.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, and Xuanjing Huang. 2015a. Gated recursive neural network for chinese word segmentation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1744–1753, Beijing, China. Association for Computational Linguistics.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and X. Huang. 2015b. Long short-term memory neural networks for chinese word segmentation. In *EMNLP*.

Xinchi Chen, Zhan Shi, Xipeng Qiu, and X. Huang. 2017. Adversarial multi-criteria learning for chinese word segmentation. In *ACL*.

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, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Sufeng Duan and Hai Zhao. 2020. Attention is all you need for chinese word segmentation. In *EMNLP*.

T. Emerson. 2005. The second international chinese word segmentation bakeoff. In *SIGHAN@IJCNLP 2005*.

Jingjing Gong, Xinchi Chen, T. Gui, and Xipeng Qiu. 2019. Switch-lstms for multi-criteria chinese word segmentation. In *AAAI*.

Han He, Lei Wu, Hua Yan, Zhimin Gao, Yi Feng, and George Townsend. 2018. Effective neural solution for multi-criteria word segmentation. *Smart Intelligent Computing and Applications*, 2:133.

Chang-Ning Huang and Hai Zhao. 2007. Chinese word segmentation: a decade review. *Journal of Chinese Information Processing*, 21(3):8–19.

Kaiyu Huang, Degen Huang, Zhuang Liu, and Fengran Mo. 2020. A joint multiple criteria model in transfer learning for cross-domain chinese word segmentation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pages 3873–3882. Association for Computational Linguistics.

S. Huang, X. Sun, and Houfeng Wang. 2017. Addressing domain adaptation for chinese word segmentation with global recurrent structure. In *IJCNLP*.

G. Jin and Xiao Chen. 2008. The fourth international chinese language processing bakeoff: Chinese word segmentation, named entity recognition and chinese pos tagging. In *IJCNLP*.

Ji Ma, Kuzman Ganchev, and David Weiss. 2018. State-of-the-art chinese word segmentation with bilstms. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4902–4908.

Hong mei Zhao and Qun Liu. 2010. The cips-sighan clp 2010 chinese word segmentation bakeoff.

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*, pages 2742–2753.

Wenzhe Pei, Tao Ge, and Baobao Chang. 2014. Max-margin tensor neural network for chinese word segmentation. In *ACL*.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING '04)*, pages 562–568, Geneva, Switzerland. Association for Computational Linguistics, Association for Computational Linguistics.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

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, ACL 2020, Online, July 5-10, 2020*, pages 8274–8285. Association for Computational Linguistics.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter for sighan bakeoff2005. In *Proceedings of the Fourth SIGHAN workshop on Chinese Language Processing*, pages 168–171, Jeju Island, Korea.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, 8(1):29–48.

Jie Yang, Yue Zhang, and F. Dong. 2017. Neural word segmentation with rich pretraining. In *ACL*.

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.

Meishan Zhang, Yue Zhang, W. Che, and T. Liu. 2014. Type-supervised domain adaptation for joint segmentation and pos-tagging. In *EACL*.

Qi Zhang, X. Liu, and Jinlan Fu. 2018. Neural networks incorporating dictionaries for chinese word segmentation. In *AAAI*.

Hai Zhao, Deng Cai, Changning Huang, and Chunyu Kit. 2019. Chinese word segmentation: Another decade review (2007-2017). *arXiv preprint arXiv:1901.06079*.

Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2010. A unified character-based tagging framework for chinese word segmentation. *ACM Transactions on Asian Language Information Processing*, 9(2):1–32.

Hai Zhao and Chunyu Kit. 2008. Unsupervised segmentation helps supervised learning of character tagging for word segmentation and named entity recognition. In *The Sixth SIGHAN Workshop on Chinese Language Processing*, pages 106–111, Hyderabad, India.

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 *IJCAI*, pages 4602–4608.

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 (EMNLP)*, pages 647–657, Seattle, Washington, USA. Association for Computational Linguistics.

Hao Zhou, Zhenting Yu, Yue Zhang, Shujian Huang, Xin-Yu Dai, and Jiajun Chen. 2017. Word-context character embeddings for chinese word segmentation. In *EMNLP*.