CWSeg: An Efficient and General Approach to Chinese Word Segmentation

Dedong Li^{+} [1], **Rui Zhao** [1], **Fei Tan*** [1]
[1] SenseTime Research
ddlecnu@gmail.com
{zhaorui, tanfei}@sensetime.com

Abstract

In this work, we report our efforts in advancing Chinese Word Segmentation for the purpose of rapid deployment in different applications. The pre-trained language model (PLM) based segmentation methods have achieved state-of-the-art (SOTA) performance, whereas this paradigm also poses challenges in the deployment. It includes the balance between performance and cost, segmentation ambiguity due to domain diversity and vague words boundary, and multi-grained segmentation. In this context, we propose a simple yet effective approach, namely CWSeg, to augment PLM-based schemes by developing cohort training and versatile decoding strategies. Extensive experiments on benchmark datasets demonstrate the efficiency and generalization of our approach. The corresponding segmentation system is also implemented for practical usage and the demo is recorded.

1 Introduction

Chinese word segmentation (CWS) is a preliminary but essential procedure for Chinese language processing tasks, and has been applied in various scenarios (Yang et al., 2018; Zhang et al., 2019; Cui et al., 2020; Han et al., 2020; Zhang et al., 2020; Tan et al., 2020; Lu et al., 2023). Especially for fast complete recall and accurate semantic understanding in search and recommendation scenarios (Bao et al., 2022), CWS is still indispensable. In addition, experiments on Chinese LLaMA and Alpaca show that the token throughput of the model that expands the vocabulary through word segmentation has greatly improved the processing of Chinese text compared with the original model (Cui et al., 2023). Recent deep learning methods have achieved remarkable results on publicly available datasets in this regard (Qiu et al., 2019). Also, the pre-trained language model (PLM) (Liu et al., 2019) further emerges as the paramount foundation of text representation for CWS as seen in other tasks (Tian et al., 2020b; Huang et al., 2020a; Maimaiti et al., 2021).

Current PLM-based approaches, however, pose three hurdles to the production deployment we need to cross: (1) One dilemma is the trade-off between the model performance and inference speed. (2) The lexical diversity and domain gap also jeopardize the fast deployment of a generic model to customized scenarios. (Maimaiti et al., 2021). (3) PLM-based schemes with single granularity are less likely to meet multi-granularity demands of practical relevance.

To tackle these issues, we propose an **e**fficient and **g**eneral approach to augmenting PLM-based **C**hinese **W**ord **S**egmentation methods, namely **CWSeg**. It can extrapolate to different sequence labeling scenarios. Recent studies showed that small models also have the potential to be comparable to large models (Ba and Caruana, 2014; Zhang et al., 2018). We thus introduce a new cohort training strategy to co-train a cohort of multi-scale model artifacts to meet the performance and real-time demands. Specifically, we employ Wasserstein distance (WD) (Rüschendorf, 1985) to orchestrate distributions of model cohorts to enable more robust learning. In addition, we propose to construct the tailored domain-specific lexicon Trie (Liu et al., 2002) and build up a versatile decoding scheme to augment the optimal segmentation path searching on the fly for diverse practical scenarios. It can flexibly adjust the segmentation granularity and benefit customized domains.

In summary, our primary goal is to build a versatile framework for strengthening different models simultaneously and then rapidly deploying them into multiple practical scenarios of CWS, which is fundamentally different from existing research works. Essentially, the output models of this framework can be regarded as complements to, not re-

[+] Work was done at SenseTime Research

[*] Corresponding author

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 5: Industry Track, pages 1–10
July 10-12, 2023 ©2023 Association for Computational Linguistics

placements for, existing SOTA methods.

Experimental results on multiple benchmark datasets demonstrate the effectiveness of our approach. Ablation studies confirm the necessity of cohort training strategy and lexicon Trie aided versatile decoding solution. The cross-domain application experiments demonstrate the generalization capacity of our holistic approach.

2 Related Work

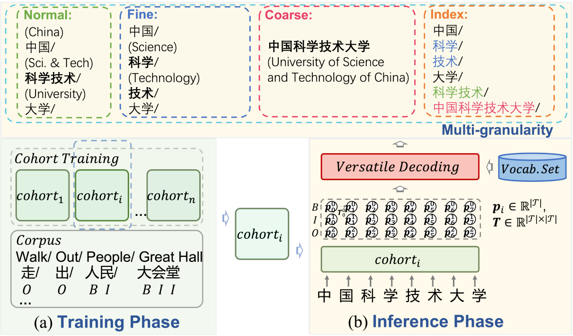
Early work in Chinese word segmentation builds upon the statistical assumption (Li and Sun, 2009; Sun et al., 2012a) by modeling rules into the learning process. Recently, PLMs have been introduced (Tian et al., 2020b,a; Maimaiti et al., 2021) and made significant advances in this regard. Our work, however, aims to alleviate their potential challenges involved in the industrial applications as mentioned in Section 1.

Recent works (Huang et al., 2020b, 2021) distill knowledge from the well-trained teacher model into a student model to balance the model scale and performance. However, it requires multiple fine-tuning rounds and models can't learn from each other collaboratively. In this work, we introduce a cohort training based learning strategy to address these two problems for CWS. Different from the pioneering mutual learning (Zhang et al., 2018) in computer vision, we propose Wasserstein distance to better enable the learning as studied in Sec. 4.3. It's a more carbon-footprint-friendly solution as compared to recent research threads.

To mitigate the effects of Chinese lexical diversity, Qiu *et al.* (Qiu et al., 2019) proposed a concise unified model to extract the criterion-aware representation for multi-criteria corpus, which requires training from scratch on the entire corpus for new criteria or domains. Gong *et al.* (Gong et al., 2017, 2020) proposed a multi-grained word segmentation by training with large-scale pseudo labels, which is relatively lagging for rapid deployment to new domains. Our work approaches this issue by a lightweight versatile decoding scheme to sidestep heavy training loads.

3 Methodology

As shown in Fig. 1 (a), we formulate CWS as a classical sequence labeling problem as with existing compelling schemes. Concretely, given a text sequence of n characters $\mathcal{X} = \{x_1, \ldots, x_n\}$, CWS is to tag involved characters sequentially with the BIO encoding by maximizing their joint probability $\mathrm{p}(y_1, \ldots, y_n|\mathcal{X})$ where $y_i \in \mathcal{T} = \{\mathrm{B}, \mathrm{I}, \mathrm{O}\}$, short for beginning, inside and outside respectively.



(a) Training Phase (b) Inference Phase

Figure 1: Overview of the CWSeg framework. (a) In the training phase, we set several SOTA models as training cohorts and initial weights from PLM. (b) In the inference stage, we select the most suitable artifacts from the cohort for the actual scenario and apply the versatile decoding strategy for the multi-granularity demands.

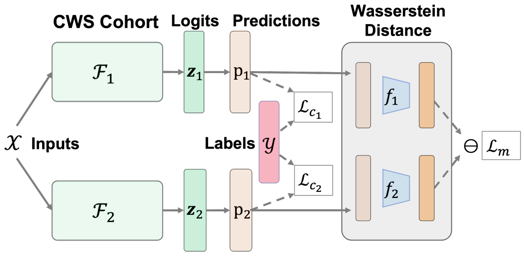


Figure 2: The cohort training strategy.

3.1 Cohort Training

The cohort training strategy enables multiple student models to teach and learn from each other. The objective function contains supervised loss $\mathcal{L}_c$ and mimicry loss $\mathcal{L}_m$. As exemplified by two models in Fig. 2, the overall loss function is:

$$\mathcal{L} = \mathcal{L}_{c_1} + \mathcal{L}_{c_2} + \lambda \cdot \mathcal{L}_m \tag{1}$$

where $\lambda \in [0, 1]$ is a hyper-parameter. $\mathcal{L}_{c_1}$ and $\mathcal{L}_{c_2}$ guide the model learning under the supervision of real segmentation tags while $\mathcal{L}_m$ can encourage different models to learn from each other collaboratively.

Specifically, $\mathcal{L}_{c1}$ and $\mathcal{L}_{c2}$ refer to the cross entropy (CE) loss. Without loss of generality, $\mathcal{L}_{c_1} = -\sum_{i=1}^{N}\sum_{t=1}^{|\mathcal{T}|} I(y_i, t) log(p_1^t(\mathbf{x}_i))$ and $p_1^t(\mathbf{x}_i) = \frac{exp(z_1^t)}{\sum_{t=1}^{|\mathcal{T}|} exp(z_1^t)}$ where $I(\cdot)$ is an indicator function, $p_1^t(x_i)$ is the prediction probability, z_1^t is the output logit of the model $\mathcal{F}_1$. For $\mathcal{L}_m$, Kullback-Leibler (KL) divergence is a naive metric to quantify the distance between two distributions $KL(\mathbf{p}_2||\mathbf{p}_1) = \sum_{i=1}^{N}\sum_{t=1}^{|\mathcal{T}|} p_2^t(\mathbf{x}_i)\frac{p_2^t(\mathbf{x}_i)}{p_1^t(\mathbf{x}_i)}$. However, KL divergence is asymmetric and possibly infinite when two distributions are disjoint or there are points such that $p_1(\mathbf{x}_i) = 0$ and $p_2(\mathbf{x}_i) > 0$, which is fragile in training (Arjovsky et al., 2017). The symmetric Jensen-Shannon (JS) divergence, suffers from the same problem (See A.1 for more details). Given the above concerns, we introduce the Wasserstein-1 distance (a.k.a. earth mover's distance):

$$W(\mathbf{p}_2, \mathbf{p}_1) = \inf_{\gamma \in \prod(\mathbf{p}_2, \mathbf{p}_1)} \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\gamma}[\|\mathbf{x} - \mathbf{y}\|] \quad (2)$$

where $\prod(\mathbf{p}_2, \mathbf{p}_1)$ is the set of all joint distributions $\gamma(\mathbf{x}, \mathbf{y})$ whose marginals are $\mathbf{p}_2$ and $\mathbf{p}_1$, respectively. As shown in Appendix A.1, Wasserstein distance can provide a meaningful and smooth representation of the in-between distance for two distributions in lower dimensional manifolds without overlaps. Eq. (2), however, is highly intractable. We thus resort to Kantorovich-Rubinstein duality:

$$W(\mathbf{p}_2, \mathbf{p}_1) = \sup_{\|f\|\leq 1} \mathbb{E}_{\mathbf{x}\sim\mathbf{p}_2}[f(\mathbf{x})] - \mathbb{E}_{\mathbf{y}\sim\mathbf{p}_1}[f(\mathbf{y})] \quad (3)$$

where the supremum is over all the 1-Lipschitz [*] function $f : \mathbb{R}^K \rightarrow \mathbb{R}$, which maps each K-dimensional feature vector in the semantic space to a real number. In practice, f is implemented as a two-layer feed-forward neural network with parameters Θ_f clipped to $[-c, c]$, where $c > 0$. Therefore, the mimicry loss $\mathcal{L}_m$ can be derived as the dual form of Wasserstein distance:

$$\mathcal{L}_m = \max_{\Theta_f} \sum_{(\mathbf{x},\mathbf{y})} [f(\mathbf{x}) - f(\mathbf{y})] \quad (4)$$

Extension to Larger Cohort The cohort training strategy can be easily extended to larger cohorts. For example, given K models ($K \geq 2$), the overall loss function $\mathcal{L}$ can be formulated as:

$$\mathcal{L} = \sum_{i=1}^{K}\mathcal{L}_{c_i} + \frac{2\cdot\lambda}{K(K-1)}\sum_{i=1}^{K}\sum_{j=i+1}^{K} W(\mathbf{p}_j, \mathbf{p}_i) \quad (5)$$

Obviously, Eq. (1) is a special case of Eq. (5) when $K = 2$.

3.2 Versatile Decoding

However, the PLM-based segmentation capacity of single-granularity barely meets diverse real-world applications. As illustrated in Fig. 3 (a), the model tends to decode the input text as "中国 (China) / 科学技术 (Science and Technology) / 大学 (University)", whereas only the input as a whole "中国科学技术大学 (University of Science and Technology of China, USTC)" refers to a meaningful entity. Additionally, for large-scale content recommendations, rapidly acquiring as much relevant content as possible is an essential step towards quality candidates on which more sophisticated methods can function. Thus, reasonably splitting the whole entity of "中国科学技术大学 (USTC)" into smaller relevant semantic units "中国 (China) / 科学 (Science) / 技术 (Technology) / 大学 (University)" is crucial in this regard.

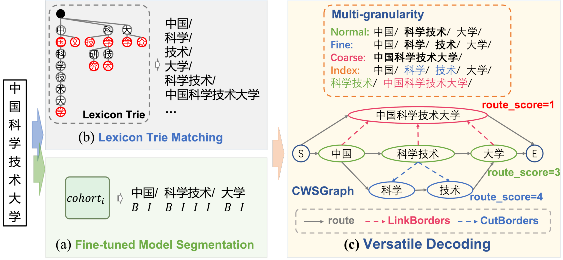


Figure 3: The versatile decoding strategy. (a) The bottom left shows the fine-tuned model prediction, which is largely affected by the training corpus. (b) The top left shows phrases matched by the lexicon Trie built from a user-defined vocabulary set. (c) The right part integrates matching results by constructing CWSGraph and uses the Viterbi algorithm for dynamic decoding according to granularity requirements.

In this context, we focus on adapting generic models trained on annotated corpora to specific domains and supporting diverse granularity. It includes the construction of lexicon Trie (Liu et al., 2002) and versatile decoding.

Lexicon Trie: The lexicon Trie is designed to store vocabulary in a compressed Trie structure and search for each word efficiently. As illustrated in Fig. 3 (b), the solid node denotes the root node, and each circle denotes a Trie node, which contains a value containing a Chinese token and a label representing whether it is a complete word from the root node so far. Here the red circle indicates that the label is equal to True. Thus, given a collected vocabulary set, we can initialize a lexicon Trie.

In the matching stage, given an input text such as "中国科学技术大学", we apply the matching algorithm to search for all complete words in the input text that can be matched on the lexicon Trie.

[*] f is 1-Lipschitz $\Leftrightarrow |f(\mathbf{x}) - f(\mathbf{x}')| \leq |\mathbf{x} - \mathbf{x}'|$ for all $\mathbf{x}$ and $\mathbf{x}'$

The matched word list is shown in Fig. 3 (b).

Diverse Modes: The granularity criterion $criteria$ is roughly determined by the $RouteScore$, which is the number of chunks in the segmented path regularized by semantic completeness. In total, we have the following four modes:

Normal Mode: High-probability segmentation that conforms to the statistics of the data.

Fine Mode: $RouteScore$ larger than normal, collecting more semantic units.

Coarse Mode: $RouteScore$ smaller than normal, perceiving more complete semantics.

Index Mode: A segmentation result that combines the above three modes.

The whole process can be formulated as Fig. 3 and Algorithm 1 (refer to A.1 for more function details). In addition to the prediction from the fine-tuned model $\mathcal{F}$, we create a lexicon Trie $\mathcal{D}$ from the pre-processed vocabulary set $\mathcal{V}$ to capture all candidate phrases $\mathcal{C}$ without training. We merge predictions $\mathcal{P}$ into candidates set $\mathcal{C}$ to construct CWSGraph $\mathcal{G}$, where each node represents a token. Viterbi algorithm is adopted for decoding according to the granularity criteria. In this way, we can flexibly tailor model-based segmentation results to multiple domain-specific scenarios while meeting the multi-granularity requirements.

Algorithm 1 Versatile Decoding

```
Input: Text sequence $\mathcal{X}$, fine-tuned model $\mathcal{F}$, lexicon Trie $\mathcal{D}$, granularity mode $m$.
Output: Text sequence label: $\mathcal{Y}$.
$\mathcal{P} = \mathcal{F}(\mathcal{X})$; $\mathcal{C} = \text{Matching}(\mathcal{X}, \mathcal{D})|\mathcal{P}$;
$\mathcal{G} = \text{CWSGraph}(\mathcal{C})$;
$borders = \text{ExtractBorders}(\mathcal{P})$;
if $m$ = "normal" then
    $\mathcal{Y} = \mathcal{P}$
else if $m$ = "fine" then
    $cands = \text{CutBorders}(\mathcal{G}, borders)$;
    $\mathcal{Y} = \text{Viterbi}(\mathcal{G}, cands, criteria_m)$;
else if $m$ = "coarse" then
    $cands = \text{LinkBorders}(\mathcal{G}, borders)$;
    $\mathcal{Y} = \text{Viterbi}(\mathcal{G}, cands, criteria_m)$;
else if $m$ = "index" then
    for $m$:["normal", "fine", "coarse"] do
        $\mathcal{Y}$ |= VersatileDecoding($\mathcal{X}, \mathcal{F}, \mathcal{D}, m$);
    end for
end if
return $\mathcal{Y}$
```

4 Experiments

4.1 Setup

Dataset We experiment with six widely-used datasets AS, CityU, CTB6, MSR, PKU, Weibo, from SIGHAN 2005 Bakeoff, Chinese Treebank and NLPCC2016 (SIGHAN2005Bakeoff; Emerson, 2005; Xue et al., 2005; Qiu et al., 2016). The basic statistics and train/dev/test settings are detailed in Table 1.

Corpus	Vocab. Size	Word Len.		Dataset Size		
		50%	75%	Train	Dev.	Test
AS	144.5k	3	3	698.9k	10.0k	14.4k
CityU	70.7k	2	3	47.7k	5.3k	1.5k
CTB6	47.5k	2	3	23.4k	2.0k	2.7k
MSR	90.1k	3	5	78.2k	8.7k	3.9k
PKU	58.1k	2	3	17.1k	1.9k	1.9k
Weibo	56.1k	2	3	20.1k	2.0k	8.5k

Table 1: The statistics of the datasets.

Baselines We select baselines both from traditional methods and the well-executed or SOTA methods, such as Jieba (jieba) (Fast CWS tool based on HMM), HanLP (pyhanlp) (CRF-based method), THU (THULAC) (Perceptron-based method), PKU (PKUSeg) (CRF-based CWS tool uses a new training method, namely, the adaptive online gradient descent method based on feature frequency (Sun et al., 2012b)). Since the major architecture of recent competing methods is CRF on top of Transformers (e.g., BERT and its variants), and as mentioned earlier, our flexible framework CWSeg is a complement to, not a replacement for, existing compelling methods, we experiment with our method on BERT-CRF (refer to A.1 for more details), which can be easily applied to other variants. WMSeg (Tian et al., 2020b), another most recent SOTA method based on this architecture utilizing memory networks to incorporate wordhood information, is also used for comparison. To be noted here, the PLMs implemented in BERT-CRF and WMSeg are the BERT base model. Since CWSeg adopts the cohort training strategy, we set base versions of BERT and NEZHA as cohorts.

Experiment Settings The PLMs used in this work are readily available, and are the widely recognized SOTA backbones in the Chinese community. Such as 'BERT' for bert-base-chinese (Devlin et al., 2019; bert-base chinese), 'RoBERTa' for chinese_roberta_wwm (Liu et al., 2019; chinese-roberta wwm), 'NEZHA' for NEZHA-Base-WWM

(Junqiu Wei, 2019; NEZHA-Base-WWM). They are based on Chinese characters (similar to subwords in English). We choose Adam optimizer (Kingma and Ba, 2014) with an initial learning rate as 2e-5 and tuned amongst {1e-4, 5e-5, 2e-5, 1e-5}. We use the early stopping mechanism (Yao et al., 2007) in the model training. The batch size was tuned amongst {32, 64, 128}. The hyper-parameter λ was set as 0.5 and tuned from [0.01, 1], and the clipping threshold c was set as 0.5 and tuned from [0.1, 0.5]. All experiments were run on Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz and NVIDIA V100-32g GPUs. Note here that all these time-cost comparison experiments are tested on the same CPU device, while deep methods run faster on CUDA devices.

	Adapt w/o Retraining	Multi-granularity	F1	Latency (s/k)
Jieba	✓	✓	80.67	0.17
HanLP	✓	✓	82.34	0.33
THU	✓	-	88.09	0.57
PKU	-	-	91.29	0.63
BERT-CRF	-	-	96.59	12.7
WMSeg	-	-	97.06	14.5
CWSeg	✓	✓	**97.65**	12.9

Table 2: Overall model comparison. 's/k' refers to seconds spent per thousand requests on the same CPU device.

Examples	Modes	Outputs
新型冠状病毒 COVID-19	Normal	新型/ 冠状/ 病毒 New Type/ Crown/ Virus
	Fine	新型/ 冠状/ 病毒
	Coarse	新型冠状病毒 COVID-19
	Index	新型/ 冠状/ 病毒/ 新型冠状病毒/ 冠状病毒 Coronavirus
上海中心大厦 Shanghai Tower	Normal	上海/ 中心/ 大厦 Shanghai/ Center/ Building
	Fine	上海/ 中心/ 大厦
	Coarse	上海中心大厦 Shanghai Tower
	Index	上海/ 中心/ 大厦/ 上海中心/ Shanghai Tower 中心大厦/ Centre 上海中心大厦
欧洲联盟 European Union	Normal	欧洲/ 联盟 Europe/ Union
	Fine	欧洲/ 联盟
	Coarse	欧洲联盟 European Union
	Index	欧洲/ 联盟/ 欧洲联盟
中国科学技术大学 USTC	Normal	中国/ 科学技术/ 大学 China/ Sci. n Tech/ University
	Fine	中国/ 科学/ 技术/ 大学 China/ Sci./ Tech/ University
	Coarse	中国科学技术大学 USTC
	Index	中国/ 科学技术/ 大学/ 科学/ 技术/ 中国科学技术大学

Table 3: The multi-granularity case study.

4.2 Main Results

Overall Performance Table 2 reports the overall performance. For the sake of fairness, we utilize a unified model and average F1 scores of six individual test sets (Luo et al., 2019). BERT-CRF stands out as compared to traditional methods due to the powerful representation capacity of the pre-trained language model. Following the PLM paradigm, (Tian et al., 2020b,a) further fuses wordhood information into the network, and achieves better performance compared to BERT-CRF. For simplicity, we set the BERT-CRF architecture as the cohort in our implementation to verify the gain effect of our framework. As shown in Table 2, our approach further advances BERT-CRF with cohort training and versatile decoding without reshaping model architecture, which also defeats the most recent SOTA method WMSeg (Tian et al., 2020b).

Multi-grained Segmentation We evaluate CWSeg on four different segmentation modes. As shown in Table 3, compared to the model without versatile decoding, CWSeg can better capture the whole words of the entity. This also illustrates the granularity gap between annotated corpora and the application scenarios. With versatile decoding, CWSeg can generate both fine-grained and coarse-grained labels. And multi-granularity results provide more knowledge and indexing, which is crucial for multiple scenarios such as retrieval, content recommendation, and advertisement.

4.3 Ablation Study

We investigate the impact of versatile decoding, cohort training, and different losses on CWSeg.

Effect of Versatile Decoding Table 4 details the performance gain of our approach in the domain adaption. It enables models to be readily applied to new domains without training. Take MSR for instance, our approach lifts the model performance by a large margin of 7%. This is reasonable as MSR has significantly different distributions compared

Train	Test	Methods	F1
All w/o AS	AS	CWSeg	**96.91**
		w/o Versatile	96.88 (-0.03)
All w/o CityU	CityU	CWSeg	**92.48**
		w/o Versatile	91.41 (-1.07)
All w/o CTB6	CTB6	CWSeg	**89.21**
		w/o Versatile	89.17 (-0.04)
All w/o MSR	MSR	CWSeg	**92.26**
		w/o Versatile	85.26 (-7.00)
All w/o PKU	PKU	CWSeg	**92.58**
		w/o Versatile	90.56 (-2.02)
All w/o Weibo	Weibo	CWSeg	**87.73**
		w/o Versatile	86.01 (-1.72)

Table 4: The effect of versatile decoding by cross-domain experiments. 'All w/o AS' means all datasets after removing AS. 'w/o Versatile' refers to the CWSeg model without the versatile decoding module.

to others as shown in Table 1, and thus requires the domain-adaptive decoding strategy.

PLM Settings		SN		MD	CH	
Net1	Net2	Net1	Net2	Net2	Net1	Net2
BERT-4	BERT-1	96.31	93.85	94.04	**96.9**	**94.84**
NEZHA-4	NEZHA-1	96.83	94.39	94.87	**97.03**	**95.37**

Table 5: The effect of cohort training experiments on CTB6 (F1). 'MD' for model distillation of Net1 distills Net2, 'SN' for single training, and 'CH' for cohort training. 'BERT-4' means the first 4 layers of the BERT base model.

Effect of Cohort Training Overall, the cohort training outperforms the classical model distillation approach in terms of small models as evidenced by Net2 (94.84 vs 94.04 and 95.37 vs 94.87) in Table 5. It's worthwhile to note that big models also benefit from the cohort training as compared to the independent training (e.g., Net1: 96.9 vs 96.31 and 97.03 vs 96.83). In this setting, the CH training policy, which is trained only once and converges faster, is about 3 times faster than MD, which requires 3 stages of training (Train Net1, train Net2, Net1 distills Net2).

Effect of Cohort Settings To study the effect of the cohort settings, we conducted a detailed analysis. As shown in Table 6, we can easily find that: (1) The cohort setting stands out in all trials, and the small model improves more significantly. (2) Larger models improve small models better. (3) Diversity in cohort settings promotes performance.

Effect of Wasserstein Distance For the cohort training, we further study the impact of mimicry loss. Specifically, we compare WD with KL and JS as detailed in Table 7 and Fig. 4. WD is slightly better than both KL and JS in large part due to the performance ceiling, whereas it can significantly accelerate cohort training by multiple folds. This is appealing, especially for multiple large-scale model learning.

	BERT-1	BERT-4	BERT-8	NZ-1	NZ-4
F1	93.85	96.31	96.97	94.39	96.83

(a) Single model training settings. 'NZ' for NEZHA.

	BERT Cohort				NZ Cohort	
PLM	BERT-1	BERT-4	BERT-1	BERT-8	NZ-1	NZ-4
F1	94.84 (+0.99)	96.9 (+0.59)	94.88 (+1.03)	97.31 (+0.34)	95.37 (+0.98)	97.03 (+0.20)

(b) Cohort training settings with the same backbone.

	BERT and NEZHA Cohort			
PLM	BERT-1	BERT-4	NZ-1	NZ-4
F1	94.85 (+1.00)	96.91 (+0.60)	95.51 (+1.12)	97.16 (+0.33)

(c) Cohort training settings with different backbones.

Table 6: The effect of cohort setting experiments.

	KL	JS	WD
BERT-1	94.39	94.48	**94.56**
BERT-2	95.81	95.82	**96.02**

Table 7: The effect of Wasserstein distance loss. 'WD' for Wasserstein distance.

4.4 Trade-off between Performance and Speed

We experiment with cohort training (CH) of BERT-1, BERT-4, BERT-8, and BERT-12. As a comparison, these 4 single networks (SN) are also fine-tuned independently. The latency for CH and SN is the same, and the units of latency are defined in Section 4.2. As shown in Fig. 5, overall, CH produces a batch of different model artifacts simultaneously as designed, which outperforms counterparts of SN without inference latency penalty. For example, CH-4 setting has almost the same segmentation performance as SN-12. These artifacts can serve different inference scenarios. Specifically, CH-1 can be used for real-time demanding applications and CH-12 works well on the offline inference scenarios with more tolerance of latency.

5 Discussion

Our latency comparisons are benchmarked on the same CPU device, while deep methods run faster on CUDA devices. Besides, we can resort to a fast-compiling language (e.g., C++) backed platform

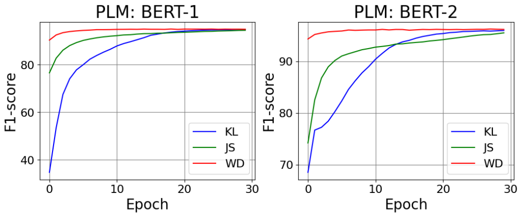


Figure 4: Loss convergence comparison for BERT-1 and BERT-2 in cohort settings.

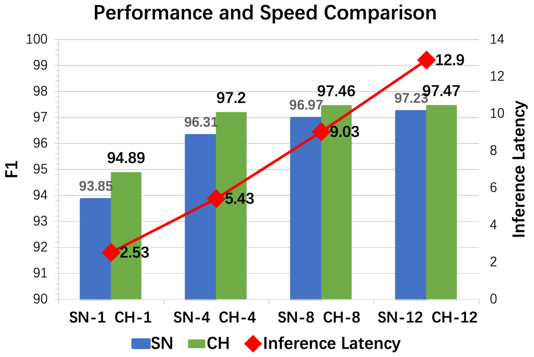


Figure 5: The trade-off of performance and speed.

or tailored toolchain (e.g., ONNX) to optimize the serving speed. How to apply diversity modes to different scenarios? Generally speaking, the coarse mode is to perceive complete semantics, and the fine mode is to perceive more extensive concepts. For example, in the scenarios of search and recommendation, the normal or coarse mode is employed to process web pages to build inverted indexes. Index mode is often used for query expansion, where we disassemble queries into multiple granularities to maximize recall of relevant documents.

6 Conclusion

In this work, we develop an efficient and general framework, CWSeg, which enables the state-of-the-art schemes of Chinese word segmentation better prepared for industrial deployment scenarios. We present Wasserstein distance-based cohort learning method and versatile decoding to facilitate the trade-off between segmentation performance and serving latency as well as the fast cross-domain adaption. Comprehensive experiments are performed to justify the efficiency and generalization of CWSeg. We believe that our work can be extrapolated to other sequence labeling problems straightforwardly.

Limitations

This study has potential limitations. When the CWSeg model is applied to a new domain, we assume that words and phrases solely related to the domain are available.

References

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.

Jimmy Ba and Rich Caruana. 2014. Do deep nets really need to be deep? *Advances in neural information processing systems*, 27.

Fuguang Bao, Wenqian Xu, Yao Feng, and Chonghuan Xu. 2022. A topic-rank recommendation model based on microblog topic relevance & user preference analysis. *Hum.-Cent. Comput. Inf. Sci*, 12(10).

bert-base chinese. bert-base-chinese. https://huggingface.co/bert-base-chinese.

chinese-roberta wwm. chinese-roberta-wwm. https://github.com/ymcui/Chinese-BERT-wwm.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. 2020. Revisiting pre-trained models for chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 657–668.

Yiming Cui, Ziqing Yang, and Xin Yao. 2023. Efficient and effective text encoding for chinese llama and alpaca.

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.

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

Chen Gong, Zhenghua Li, Min Zhang, and Xinzhou Jiang. 2017. Multi-grained chinese word segmentation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 692–703.

Chen Gong, Zhenghua Li, Bowei Zou, and Min Zhang. 2020. Multi-grained chinese word segmentation with weakly labeled data. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2026–2036.

Xuehua Han, Juanle Wang, Min Zhang, and Xiaojie Wang. 2020. Using social media to mine and analyze public opinion related to covid-19 in china. *International Journal of Environmental Research and Public Health*, 17(8):2788.

Kaiyu Huang, Degen Huang, Zhuang Liu, and Fengran Mo. 2020a. 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)*, pages 3873–3882.

Kaiyu Huang, Junpeng Liu, Degen Huang, Deyi Xiong, Zhuang Liu, and Jinsong Su. 2021. Enhancing chinese word segmentation via pseudo labels for practicability. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4369–4381.

Weipeng Huang, Xingyi Cheng, Kunlong Chen, Taifeng Wang, and Wei Chu. 2020b. Towards fast and accurate neural chinese word segmentation with multi-criteria learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2062–2072.

jieba. jieba. https://github.com/fxsjy/jieba.

Xiaoguang Li Wenyong Huang Yi Liao Yasheng Wang Jiashu Lin Xin Jiang Xiao Chen Qun Liu Junqiu Wei, Xiaozhe Ren. 2019. Nezha: Neural contextualized representation for chinese language understanding. *arXiv preprint arXiv:1909.00204*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Zhongguo Li and Maosong Sun. 2009. Punctuation as implicit annotations for chinese word segmentation. *Computational Linguistics*, 35(4):505–512.

Cheng-Lin Liu, Masashi Koga, and Hiromichi Fujisawa. 2002. Lexicon-driven segmentation and recognition of handwritten character strings for japanese address reading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(11):1425–1437.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Jinghui Lu, Rui Zhao, Brian Mac Namee, and Fei Tan. 2023. Punifiedner: A prompting-based unified ner system for diverse datasets.

Ruixuan Luo, Jingjing Xu, Yi Zhang, Xuancheng Ren, and Xu Sun. 2019. Pkuseg: A toolkit for multi-domain chinese word segmentation. *arXiv preprint arXiv:1906.11455*.

Mieradilijiang Maimaiti, Yang Liu, Yuanhang Zheng, Gang Chen, Kaiyu Huang, Ji Zhang, Huanbo Luan, and Maosong Sun. 2021. Segment, mask, and predict: Augmenting chinese word segmentation with self-supervision. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2068–2077.

NEZHA-Base-WWM. Nezha-base-wwm. https://github.com/huawei-noah/Pretrained-Language-Model.

PKUSeg. Pkuseg. https://github.com/lancopku/PKUSeg-python.

pyhanlp. pyhanlp. https://github.com/hankcs/pyhanlp.

Xipeng Qiu, Hengzhi Pei, Hang Yan, and Xuanjing Huang. 2019. A concise model for multi-criteria chinese word segmentation with transformer encoder. *arXiv preprint arXiv:1906.12035*.

Xipeng Qiu, Peng Qian, and Zhan Shi. 2016. Overview of the NLPCC-ICCPOL 2016 shared task: Chinese word segmentation for micro-blog texts. In *Proceedings of The Fifth Conference on Natural Language Processing and Chinese Computing & The Twenty Fourth International Conference on Computer Processing of Oriental Languages*.

Ludger Rüschendorf. 1985. The wasserstein distance and approximation theorems. *Probability Theory and Related Fields*, 70(1):117–129.

SIGHAN2005Bakeoff. Sighan2005bakeoff. http://sighan.cs.uchicago.edu/bakeoff2005/.

Xu Sun, Houfeng Wang, and Wenjie Li. 2012a. Fast online training with frequency-adaptive learning rates for Chinese word segmentation and new word detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 253–262, Jeju Island, Korea. Association for Computational Linguistics.

Xu Sun, Houfeng Wang, and Wenjie Li. 2012b. Fast online training with frequency-adaptive learning rates for chinese word segmentation and new word detection. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 253–262.

Fei Tan, Yifan Hu, Changwei Hu, Keqian Li, and Kevin Yen. 2020. TNT: Text normalization based pretraining of transformers for content moderation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4735–4741, Online. Association for Computational Linguistics.

THULAC. Thulac. https://github.com/thunlp/THULAC-Python.

Yuanhe Tian, Yan Song, Xiang Ao, Fei Xia, Xiaojun Quan, Tong Zhang, and Yonggang Wang. 2020a. Joint chinese word segmentation and part-of-speech tagging via two-way attentions of auto-analyzed knowledge. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8286–8296.

Yuanhe Tian, Yan Song, Fei Xia, Tong Zhang, and Yonggang Wang. 2020b. Improving chinese word segmentation with wordhood memory networks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8274–8285.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2):207–238.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Improving neural machine translation with conditional sequence generative adversarial nets. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1346–1355.

Yuan Yao, Lorenzo Rosasco, and Andrea Caponnetto. 2007. On early stopping in gradient descent learning. *Constructive Approximation*, 26(2):289–315.

Shaohua Zhang, Haoran Huang, Jicong Liu, and Hang Li. 2020. Spelling error correction with soft-masked bert. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 882–890.

Ying Zhang, Tao Xiang, Timothy M Hospedales, and Huchuan Lu. 2018. Deep mutual learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4320–4328.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. Ernie: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451.

A Appendix

A.1 Model Details

Cohort Model We set the SOTA CWS model architecture BERT-CRF as cohort model implementations to exploit the PLM strength and transition patterns of the labeling system.

For each character x_i is mapped to $\mathbf{x}_i \in \mathbb{R}^{d_e}$, where d_e is the embedding size. The PLM encoder extract the contextual features $\mathbf{h}_i \in \mathbb{R}^{d_h}$ automatically for each character x_i by

$$[\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_{|\mathcal{X}|}] = Encoder(\mathbf{X}), \qquad (6)$$

where $\mathbf{X} \in \mathbb{R}^{d_e \times |\mathcal{X}|}$ is the embedding matrix of $\mathcal{X}$, d_h is the size of hidden features. There are several prevalent choices for $Encoder$ model, such as BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019).

There are rules in the labeling systems, such as the I can only be after the B label. We thus utilize the conditional random fields (CRF) (Lafferty et al., 2001) to model the transition patterns, which can be formulated as:

$$p(y_i|x_i) = \frac{exp(\mathbf{W}_c\mathbf{W}_o^\top \mathbf{h}_i + \mathbf{b}_c)}{\sum_{y_{i-1}y_i} exp(\mathbf{W}_c\mathbf{W}_o^\top \mathbf{h}_i + \mathbf{b}_c)}, \qquad (7)$$

where $\mathbf{W}_o \in \mathbb{R}^{d_h \times |\mathcal{T}|}$, $\mathbf{W}_c \in \mathbb{R}^{|\mathcal{T}| \times |\mathcal{T}|}$, and $\mathbf{b}_c \in \mathbb{R}^{|\mathcal{T}|}$ are training parameters to model the transition from y_{i-1} to y_i.

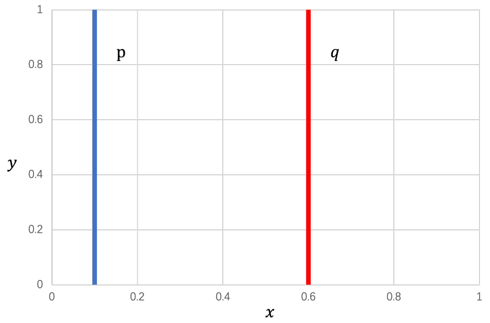


Figure 6: Suppose two probability distributions P and Q. $\forall(x,y) \in P, x = 0, y \sim U(0,1)$; $\forall(x,y) \in Q, x = \theta, 0 \leq \theta \leq 1, y \sim U(0,1)$.

Wasserstein Distance As shown in Fig. 6, there is no overlap between P and Q when $\theta \neq 0$, and:

$$\begin{aligned}
&KL(P||Q) = \sum_{x=0,y\sim U(0,1)} 1 \cdot log\frac{1}{0} = +\infty, \\
&KL(Q||P) = \sum_{x=\theta,y\sim U(0,1)} 1 \cdot log\frac{1}{0} = +\infty, \\
&JS(P,Q) = \\
&\frac{1}{2}(\sum_{x=0,y\sim U(0,1)} 1 \cdot log\frac{1}{\frac{1}{2}} + \sum_{x=0,y\sim U(0,1)} 1 \cdot log\frac{1}{\frac{1}{2}}) \\
&= log2, \\
&W(P,Q) = |\theta|,
\end{aligned} \qquad (8)$$

when $\theta = 0$:

$$\begin{aligned} &KL(P||Q) = KL(Q||P) = JS(P,Q) = 0, \\ &W(P,Q) = 0 = |\theta|, \end{aligned} \tag{9}$$

where $KL(\cdot)$ gives infinity when two distributions are disjoint, and $JS(\cdot)$ is always a constant. And they are both equal to 0 when $\theta = 0$, so they both have a sudden jump at $\theta = 0$. While the Wasserstein distance provides a smooth measure, which contributes to stable gradient descents.

Versatile Decoding Pseudocode ExtractBorders aims to obtain the border indices of the prediction, such as the borders of “中国 / 科学技术 / 大学” is [0, 2, 6, 8]. CutBorders is designed to filter out the candidates in $\mathcal{C}$ that cross the borders, such as “中国科学技术大学” will be filtered out, and “科学” “技术” will be preserved. LinkBorders is designed to obtain all candidates in $\mathcal{C}$ that match one-skip or multi-skip borders, such as “中国科学技术大学” will be preserved for it skip two borders [2, 6].

```
# extract borders of the segmented token_list
def extract_borders(token_list):
    borders = set()
    for token in token_list:
        borders.add(token.start_offset)
        borders.add(token.end_offset+1)
    return borders

# find candidates that no-cross borders
def cut_borders(token_list, borders):
    cut_borders = []
    cross_border = False
    for token in token_list:
        cross_border = False
        for idx in range(token.start_offset+1,
             token.end_offset+1):
            if idx in borders:
                cross_border = True
                break
        if not cross_border:
            cut_borders.append(token)
    return cut_borders

# find all candidates in token_list that match
     one-skip or multi-skip borders
def link_borders(token_list, borders):
    link_borders = []
    for token in token_list:
        if token.start_offset in borders and
             (token.end_offset+1) in borders:
            link_borders.append(token)
    return link_borders
```