Word Segmentation by Separation Inference for East Asian Languages

Yu Tong[†], **Jingzhi Guo**[†*], **Jizhe Zhou**[‡*], **Ge Chen**[§], **Guokai Zheng**[§]
[†]Department of Computer Science, University of Macau, Macau, China
[‡]Department of Computer Science, Sichuan University, China
[§]vivo AI Lab, Shenzhen, China
[†‡]{yb87462, jzguo, yb87409}@umac.mo
[§]{gechen.nlp, gkzheng.nlp}@gmail.com

Abstract

Chinese Word Segmentation (CWS) intends to divide a raw sentence into words through sequence labeling. Thinking in reverse, CWS can also be viewed as a process of grouping a sequence of characters into a sequence of words. In such a way, CWS is reformed as a separation inference task in every adjacent character pair. Since every character is either connected or not connected to the others, the tagging schema is simplified as two tags "Connection" (C) or "NoConnection" (NC). Therefore, bigram is specially tailored for "C-NC" to model the separation state of every two consecutive characters. Our Separation Inference (SpIn) framework is evaluated on five public datasets, is demonstrated to work for machine learning and deep learning models, and outperforms state-of-the-art performance for CWS in all experiments. Performance boosts on Japanese Word Segmentation (JWS) and Korean Word Segmentation (KWS) further prove the framework is universal and effective for East Asian Languages. [1]

1 Introduction

In Natural Language Processing (NLP), word segmentation is the commencement of Part-of-Speech (POS) tagging, semantic role labeling (SRL), and other similar studies. Particularly for Chinese, Japanese and Korean languages, the absence of explicit boundaries between characters makes the Word Segmentation (WS) task indispensable in NLP tasks. Dominant word segmentation methods considered WS as a sequence tagging task (Xue, 2003). Various tagging schemas such as "BMES" (Begin, Middle, End, Single), "BIES" (Begin, Inside, End, Single), "SEP-APP" (Separate, Append), "BI" (Begin, Inside), and "START-NONSTART" were employed to tackle the sequence labeling task. These tagging schemas are all character-based and summarized as four-tags ("BMES", "BIES") and two-tags ("SEP-APP", "BI" "START-NONSTART"). Despite diverse tagging schemas, they all carry implicit position information. For four-tags tagging schemas, the implicit information restricts the transition between tags. Take "BMES" as an example; tag "B" can not be followed by "B" or "S". These two tagging schemas heavily rely on the precise prediction of the relative position of each character in one segment. However, the exact position information is not essential for the WS task. Any unreasonable inner prediction representing the character's relative position results in incorrect segmentation, although the correct boundary prediction. There is no limitation of tag-to-tag transition for the two-tags schema, but according to common sense, the first character of a sentence must be predicted as "SEP", "B" or "START". The implicit constraint of position for the first tag of the sentence still exists. It is necessary to ensure the prediction accuracy of the first tag during the inference. Therefore, CRF is required to revise unreasonable tag-to-tag transitions and learn the implicit restriction including the first tag of a sentence. The CRF has alleviated the unreasonable tag prediction to some degree, but the simultaneous learning of transition and emission matrix still results in the tag inference being intractable. Current works attempt to complicate the network (Chen et al., 2017; Tian et al., 2020) and introduce more information (Cai et al., 2017) such as rich context, linguistic and extra knowledge to tackle the abovementioned problem. However, the intrinsic problem, which is the implicit restriction of the position in the existing tagging schemas, is not well solved. In this paper, we propose "Connection(C)-No-Connection(NC)", which targets on character-to-character connections, to deal with the WS task directly. "C-NC" is independent of the previous state, and there is no dependency between states.

*Corresponding authors.
[1]Our source code will be released as soon as possible at https://github.com/UM-NLPer/SpIn-WS.

Findings of the Association for Computational Linguistics: ACL 2022, pages 3924 - 3934

Moreover, there is no restriction for the first state as it is located between the first and the secondary characters. It can be either "C" or "NC". "C" or "NC" is a binary classification task. Therefore, CRF is not required and can then be substituted with a classification network. The tag-transition and implicit restriction burdens can be substantially alleviated through such "C-NC" states. Because "C-NC" describes the connection state between two adjacent characters, we employ bigram features to cooperate with the "C-NC". Compared with existing tagging schemas, which are character-based and the bigram features are considered as extra information, the bigram features in SpIn are the basic **processing unit**. Therefore, a brand-new Separation Inference (SpIn) framework is proposed and constructed on the bigram features and the classification layer. Sliding one-after-one along all the bigrams, words are yielded by allocating "C" and "NC" tags in the interval of characters. SpIn significantly reduces the inference complexity (inference layer CRF is degraded as the softmax network); dispels extra context information (merely bigram feature is in consideration); and gains competitive performance of CWS on the machine learning in contrast with the deep learning models. Besides its effectiveness on Chinese Word Segmentation, our extensive experiments also verify the universality by attaining state-of-the-art (SOTA) performance in Japanese and Korean Word Segmentation benchmark tests. Our contributions are summarized as follows:

- SpIn provides a new tagging schema from a novel perspective and solves the intrinsic problems of the existing tagging schemas.
- SpIn is a universal framework that gains state-of-the-art performance on the Word Segmentation task in East Asian Languages.
- The SpIn framework is also suitable for machine learning models and has achieved competitive results.

2 Related Work

Researchers have explored the CWS task from various directions since the 1990s (Sproat et al., 1996). Widely applied methodologies considered it as the sequence tagging task based on various label schemas. CWS was first treated as a sequence tagging task in (Xue, 2003). The Maximum Entropy (Low et al., 2005) model and the CRF (Lafferty et al., 2001) were the most adopted sequence tagger. There are two main problems in the WS task: the ambiguities and the Out-of-Vocabulary (OOV) words. Researchers tried to leverage extra context features such as the bigram (Zhao et al., 2006; Chen et al., 2015; Pei et al., 2014; Yang et al., 2017; Zhang et al., 2013) and the word features (Morita et al., 2015; Zhang et al., 2016; Zhang and Clark, 2007) to tackle word ambiguities and improve the model's generalization capability. Moreover, language-specific knowledge such as dictionaries was employed (Sun and Xu, 2011) for better CWS. Extra punctuation marks from large manually segmented corpus were introduced to the learning model and proved effective for solving the unknown words (Li and Sun, 2009). Meanwhile, the external knowledge was explored through the semi-supervised models for better segmentation (Sun and Xu, 2011; Wang et al., 2011; Liu and Zhang, 2012; Zhang et al., 2013). Along with the development of pre-trained models like BERT (Devlin et al., 2018), ELMo (Peters et al., 2018), and GPT (Radford et al., 2018), striking improvements on CWS are observed by replacing the feature extraction layer with these powerful pre-trained models. Except for the investigation of the effect of features, various tagging schemas were also discussed. Widely applied tagging schema in CWS contains "BMES" (Meng et al., 2019; Huang et al., 2020; Yang et al., 2019, 2017), "BIES" (Ma et al., 2018), "SEP-APP" (Zhang et al., 2016, 2018; Yan et al., 2020), "BI" (Lee and Kim, 2013), and "START-NONSTART" (Tseng et al., 2005; Peng et al., 2004). There is either the limitation of tag-to-tag transitions or the implicit constraint for the first tag for these tagging schemas. These inherent problems were not well solved. Hence, we propose the SpIn framework constructed on the "C-NC" tagging schema and its specially tailored bigram features. SpIn eliminates the implicit restriction of existing tagging schemas and boosts the performance of the WS task in East Asian languages.

3 Proposed Method

We propose adopting the bigram feature to adapt to the "C-NC" tagging schema to model the connection of adjacent characters. Distinguished from character-based models leveraging bigram feature as extra information, merely bigram is employed

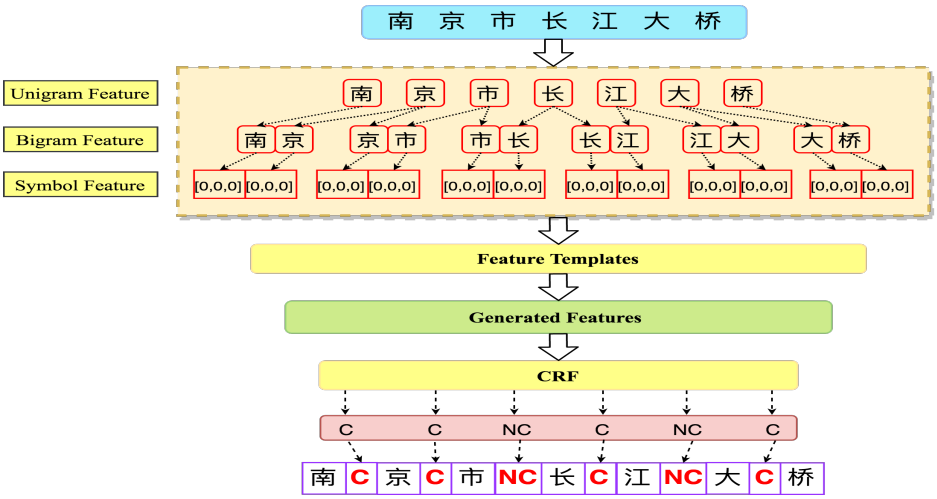

Figure 1: The figure is the architecture of SpIn applied to the machine learning model. The features are constructed based on the bigram and symbol features by applying the feature templates.

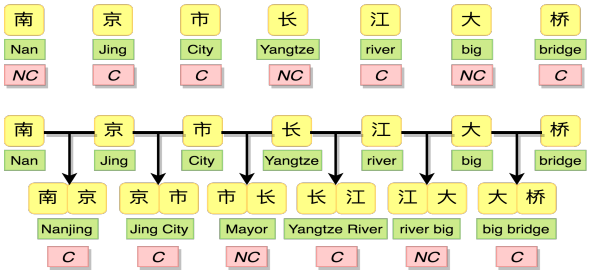

Figure 2: The figure is the comparison between the traditional two-tags tagging schema and "C-NC". The traditional two-tags tagging schema (upper) is tagged on the character. However, "C-NC" (bottom) is located in the interval between the characters.

and set up as **input unit**. Adaptation of SpIn involves machine and deep learning models. Figure 1 and Figure 5 summarize the SpIn framework architecture adapted to the machine and deep learning models separately.

Before exploring the structure of SpIn, we firstly elaborate definition of the proposed "C-NC" and distinction with the traditional two-tags tagging schema that indicates whether the current character is the boundary or not. In the later part of this section, we present the detailed structure of SpIn, including how to apply the SpIn framework to the machine learning and deep learning models. For machine learning, we explain how to build features based on the bigram through applying feature templates. Meanwhile, we present how to build the bigram features based on the feature extractor layer for the deep learning model. In the last subsection, we illustrate the inference layer.

3.1 Connection and No-Connection Tagging Schema

Tags "Connection" and "No-Connection" are proposed to model whether two adjacent characters (bigram) are in the same segment or not. If two characters in the bigram are not in the same segment, the corresponding label is "NC"; otherwise, the tag is "C".

Borrow "C-NC" to model traditional two-tags tagging schema indicating the current character as the beginning of a word or the continuation. The tagging procedure is illustrated in the upper section in Figure 2. By contrast, "C-NC" represents the connection state of two adjacent characters as illustrated in the lower section. Comparison between traditional two-tags and "C-NC" is summarized from three aspects:

- Traditional two-tags tagging schemas are labeled on each character. However, the tag "C" or "NC" is located in the interval between two characters.
- The total number of tags of "C-NC" is one less than the traditional two-tags tagging schema.
- The implicit restriction of the first character in a sentence exists for the traditional tagging schema. In contrast, there is no limitation of the first state for the "C-NC".

3.2 Feature Templates for Machine Learning

Feature engineering directly results in the model performance for machine learning models. Therefore, we leverage the bigrams and symbol information to enrich features by applying feature templates. We define the feature templates below:

Type	Feature	Example	Description
bigram	current_bigram	市长(Mayor)	the current bigram
unigram	bigram_head	市(City)	the head token of the current bigram
unigram	bigram_tail	长(Yangtze)	the tail token of the current bigram
date,digit,letter	bigram_head.is_symbol	[0,0,0]	whether the head token is a symbol or not
date,digit,letter	bigram_tail.is_symbol	[0,0,0]	whether the tail token is a symbol or not
bigram	pre_bigram	京市(Jing City)	the previous bigram of the current bigram
date,digit,letter	pre_bigram.is_symbol	[0,0,0]	whether the previous bigram is a symbol or not
bigram	pre_pre_bigram	南京(Nanjing)	the previous bigram of the previous bigram
date,digit,letter	pre_pre_bigram.is_symbol	[0,0,0]	whether it is a symbol or not
bigram	next_bigram	长江(Yangtze River)	the next bigram of the current bigram
date,digit,letter	next_bigram.is_symbol	[0,0,0]	whether the next bigram is a symbol or not
bigram	next_next_bigram	江大(River Big)	the next bigram of the next bigram
date,digit,letter	next_next_bigram.is_symbol	[0,0,0]	whether it is a symbol or not

Figure 3: The figure is the explanation of the element features.

Feature	Example	Description
Feature(0)	市长+市+长+[0,0,0]+[0,0,0]	represents the feature of the current bigram
Feature(-1)	京市+[0,0,0]	represents the feature of the previous bigram
Feature(-2)	南京+[0,0,0]	represents the feature of the previous bigram of the previous bigram
Feature(+1)	长江+[0,0,0]	represents the feature of the next bigram
Feature(+2)	江大+[0,0,0]	represents the feature of the next bigram of the next bigram

Figure 4: The figure is the explanation of generated features through applying feature templates.

- Feature(0) = current_bigram + bigram_head + bigram_tail + bigram_head.is_symbol + bigram_tail.is_symbol
- Feature(-1) = pre_bigram + pre_bigram.is_symbol
- Feature(-2) = pre_pre_bigram + pre_pre_bigram.is_symbol
- Feature(+1) = next_bigram + next_bigram.is_symbol
- Feature(+2) = next_next_bigram + next_next_bigram.is_symbol

Figure 3 explains the element feature. The symbol feature is a one-dimensional array. It indicates whether the character belongs to symbols or not. The symbols include the date, digit, or letter. Figure 4 illustrates the generated features through applying feature templates for the current bigram. The final features are the concatenation of Feature(0), Feature(-1), Feature(-2), Feature(+1) and Feature(+2).

3.3 Feature Extraction Layer

As recent state-of-the-art results on CWS tasks are achieved by applying BERT (Devlin et al., 2018) as the feature extraction layer, we follow the same step. Moreover, we customize the feature by adding the additional symbol feature. Through symbol projection, each character is project into a one-dimensional array such as $[0, 0, 1]$, each position represents $[date, digit, letter]$. This case indicates that the current character belongs to letter. Followed by an activate function ReLU, symbol embedding is generated with the vector size of 3 and denoted as S_n. The character embedding generated from BERT is a 768-dimensional vector (denoted as c_n) and is resized as $(768 + 3)$ through concatenating with symbol embedding. The customized character embedding is represented as e_n. Two adjacent character embeddings with their symbol embeddings are concatenated as bigram features. Hence, the corresponding bigram features (denoted as b_n) are the size of $(768 + 3) * 2$. Two Fully Connected layers follow the constructed bigram features. The CRF layer (or softmax layer) is em-

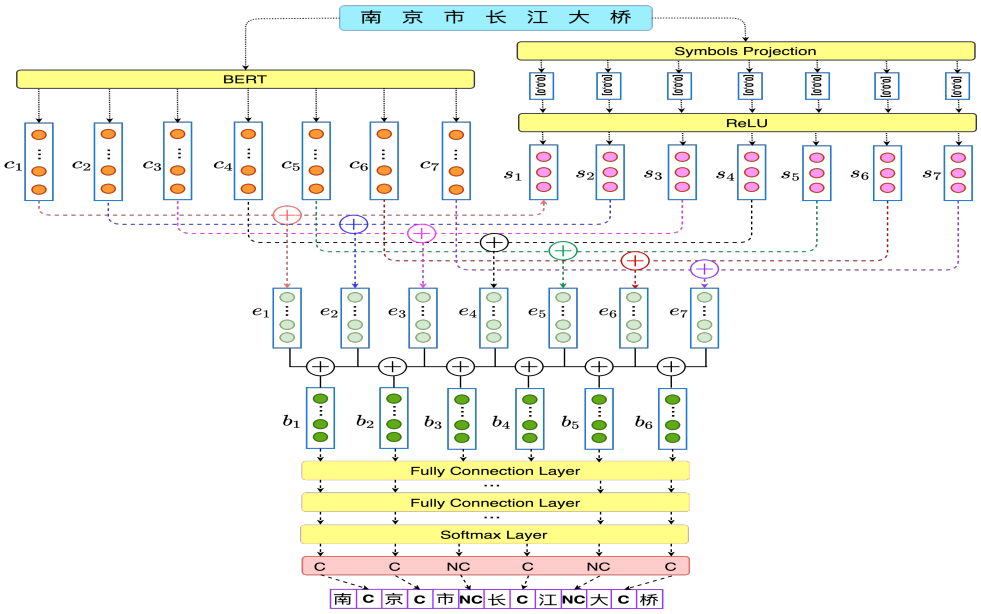

Figure 5: The architecture of SpIn applied to the deep learning model. Orange circles below the BERT are the unigram features for each character. Pink circles are the symbol features generated through symbols projection and a ReLU activation function. "+" is the concatenation operation. The unigram features concatenate with symbol features. Dark green circles are bigram features generated after concatenating every two light green circles.

ployed as the inference layer. The architecture of SpIn that is applied to the deep learning model is shown in Figure 5.

3.4 Inference Layer

Following previous work (Tseng et al., 2005; Peng et al., 2004), the CRF (Lafferty et al., 2001) layer is adopted as an inference layer for the machine learning model for a fair comparison. The CRF tries to find the optimal tag sequence $Y^{'}$ regarding the input sequence X where:

$$Y^{'} = \arg\max_{Y \in L^n} P(Y|X) \tag{1}$$

$$P(Y|X) = \frac{1}{Z(x)} \exp(\sum_{i,k} \lambda_k t_k(y_{i-1}, y_i, x, i) + \sum_{i,l} \mu_l s_l(y_i, x, i)) \tag{2}$$

L^n are all the possible tag sequences, Z is the normalization factor, t_k, s_l are status feature function and λ_k, μ_l are trainable parameters.

4 Experiments

Evaluation is first conducted on the CWS to prove the SOTA performance of SpIn. Contrast experiments involve both machine learning and deep learning models for further demonstrating the robustness of SpIn. An ablation study is conducted to investigate the effect of each component.

4.1 Datasets

Five Chinese word segmentation datasets are evaluated in the experiments, including Chinese Penn Treebank 6.0 (CTB6) (Xue et al., 2005) and CITYU, AS, PKU, MSR from SIGHAN 2005 bakeoff task (Emerson, 2005). PKU, MSR, and CTB6 are simplified Chinese, and the other two AS and CITYU are traditional Chinese.

4.2 Evaluation of Machine Learning Model

4.2.1 Parameters & Evaluation Metrics

We set L-BFGS as the optimization algorithm for the CRF layer. The L1-norm is 0.598, and the L2-norm is 0.0323. The maximum iterations are 150. Following the widely accepted evaluation methodologies, the F1 score is adopted as the metric for exhibiting reliability.

4.2.2 Experiment Results

The evaluation results of SpIn adapted to the machine learning model are listed in Table 1. For a fair comparison, the baseline is selected from the paper in which the machine learning model is applied. Compared with the baseline which is the best result of Bakeoff2005 [2], SpIn achieves a significant improvement up to **+1.3%** F1 score on the AS dataset. Likewise, SpIn performs better on all similar longitudinal comparisons conducted on the CITYU and MSR datasets.

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

	CITYU	AS	PKU	MSR	CTB6
Baseline	94.3	95.2	95.0	96.4	-
SpIn_ML	**95.5**	**96.5**	94.6	**96.5**	**96.0**
	+1.2	+1.3	-0.4	+0.1	-

Table 1: SpIn of Machine Learning version (SpIn_ML) v.s. the best results of SIGHAN 2005 Bakeoff. The F1 score is employed as the metric.

	CITYU	AS	PKU	MSR	CTB6
BMES	94.4	94.7	91.3	95.8	95.2
BIS	95.2	95.6	91.8	96.2	95.7
BI	93.5	93.3	93.5	95.1	93.6
C-NC	**95.5**	**96.5**	**94.6**	**96.5**	**96.0**

Table 2: "C-NC" v.s traditional tagging schemas. The F1 score is employed as the metric.

4.2.3 Ablation Study

As detailed in Figure 1 and Figure 5, the structure of the SpIn contains four main components: the "C-NC" tagging schema, the bigram features, the symbol features, and the inference layer. Since the CRF layer is a common approach and widely used in the era of machine learning as a decoder to restrict unreasonable tag transition, we exclude it in this ablation section and concentrate on the efficacy of the other three components. Our investigation is mainly carried out through:

- substituting "C-NC" with traditional tagging schemas;
- substituting bigram with unigram features;
- removing symbol features;

Substitution of "C-NC" Contrast experiments of tagging schemas are illustrated in Table 2. Keep bigram features, substitute "C-NC" with traditional "BMES", "BIS" and "BI" (equivalent to "START-NONSTART" and "SEP-APP") tagging schemas. Experiment conditions are set still. For adapting these three character-based tagging schemas, the bigram feature is considered rich context information for the current character. Each character feature is substituted with the bigram feature, representing the concatenation of the current and the previous character feature with their corresponding symbol feature. For the first character in the sentence, we put a "PAD" token to join the first character and form its bigram. The corresponding tag of the original character is labeled on the substituted bigram. The experiment results in Table 2 illustrate that "C-NC" does promote performance on all five datasets compared with traditional tagging schemas.

Substitution of Bigram Features Keep the "C-NC" tagging schema and conduct the contrast experiment to investigate the effect of features. Integrating "C-NC" with unigram features downgrades "C-NC" as "BI" or "START-NONSTART". The comparison between bigram and traditional unigram features is illustrated in Table 3. Although "C-NC" is employed, the traditional unigram feature performs worse than SpIn. Therefore, bigram is essential and specially tailored for our proposed "C-NC".

	CITYU	AS	PKU	MSR	CTB6
Unigram	86.5	88.0	86.5	87.1	89.6
Bigram	**95.5**	**96.5**	**94.6**	**96.5**	**96.0**

Table 3: unigram v.s. bigram features. The F1 score is employed as the metric.

	CITYU	AS	PKU	MSR	CTB6
W/O Symbols	94.6	95.4	92.7	96.1	93.4
Symbols	**95.5**	**96.5**	**94.6**	**96.5**	**96.0**

Table 4: with symbols v.s. without symbols. The F1 score is employed as the metric.

Substitution of Symbol Features Table 4 illustrates the effect of the symbol features. After employing the symbol features, the result is further pushed up to **+2.6%** F1 score on the CTB6 dataset. Symbol features promote the performance of SpIn on the CWS task. Hence, the symbol features are leveraged in the following experiments by default.

For the "C-NC" tagging schema, if unigram is adopted, it will be equivalent to "BI" or "START-NONSTART", and significant performance loss has been observed on all datasets. Similarly, the decline in F score has been observed after removing the symbol feature. In summary, **the whole framework contributes to the performance boosts instead of any component.**

4.3 Evaluation of Deep Learning Model

4.3.1 Parameters & Evaluation Metrics

The sequence length is 128; the learning rate is $2e-5$; batch size is 64, and the training epochs are 10. The early stop mechanism is introduced to avoid over-fitting. Adam is employed as the optimizer. All the parameters mentioned above are still set in the following experiments. Besides the F1 score, the recall of Out-of-Vocabulary words (R_oov) is a critical metric to evaluate the generalization of the word segmentation model. Hence, R_oov is also employed to prove SpIn is robust and effective for East Asian Languages. Besides the F1 and R_oov, we employ the **Standard Deviation (SD)** of five experiments to indicate model reliability.

	CITYU		AS		PKU		MSR		CTB6	
	F1	R_oov	F1	R_oov	F1	R_oov	F1	R_oov	F1	R_oov
Chen et al., 2017	95.6	81.40	94.6	73.50	94.3	72.67	96.0	71.60	96.2	82.48
Gong et al., 2019	96.2	73.58	95.2	77.33	96.2	69.88	97.8	64.20	97.3	83.89
Huang et al., 2020	97.6	87.27	96.6	79.26	96.6	79.71	97.9	83.35	97.6	87.77
Meng et al., 2019	97.9	-	96.7	-	96.7	-	98.3	-	-	-
Tian et al., 2020	97.8	87.57	96.58	78.48	96.51	**86.76**	98.28	**86.67**	97.16	88.00
Qiu et al., 2020	96.91	86.91	96.44	76.39	96.41	78.91	98.05	78.92	96.99	87.0
Ke et al., 2021	**98.20**	**90.66**	**97.01**	**80.89**	**96.92**	80.90	**98.50**	83.03	**97.89**	**89.21**
SpIn_DL	**98.6 (0.06)**	**90.68 (0.02)**	**97.5 (0.01)**	**81.36 (0.05)**	**98.0 (0.02)**	**93.53 (0.10)**	**98.7 (0.04)**	**93.13 (0.02)**	**98.6 (0.10)**	**93.90 (0.06)**

Table 5: SpIn of Deep Learning version (SpIn_DL) v.s. dominant deep neural methods on the CWS task. Values in the brackets are SD of five experiments.

	CITYU	AS	PKU	MSR	CTB6
BMES	97.7	96.8	96.3	97.7	97.2
BIS	98.1	97.1	96.8	98.1	97.5
BI	98.3	97.2	97.4	98.3	98.0
C-NC	**98.6**	**97.5**	**98.0**	**98.7**	**98.6**

Table 6: "C-NC" v.s. traditional tagging schemas. Refer to Table 5 for baseline. The F1 score is employed as the metric.

	CITYU	AS	PKU	MSR	CTB6
Unigram	98.3	97.3	97.7	98.4	98.3
Bigram	**98.6**	**97.5**	**98.0**	**98.7**	**98.6**

Table 7: bigram v.s unigram features. Refer to Table 5 for baseline. The F1 score is employed as the metric.

4.3.2 Experiment Results

The experiment results are reported in Table 5. SpIn brought an improvement up to **+1.08%** F1 score on the PKU dataset and at least **+0.2%** F1 score on the MSR dataset. Moreover, the best OOV performance observed on all five datasets shows the effectiveness of SpIn on OOV words. **+6.77%** improvement is achieved on the PKU dataset. The promotions on the OOV recall demonstrate the better generalization capability and robustness of SpIn.

Similar to the above experiments of the machine learning model, we also conduct the **ablation study** to evaluate the effects of different factors on the deep learning model as reported in Table 6, 7, 8, 9. The F1 score is employed in these four contrast experiments as the metric. The baseline refers to previous work mentioned in Table 5 from line 2 to line 8.

Bigram features are also applied as context features to adapt traditional tagging schemas. The bigram feature is generated by concatenating the current and the previous character feature with their corresponding symbol feature. Similarly, we add extra "PAD" for the first character to construct the first bigram feature. The corresponding tag of the original character is labeled on the bigram feature. The experiment results in Table 6 show that "C-NC" achieves the best performance. Therefore, in the situation of rich features, the "C-NC" tagging schema also works for deep learning models.

	CITYU	AS	PKU	MSR	CTB6
W/O Symbols	98.4	97.3	98.0	98.6	98.5
Symbols	**98.6**	**97.5**	98.0	**98.7**	**98.6**

Table 8: with symbols v.s. without symbols. Refer to Table 5 for baseline. The F1 score is the metric.

	CITYU	AS	PKU	MSR	CTB6
CRF	98.5	**97.5**	98.0	98.6	**98.6**
softmax	**98.6**	97.4	98.0	**98.7**	98.5

Table 9: softmax v.s. CRF as inference layer. Refer to Table 5 for baseline. The F1 score is the metric.

We also adapt the unigram feature to the "C-NC" tagging schema to follow the variable-controlling method. It makes "C-NC" the same as "BI". The contrast experiment between the bigram and the unigram feature is conducted. The results are shown in Table 7. In contrast with SpIn(ML), the bigram feature achieves insignificant improvement in SpIn(DL) because of rich pre-trained feature representation. Nevertheless, there are still **+0.3%** F1 score boosts are observed on CITYU, PKU, MSR, and CTB6 datasets.

Table 8 illustrates the effect of the symbol features for the deep neural model. In contrast with the results in Table 4, the symbol features are insignificant in result improvements. Nevertheless, **+0.2**% F1 score improvements are gained on CITYU and AS datasets. The reason for inconspicuous performance is that BERT simplifies feature engineering with its rich representation.

As SpIn eliminates the restriction of tag-to-tag transition and the first tag in a sentence, the softmax can further substitute the CRF. Table 9 illustrates that replacing the CRF with the softmax does not affect the performance. The competitive results are achieved with less complexity of the network.

4.4 Comparison of SpIn_DL and SpIn_ML

Table 11 illustrates the comparison between the SpIn_DL and SpIn_ML. The model size and response time are approximated to the nearest integer. The model size of SpIn_DL is four times as large as SpIn_ML. For SpIn_DL, model size depends

	BCCWJ	
	F1	R_oov
Kitagawa and Komachi, 2018	98.42	-
Higashiyama et al., 2019	98.93	-
BMES+Unigram	97.71	90.08
BIS+Unigram	98.17	91.73
BI+Unigram	98.39	92.51
SpIn	**98.94 (0.08)**	**93.01 (0.01)**

Table 10: SpIn v.s. dominant methods on JWS. Values in the brackets are SD of five experiments.

	Size	**Time (CPU)**	**F1 score**
SpIn_DL	400M	15000us/char	97.5
SpIn_ML	100M	30us/char	96.5

Table 11: SpIn_DL v.s. SpIn_ML.

on the network structure. However, for SpIn_ML, the model size depends on the scale of training data. We choose the AS (the largest dataset) from the five datasets to conduct the comparative experiment. Therefore, the maximum model size of SpIn_ML is near 100M. The inference process is performed on the empty CPU machine. We randomly select 2000 sentences from all datasets for testing. The sentence length is limited to $[10, 50]$. We conducted 10 experiments and get the average value. The speed of SpIn_ML is 500 times as fast as SpIn_DL. In contrast, the performance difference (F1 score) between SpIn_ML and SpIn_DL is only 1%.

4.5 Qualitative Analysis

Besides the academic studies, we also compare SpIn with the well-established commercial model LTP4.0 (Che et al., 2021). LTP4.0 leverages large training datasets. However, in this qualitative analysis, SpIn is merely trained on the smaller CTB6 dataset. In Figure 6, the ground truth agrees with SpIn for both sentences. The main issue focuses on the words "precalcining kiln" in the top sentence and "total failure" at the bottom. "Precalcining kiln" is a professional word leading to the out-of-vocabulary problem. The word "the whole chessboard" tends to be associated with "lose all" because the word is an idiom indicating "lose the whole chess game". These two featured cases reveal the generalization capacity of SpIn while handling biased samples.

5 Adaptation to Asian Languages

Japanese Word Segmentation (JWS) and Korean Word Segmentation (KWS) are evaluated on SpIn_DL to further prove SpIn is universal.

	KAIST		GSD	
	F1	R_oov	F1	R_oov
BMES+Unigram	87.62	78.34	87.12	78.27
BIS+Unigram	92.19	83.72	89.94	81.97
BI+Unigram	92.26	83.78	90.03	82.08
SpIn	**92.37 (0.04)**	**83.81 (0.08)**	**91.19 (0.09)**	**82.24 (0.12)**

Table 12: SpIn v.s. dominant methods on KWS. Values in the brackets are SD of five experiments.

Input1: 中国近年来还从国外引进了预分解窑生产线
SpIn: [中国] [近年来] [还从] [国外] [引进] [了] [预分解窑] [生产线]
LTP: [中国] [近年来] [还从] [国外] [引进] [了] [预分] [解] [窑] [生产线]
Input2: 只要有百分之一的漏失,就可能全盘皆输
SpIn: [只要] [有] [百分之一] [的] [漏失] [,] [就] [可能] [全盘皆输]
LTP: [只要] [有] [百分之一] [的] [漏失] [,] [就] [可能] [全盘] [皆] [输]

Figure 6: SpIn v.s. LTP4.0

5.1 Datasets & Settings

The widely used dataset Balanced Corpus of Contemporary Written Japanese (BCCWJ) version 1.1 (Maekawa et al., 2014) is evaluated in JWS. We follow the same dataset split with the Project Next NLP for BCCWJ. UD_Korean-GSD corpora [3] and KAIST [4] are used to evaluate KWS. These two widely used datasets in syntactic parsing tasks are automatically converted from structural trees in the Google UD Treebank (McDonald et al., 2013) and the KAIST Treebank (Choi et al., 1994). BERT-base-Chinese is substituted with BERT_Multilingual that contains Japanese and Korean as the feature extraction layer.

5.2 Results of JWS and KWS

As LSTM (Long Short Term Memory) neural network is employed in (Kitagawa and Komachi, 2018), we exclude performance boosts gained from BERT and conduct the contrast experiment between the traditional methods and the SpIn. We employ unigram and traditional tagging schemas in the comparative experiments. Table 10 demonstrates that SpIn also achieves SOTA results on JWS. In contrast with works leveraging word dictionaries and character type information, SpIn is closed without any extra knowledge. Besides, compared with the traditional methods that also leverage BERT, significant improvement up to **+0.55%** F1 score is obtained. Meanwhile, the best R_oov is observed. As no WS work was conducted on these two Korean datasets, we report results compared with traditional methods in Table 12. Performance boosts are observed on both datasets especially up to **+1.25%** F1 improvement on the GSD dataset.

[3] https://github.com/emorynlp/ud-korean/tree/master/google
[4] https://github.com/UniversalDependencies/UD_Korean-Kaist

R_oov boosts indicate SpIn is with good generalization ability and works effectively for Korean.

6 Conclusion

SpIn provides a novel viewpoint and implements the WS task by modeling two consecutive characters' separation states. Our simple but effective framework is robust and universal. State-of-the-art performances of word segmentation tasks are achieved in East Asian languages. Moreover, the significant boosts on OOV words demonstrate that SpIn has the robustness and generalization ability.

References

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.

Wanxiang Che, Yunlong Feng, Libo Qin, and Ting Liu. 2021. N-LTP: An open-source neural language technology platform for Chinese. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 42–49, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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.

Key-Sun Choi, Young S Han, Young G Han, and Oh W Kwon. 1994. Kaist tree bank project for korean: Present and future development. In *Proceedings of the International Workshop on Sharable Natural Language Resources*, pages 7–14. Citeseer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Thomas Emerson. 2005. The second international Chinese word segmentation bakeoff. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Jingjing Gong, Xinchi Chen, Tao Gui, and Xipeng Qiu. 2019. Switch-lstms for multi-criteria chinese word segmentation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 6457–6464. AAAI Press.

Shohei Higashiyama, Masao Utiyama, Eiichiro Sumita, Masao Ideuchi, Yoshiaki Oida, Yohei Sakamoto,

and Isaac Okada. 2019. Incorporating word attention into character-based 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 2699–2709, Minneapolis, Minnesota. Association for Computational Linguistics.

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 28th International Conference on Computational Linguistics*, pages 2062–2072, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Zhen Ke, Liang Shi, Songtao Sun, Erli Meng, Bin Wang, and Xipeng Qiu. 2021. Pre-training with meta learning for chinese word segmentation. pages 5514–5523.

Yoshiaki Kitagawa and Mamoru Komachi. 2018. Long short-term memory for Japanese word segmentation. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Changki Lee and Hyunki Kim. 2013. Automatic korean word spacing using pegasos algorithm. *Inf. Process. Manage.*, 49(1):370–379.

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

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

Jin Kiat Low, Hwee Tou Ng, and Wenyuan Guo. 2005. A maximum entropy approach to Chinese word segmentation. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

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.

Kikuo Maekawa, Makoto Yamazaki, Toshinobu Ogiso, Takehiko Maruyama, Hideki Ogura, Wakako Kashino, Hanae Koiso, Masaya Yamaguchi, Makiro Tanaka, and Yasuharu Den. 2014. Balanced corpus of contemporary written japanese. language resources and evaluation. *Language Resources and Evaluation*, 48.

Ryan McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith Hall, Slav Petrov, Hao Zhang, Oscar Täckström, Claudia Bedini, Núria Bertomeu Castelló, and Jungmee Lee. 2013. Universal Dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria. 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 H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 2746–2757. Curran Associates, Inc.

Hajime Morita, Daisuke Kawahara, and Sadao Kurohashi. 2015. Morphological analysis for unsegmented languages using recurrent neural network language model. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 2292–2297, Lisbon, Portugal. Association for Computational Linguistics.

Wenzhe Pei, Tao Ge, and Baobao Chang. 2014. Max-margin tensor neural network for Chinese word segmentation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 293–303, Baltimore, Maryland. Association for Computational Linguistics.

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

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. 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 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Xipeng Qiu, Hengzhi Pei, Hang Yan, and Xuanjing Huang. 2020. A concise model for multi-criteria Chinese word segmentation with transformer encoder. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2887–2897, Online. Association for Computational Linguistics.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Richard W. Sproat, Chilin Shih, William Gale, and Nancy Chang. 1996. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics*, 22(3):377–404.

Weiwei Sun and Jia Xu. 2011. Enhancing Chinese word segmentation using unlabeled data. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 970–979, Edinburgh, Scotland, UK. Association for Computational 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*, pages 8274–8285, Online. Association for Computational Linguistics.

Huihsin Tseng, Pichuan Chang, Galen Andrew, Daniel Jurafsky, and Christopher Manning. 2005. A conditional random field word segmenter for sighan bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*.

Yiou Wang, Jun'ichi Kazama, Yoshimasa Tsuruoka, Wenliang Chen, Yujie Zhang, and Kentaro Torisawa. 2011. Improving Chinese word segmentation and POS tagging with semi-supervised methods using large auto-analyzed data. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 309–317, Chiang Mai, Thailand. Asian Federation of 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. *Nat. Lang. Eng.*, 11(2):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.

Jie Yang, Yue Zhang, and Fei Dong. 2017. Neural word segmentation with rich pretraining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 839–849, Vancouver, Canada. Association for Computational Linguistics.

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.

Longkai Zhang, Houfeng Wang, Xu Sun, and Mairgup Mansur. 2013. Exploring representations from unlabeled data with co-training for Chinese word segmentation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 311–321, Seattle, Washington, USA. Association for Computational Linguistics.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Transition-based neural word segmentation. pages 421–431.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2018. Transition-based neural word segmentation using word-level features. *J. Artif. Int. Res.*, 63(1):923–953.

Yue Zhang and Stephen Clark. 2007. Chinese segmentation with a word-based perceptron algorithm. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 840–847, Prague, Czech Republic. Association for Computational Linguistics.

Hai Zhao, Chang-Ning Huang, and Mu Li. 2006. An improved Chinese word segmentation system with conditional random field. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 162–165, Sydney, Australia. Association for Computational Linguistics.