Joint Chinese Word Segmentation and Part-of-speech Tagging via Two-stage Span Labeling

Duc-Vu Nguyen[1,3], Linh-Bao Vo[2,3], Ngoc-Linh Tran[2,3]
Kiet Van Nguyen[2,3], Ngan Luu-Thuy Nguyen[2,3]
[1]Multimedia Communications Laboratory, University of Information Technology, Ho Chi Minh City, Vietnam
[2]University of Information Technology, Ho Chi Minh City, Vietnam
[3]Vietnam National University, Ho Chi Minh City, Vietnam
vund@uit.edu.vn, {18520503,20521538}@gm.uit.edu.vn
{kietnv,ngannlt}@uit.edu.vn

Abstract

Chinese word segmentation and part-of-speech tagging are necessary tasks in terms of computational linguistics and application of natural language processing. Many researchers still debate the demand for Chinese word segmentation and part-of-speech tagging in the deep learning era. Nevertheless, resolving ambiguities and detecting unknown words are challenging problems in this field. Previous studies on joint Chinese word segmentation and part-of-speech tagging mainly follow the character-based tagging model focusing on modeling n-gram features. Unlike previous works, we propose a neural model named SPANSEGTAG for joint Chinese word segmentation and part-of-speech tagging following the span labeling in which the probability of each n-gram being the word and the part-of-speech tag is the main problem. We use the biaffine operation over the left and right boundary representations of consecutive characters to model the n-grams. Our experiments show that our BERT-based model SPANSEGTAG achieved competitive performances on the CTB5, CTB6, and UD, or significant improvements on CTB7 and CTB9 benchmark datasets compared with the current state-of-the-art method using BERT or ZEN encoders.

1 Introduction

Chinese word segmentation (CWS) and part-of-speech (POS) tagging are necessary tasks in terms of computational linguistics and application of natural language processing (NLP). There are two primary approaches for joint CWS and POS tagging, including the two-step and one-step methods. The two-step approach is to find words and then assign POS tags to found words. Ng and Low (2004) proposed the one-step approach that combines CWS and POS tagging into a unified joint task. The one-step approach was proved better than two-step approach by many prior studies (Jiang et al., 2008; Jiang et al., 2009; Sun, 2011; Zeng et al., 2013; Zheng et al., 2013; Kurita et al., 2017; Shao et al., 2017; Zhang et al., 2018). These studies proposed various methods incorporating linguistic features or contextual information into their joint model. Remarkably, Tian et al. (2020a) proposed a two-way attention mechanism incorporating both context features and corresponding syntactic knowledge from off-the-shelf toolkits for each input character.

To our best knowledge, we observed all previous studies for joint CWS and POS tagging following the character-based tagging paradigm. The character-based tagging effectively produces the best combination of word boundary and POS tag. However, this character-based tagging paradigm does not give us a clear explanation when processing overlapping ambiguous strings. From the view of experimental psychology, human perception and performance, Ma et al. (2014) concluded that multiple words constituted by the characters in the perceptual span are activated when processing overlapping ambiguous strings. Besides, Tian et al. (2020b) shown that modeling word-hood for n-gram information is essential for CWS. Next, the current state-of-the-art method for joint CWS and POS tagging also confirmed the importance of modeling words and their knowledge, e.g., POS tag (Tian et al., 2020a).

The previous studies in two views of experimental psychology, human perception and performance, (Ma et al., 2014) and computational linguistics (Tian et al., 2020b; Tian et al., 2020a) inspired us to

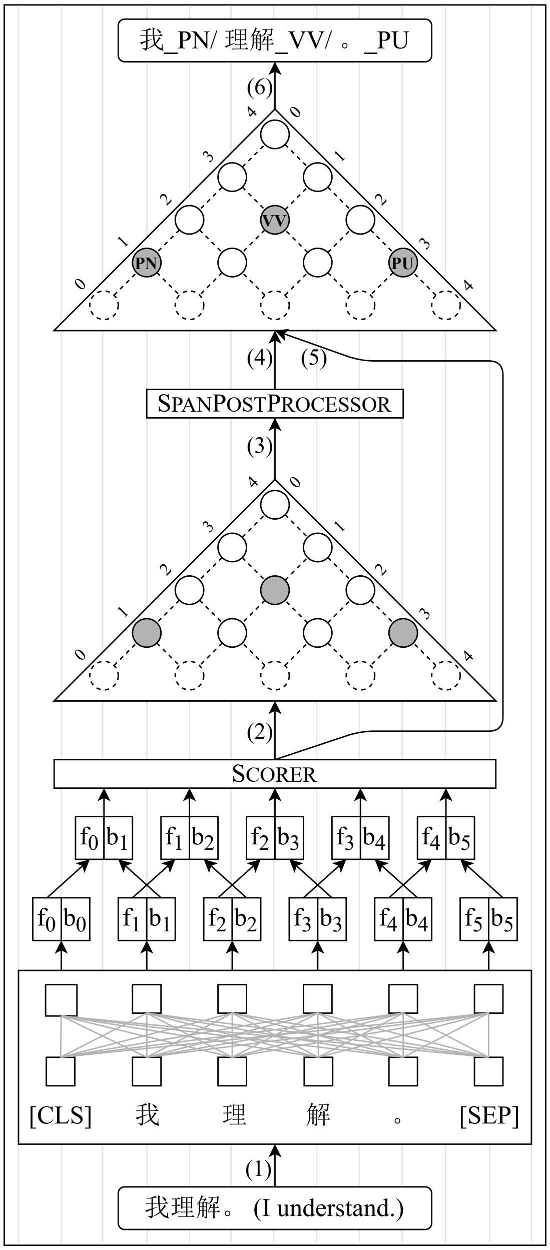

Figure 1: The architecture of SPANSEGTAG for the joint CWS and POS tagging with two stages via span labeling: word segmentation and POS tagging.

propose the span labeling approach for joint CWS and POS tagging. To avoid the model size dependent on numbers of n-grams and their corresponding POS tag, we use span to model n-gram and n-gram with POS tag instead of using the memory networks in (Tian et al., 2020b; Tian et al., 2020a). More particularly, inspired by Stern et al. (2017), Zhang et al. (2020), and (Nguyen et al., 2021), we use the biaffine operation over the left and right boundary representations of consecutive characters to model n-grams and their POS tag. As the prior work of Nguyen et al. (2021), we use a simple post-processing heuristic algorithm instead of using other models to deal with the overlapping ambiguity phenomenon (Li et al., 2003; Gao et al., 2005). Finally, we experimented with BiLSTM (Hochreiter and Schmidhuber, 1997) and BERT encoders (Devlin et al., 2019).

Our experiments show that our BERT-based model SPANSEGTAG achieved competitive performances on the CTB5, CTB6, UD1, and UD2, and significant improvements on the two large benchmark datasets CTB7 and CTB9 compared with the current state-of-the-art method using BERT or ZEN encoders (Tian et al., 2020a). Our SPANSEGTAG did not perfectly perform in five Chinese benchmark datasets. However, SPANSEGTAG achieved a good recall of in-vocabulary words and their POS tag scores on CTB6, CTB7 and CTB9 datasets. This score is used to measure the performance of the segmenter in resolving ambiguities in word segmentation (Gao et al., 2005).

2 The Proposed Framework

We present the architecture of our proposed framework, namely SPANSEGTAG, for joint CWS and POS tagging in Figure 1. As we can see in Figure 1, data path (1) indicates the input sentence to be fed into the BERT encoder. The hidden state vector from the BERT encoder is chunk into two vectors with the same size as the forward and backward vectors in the familiar encoder, BiLSTM. Next, all boundary representations are fed into the SCORER module. Data path (2) indicates the span representations for the word segmentation task, and data path (5) indicates the span representations for the POS tagging task. Data path (3) indicates predicted spans representing predicted word boundaries. The SPANPOSTPROCESSOR module produces the predicted spans satisfying non-overlapping between every two spans. Finally, given data paths (4) and (5), the data path (6) indicates the joint CWS and POS tagging.

2.1 Joint Chinese Word Segmentation and Part-of-speech Tagging as Two Stages Span Labeling

The input sentence of joint CWS and POS tagging is a sequence of characters $\mathcal{X} = x_1 x_2 \dots x_n$ with the length of n. Given the input sentence $\mathcal{X}$, the output of CWS is a sequence of words $\mathcal{W} = w_1 w_2 \dots w_m$ with the length of m, and the output of Chinese POS tagging is a sequence of POS tags $\mathcal{T} = t_1 t_2 \dots t_m$ with the length of m, where $1 \leq m \leq n$. Besides, we have a property that the Chinese word w_j is constituted by one Chinese character or consecutive characters. Therefore, we use the sequence of characters $x_i x_{i+1} \dots x_{i+k-1}$ to denote that the word w_j is constituted by k consecutive characters beginning at character x_i, where $1 \leq k \leq n$ and $k = 1$ representing single words and $2 \leq k \leq n$ representing compound words. We get the inspiration of span representation in constituency parsing (Stern et al., 2017) to use the span $(i-1, i-1+k)$ representing the word constituted by k consecutive characters $x_i x_{i+1} \dots x_{i+k-1}$ beginning at character x_i, where $i-1$ and $i-1+k$ are the left and the right boundary index of word $x_i x_{i+1} \dots x_{i+k-1}$, respectively.

After presenting notations, we propose our approach for the joint CWS and POS tagging problem. Firstly, to our knowledge, most recent works focus on modeling the probability that a Chinese character can be one in the combination of {B, I, E, S} and Chinese POS tags set. Next, the current state-of-the-art method for CWS approaching BIES tagging of Tian et al. (2020b) proposed word-hood memory to model n-gram information. Additionally, the current state-of-the-art method for joint CWS and POS tagging approaching BIES tagging of Tian et al. (2020a) shown that modeling n-gram knowledge, e.g., word and POS tag, is essential. Therefore, we get inspiration of Tian et al. (2020b) and Tian et al. (2020a) to focus on modeling words and POS tags in a straightforward way rather than modeling BIES tags of characters. Given the input sentence $\mathcal{X}$, our idea is to model the probability that the consecutive Chinese characters can be a word via one formulation. Similarly, given the input sentence $\mathcal{X}$, we also model the probability that consecutive Chinese characters can be assigned a specific POS tag or the non-word tag via one formulation. To summarize, given span representations, we formalize the joint CWS and POS tagging task as two continuous sub-tasks in our SPANSEGTAG as following: (i) binary classification dealing with word segmentation; (ii) multi-class classification dealing with POS tagging.

Formally, the first stage of our SPANSEGTAG for CWS can be formalized as:

$$\hat{\mathcal{S}}_{\text{novlp}} = \text{SpanPostProcessor}(\hat{\mathcal{S}}) \tag{1}$$

where SPANPOSTPROCESSOR($\hat{\mathcal{S}}$) is introduced in the work of Nguyen et al. (2021). SPANPOSTPROCESSOR($\hat{\mathcal{S}}$) solely is an algorithm for producing the word segmentation boundary guaranteeing non-overlapping between every two spans. The $\hat{\mathcal{S}}$ is the set of predicted spans as follows:

$$\hat{\mathcal{S}} = \Big\{(l, r) \text{ for } 0 \leq l \leq n-1 \text{ and } l < r \leq n \text{ and } \text{Scorer}(\mathcal{X}, l, r).\text{Seg} > 0.5\Big\} \tag{2}$$

where n is the length of the input sentence. The l and r denote left and right boundary indexes of the specific span. The SCORER($\mathcal{X}, l, r$).SEG is the scoring module for the span (l, r) of sentence $\mathcal{X}$. The output of SCORER($\mathcal{X}, l, r$).SEG has a value in the range of 0 to 1. We choose the sigmoid function as the activation function at the last layer of SCORER($\mathcal{X}, l, r$).SEG module.

Next, given the set of predicted spans $\hat{\mathcal{S}}_{\text{novlp}}$ satisfying non-overlapping between every two spans for the input sentence $\mathcal{X}$, the second stage of our SPANSEGTAG to perform Chinese POS tagging can be formalized as:

$$\hat{\mathcal{Y}} = \Big\{\Big((l, r), \operatorname*{argmax}_{\hat{t} \in \mathcal{T}} \text{Scorer}(\mathcal{X}, l, r).\text{Tag}[\hat{t}]\Big) \text{ for } (l, r) \in \hat{\mathcal{S}}_{\text{novlp}}\Big\} \tag{3}$$

where $\mathcal{T}$ is the union of Chinese POS tag set and the non-word tag since the $\hat{\mathcal{S}}_{\text{novlp}}$ can include the incorrectly predicted span. The SCORER($\mathcal{X}, l, r$).TAG$[\hat{t}]$ is the scoring module for the span (l, r) of sentence $\mathcal{X}$ assigned tag $\hat{t}$. To sum up, given the input sentence $\mathcal{X}$, the set $\hat{\mathcal{Y}}$ includes predicted spans with the

POS tag. Therefore, the set $\hat{\mathcal{Y}}$ is the result of the second stage of our SPANSEGTAG and of the joint CWS and POS tagging task.

The main idea of our SPANSEGTAG is formalized through three Equations 1, 2, and 3. To train our SPANSEGTAG, we have to optimize parameters in $\text{SCORER}(\mathcal{X}, l, r)$.SEG and $\text{SCORER}(\mathcal{X}, l, r).\text{TAG}[\hat{t}]$ modules. As we clearly see that there is no parameters in the SPANPOSTPROCESSOR($\hat{\mathcal{S}}$) module. However, the optimization of parameters in $\text{SCORER}(\mathcal{X}, l, r).\text{TAG}[\hat{t}]$ based on the $\hat{\mathcal{S}}_{\text{novlp}}$ indirectly optimizes parameters in our SPANSEGTAG by learning from the result of SPANPOSTPROCESSOR($\hat{\mathcal{S}}$). For example, if an incorrect span is assigned non-word tag, then our SPANSEGTAG is trained to deal with this case via $\text{SCORER}(\mathcal{X}, l, r).\text{TAG}[\hat{t}]$ module.

Therefore, the cost function for training our SPANSEGTAG is the combined loss of binary classification and multi-class classification. The cost function for training CWS in our SPANSEGTAG is

$$
\begin{aligned}
&J_{\text{SEG}}(\theta, \theta_{\text{SEG}}) = -\frac{1}{|\mathcal{D}|} \sum_{\mathcal{X}, \mathcal{S} \in \mathcal{D}} \Bigg(\\
&\frac{1}{(n(n+1))/2} \sum_{l=0}^{n-1} \sum_{r=l+1}^{n} \Bigg(\\
&[(l, r) \in \mathcal{S}] \log \Big(\text{SCORER}(\mathcal{X}, l, r).\text{SEG} \Big) \\
&+ [(l, r) \notin \mathcal{S}] \log \Big(1 - \text{SCORER}(\mathcal{X}, l, r).\text{SEG} \Big) \Bigg) \Bigg)
\end{aligned} \tag{4}
$$

where $\mathcal{D}$ is the training set and $|\mathcal{D}|$ is the size of the training set. For each pair $(\mathcal{X}, \mathcal{S})$ in training set $\mathcal{D}$, we compute binary cross-entropy loss for all spans (l, r), where $0 \leq l \leq n-1$ and $l < r \leq n$, and n is the length of sentence $\mathcal{X}$. The term $[(l, r) \in \mathcal{S}]$ has the value of 1 if span (l, r) belongs to the list $\mathcal{S}$ of sentence $\mathcal{X}$ and conversely, of 0. Similarly, the term $[(l, r) \notin \mathcal{S}]$ has the value of 1 if span (l, r) does not belong to the list $\mathcal{S}$ of sentence $\mathcal{X}$ and conversely, of 0. Notably, our training and prediction progress, we discard spans with length greater than 7 as the maximum n-gram length following (Diao et al., 2020) to reduce negative spans.

Next, the cost function for training Chinese POS tagging in our SPANSEGTAG is the cross entropy loss:

$$
\begin{aligned}
&J_{\text{TAG}}(\theta, \theta_{\text{TAG}}) = \frac{1}{|\mathcal{D}|} \sum_{\mathcal{X}, \mathcal{Y} \in \mathcal{D}} \Bigg(\frac{1}{|\hat{\mathcal{S}}_{\text{novlp}}|} \sum_{(l,r) \in \hat{\mathcal{S}}_{\text{novlp}}} \\
&\Bigg(- \text{SCORER}(\mathcal{X}, l, r).\text{TAG}[t] \\
&+ \log \Big(\sum_{\hat{t} \in \mathcal{T}} \exp \big(\text{SCORER}(\mathcal{X}, l, r).\text{TAG}[\hat{t}] \big) \Big) \Bigg) \Bigg)
\end{aligned} \tag{5}
$$

where t denotes the truth label of span (l, r) from $\mathcal{Y}$ in the input sentence $\mathcal{X}$. Finally, the cost function for training our SPANSEGTAG is

$$
J(\theta, \theta_{\text{SEG}}, \theta_{\text{TAG}}) = J_{\text{SEG}}(\theta, \theta_{\text{SEG}}) + J_{\text{TAG}}(\theta, \theta_{\text{TAG}}) \tag{6}
$$

2.2 Decoding Algorithm for Predicted Span

As the problem in prior work of Nguyen et al. (2021), in the predicted span set $\hat{\mathcal{S}}$ mentioned in Equation 2 there exists overlapping between some two spans. To solve this, Nguyen et al. (2021) keep the spans with the highest score and eliminate the remainder. The overlapping ambiguity phenomenon happens during our SPANSEGTAG predicting compound words. Additionally, our SPANSEGTAG encounters the missing word boundary problem. That problem can be caused by originally predicted spans, the consequence of solving overlapping ambiguity, or more than seven-character spans mentioned in subsection 2.1. Finally, we add the missing word boundary based on all predicted spans $(i-1, i-1+k)$ with $k = 1$ to single words to deal with the missing word boundary problem following Nguyen et al. (2021). The detail of this algorithm is shown in the work of Nguyen et al. (2021). To sum up, SPANPOSTPROCESSOR($\hat{\mathcal{S}}$) is considered as the heuristic algorithm, while the inference algorithm in (Ye and Ling, 2018) is optimal.

2.3 Span Scoring

Inspired by Zhang et al. (2020), the span scoring module $\text{SCORER}(\mathcal{X}, l, r)$.SEG for finding probability of word is computed by using a biaffine operation over the left boundary representation of character x_l and the right boundary representation of character

x_r:

$$\textsc{Scorer}(\mathcal{X}, l, r).\textsc{Seg} = \text{sigmoid}\Bigg(\begin{bmatrix} \text{MLP}^{\text{left}}_{\text{seg}}(\mathbf{f}_l \oplus \mathbf{b}_{l+1}) \\ 1 \end{bmatrix}^{\mathrm{T}} \mathbf{W}\big(\text{MLP}^{\text{right}}_{\text{seg}}(\mathbf{f}_r \oplus \mathbf{b}_{r+1})\big)\Bigg) \tag{7}$$

where $\mathbf{W} \in \mathbb{R}^{(d+1)\times d}$ and the symbol $\oplus$ denote the concatenation operation. Similarly, the span scoring module $\textsc{Scorer}(\mathcal{X}, l, r).\textsc{Tag}[\hat{t}]$ for finding score of a POS tag $\hat{t} \in \mathcal{T}$ is computed by:

$$\textsc{Scorer}(\mathcal{X}, l, r).\textsc{Tag}[\hat{t}] = \begin{bmatrix} \text{MLP}^{\text{left}}_{\text{tag}}(\mathbf{f}_l \oplus \mathbf{b}_{l+1}) \\ 1 \end{bmatrix}^{\mathrm{T}} \mathbf{W}_{\hat{t}} \begin{bmatrix} \text{MLP}^{\text{right}}_{\text{tag}}(\mathbf{f}_r \oplus \mathbf{b}_{r+1}) \\ 1 \end{bmatrix} \tag{8}$$

where $\mathbf{W}_{\hat{t}} \in \mathbb{R}^{(d+1)\times(d+1)}$. As mentioned in subsection 2.1, we have $0 \le l \le n-1$ and $l < r \le n$, where n is the length of input sentence $\mathcal{X}$. The $\text{MLP}^{\text{left}}_{\text{seg}}$, $\text{MLP}^{\text{right}}_{\text{seg}}$, $\text{MLP}^{\text{left}}_{\text{tag}}$ and $\text{MLP}^{\text{right}}_{\text{tag}}$ are multilayer perceptrons for transforming hidden states from encoder to left and boundary representations with the output dimension of d for CSW and POS tagging tasks. Vectors $\mathbf{f}_i$ and $\mathbf{b}_i$ denote forward and backward hidden state vectors from BiLSTM encoder. In case we use BERT encoder, we chunk the hidden state vector from BERT encoder into two vectors with the same size as the forward and backward hidden state vectors in the BiLSTM encoder.

2.4 Encoder Architecture

To experiment with our proposed SpanSegTag, we use BiLSTM encoder (Hochreiter and Schmidhuber, 1997) and $\text{BERT}_{\text{BASE}}$ encoder for Chinese (Devlin et al., 2019). In case we use LSTM encoder, we use character pre-trained Chinese embedding with the dimension of 64 provided Shao et al. (2017). In case we use BERT encoder, we use only the hidden state of the last layer of BERT as Tian et al. (2020a).

Datasets		# Sent	# Char	# Word	OOV
CTB5	Train	18,104	804,587	493,930	-
	Dev	352	11,543	6,821	8.1
	Test	348	13,738	8,008	3.5
CTB6	Train	23,420	1,055,583	641,368	-
	Dev	2,079	100,316	59,955	5.4
	Test	2,796	134,149	81,578	5.6
CTB7	Train	31,112	1,160,209	717,874	-
	Dev	10,043	387,209	236,590	5.5
	Test	10,292	398,626	245,011	5.2
CTB9	Train	105,971	2,642,998	1,696,340	-
	Dev	9,850	209,739	136,468	2.9
	Test	15,929	378,502	242,317	3.1
UD	Train	3,997	156,309	98,608	-
	Dev	500	20,000	12,663	12.1
	Test	500	19,206	12,012	12.4

Table 1: Statistics of five Chinese benchmark datasets. We provide the number of sentences, characters, and words. We also compute the out-of-vocabulary (OOV) rate as the percentage of unseen words in the dev and test set.

3 Experiments

3.1 Datasets

We employ the CTB5, CTB6, CTB5, and CTB9[1] benchmark datasets from the Penn Chinese Treebank (Xue et al., 2005), which has been widely used in research on joint CWS and POS tagging. There are 33 POS tags in CTB. The train/dev/test split for CTB5, CTB6, CTB7 and CTB9 is according to previous studies (Zhang et al., 2014; Yang and Xue, 2012; Wang et al., 2011; Shao et al., 2017). We also employ UD1 and UD2 to denote the datasets using universal tag set and Chinese tag set from UD (Nivre et al., 2016)[2] following the research of Tian et al. (2020a), respectively.

3.2 Implementation

The number of layers of BiLSTM is 1, and the hidden state size of BiLSTM is 200. The dropout rate for embedding, BiLSTM, and MLPs is 0.1. We in-

[1]We officially employ the Penn Chinese TreeBank data (LDC2016T13) from the Linguistic Data Consortium.

[2]We use the UD_Chinese-GSD dataset with the version 2.4, which extracted from `https://universaldependencies.org/`.

SpanSegTag		CTB5		CTB6		CTB7		CTB9		UD1		UD2	
Encoder	MLP Size	Seg	Tag	Seg	Tag	Seg	Tag	Seg	Tag	Seg	Tag	Seg	Tag
BiLSTM	100	96.71	92.80	94.33	89.43	94.46	89.17	95.64	91.27	91.84	85.21	91.48	84.80
	200	96.90	93.08	94.90	90.06	94.70	89.36	95.96	91.57	92.36	85.92	92.27	85.78
	300	97.03	93.21	95.00	90.06	94.86	89.39	96.05	91.61	92.43	86.14	92.72	85.93
	400	96.82	93.27	95.18	90.16	95.04	89.53	96.15	91.54	93.02	86.45	92.84	86.03
	500	97.30	**93.39**	95.29	**90.19**	95.10	**89.53**	96.27	**91.61**	93.08	**86.74**	93.12	**86.29**
BERT	100	98.76	97.78	97.71	95.25	97.06	94.16	97.75	94.92	98.21	95.51	98.22	95.38
	200	98.78	97.71	97.66	95.25	97.11	94.24	97.78	95.07	98.23	95.64	98.21	**95.50**
	300	98.56	97.54	97.70	95.24	97.12	**94.27**	97.74	95.02	98.35	**95.72**	98.22	95.49
	400	98.57	97.64	97.69	**95.26**	97.05	94.18	97.80	**95.10**	98.28	95.70	98.17	95.44
	500	98.81	**97.78**	97.69	95.23	97.10	94.22	97.80	95.01	98.30	95.66	98.30	95.44

Table 2: Experimental results on development sets of six Chinese benchmark datasets.

herit hyper-parameters from the work of (Dozat and Manning, 2017). We trained all models up to 100 with the early stopping strategy with patience epochs of 20. We used AdamW optimizer (Ilya Loshchilov and Frank Hutter, 2019) with the default configuration and learning rate of 10^{-3}. The batch size for training and evaluating is up to 5000.

We did fine-tuning experiments based on BERT (Devlin et al., 2019). We trained all models up to 100 with the early stopping strategy with patience epochs of 15 following Tian et al. (2020a). The dropout rate for MLPs is 0.1. We used AdamW optimizer (Ilya Loshchilov and Frank Hutter, 2019) with the default configuration and learning rate of 10^{-5}. The batch size for training is 16.

All models were selected based on the performance of the development set. The measure we use for the main result is F-score following previous research. To evaluate F-score of joint CWS and POS tagging, we use the library[3] following the research of Tian et al. (2020a). We also use paired t-test following the guide of the research (Dror et al., 2018) to test the significance of our research.

3.3 Development Performance

In Table 2, we show the performance of SpanSegTag with the output size of MLPs mentioned in subsection 2.3. Concerning the BiLSTM encoder, the larger MLP size gives the higher performance in all datasets. Because we regard the joint CWS and POS tagging as a span labeling task, it requires more contextual information. In view of dependency parsing, Dozat and Manning (2017) chose the MLP size to be 500 for unlabeled parsing. Regarding the BERT encoder, the results of different MLP sizes are not clearly distinguished as those of the BiLSTM encoder since the BERT encoder provides better contextual information.

3.4 Overall Performance

We run the final testing experiment with the BERT encoder on six datasets compared to previous results, as shown in Table 3. Firstly, we can see our SpanSegTag achieve competitive results on CTB5, UD1, and UD2 compared with research of Tian et al. (2020a) using BERT encoder. Our SpanSegTag achieved the competitive or higher F-score on joint CWS and POS tagging even we get the lower CWS performance on CTB5, UD1, and UD2. Besides, our SpanSegTag obtained the higher F-scores of joint CWS and POS tagging on CTB6, CTB7, and CTB9 compared with (Tian et al., 2020a).

Compared with Tian et al. (2020a) using ZEN (Diao et al., 2020) encoder, we note that the ZEN encoder, which enhances the n-gram information, was better than the BERT encoder on many Chinese NLP tasks (Diao et al., 2020). Though, our SpanSegTag with BERT also obtained the higher joint CWS and POS tagging performance on CTB6, CTB7, CTB9,

[3] https://github.com/chakki-works/seqeval.

[5] We downloaded all pre-trained models of Tian et al. (2020a) from their publicly resource https://github.com/SVAIGBA/TwASP. However, we can not reproduce the result on the UD2 dataset.

	CTB5		CTB6		CTB7		CTB9		UD1		UD2	
	Seg	Tag	Seg	Tag	Seg	Tag	Seg	Tag	Seg	Tag	Seg	Tag
Jiang et al. (2008)	97.85	93.41	-	-	-	-	-	-	-	-	-	-
Kruengkrai et al. (2009)	97.87	93.67	-	-	-	-	-	-	-	-	-	-
Sun (2011)	98.17	94.02	-	-	-	-	-	-	-	-	-	-
Wang et al. (2011)	98.11	94.18	95.79	91.12	95.65	90.46	-	-	-	-	-	-
Shen et al. (2014)	98.03	93.80	-	-	-	-	-	-	-	-	-	-
Kurita et al. (2017)	98.41	94.84	-	-	96.23	91.25	-	-	-	-	-	-
Shao et al. (2017)	98.02	94.38	-	-	-	-	96.67	92.34	95.16	89.75	95.09	89.42
Zhang et al. (2018)	98.50	94.95	96.36	92.51	96.25	91.87	-	-	-	-	-	-
Tian et al. (2020a) (BERT)	98.77	96.77	97.39	94.99	**97.32**	94.28	97.75	94.87	98.32	95.60	98.33	95.46
Tian et al. (2020a) (ZEN)	**98.81**	**96.92**	97.47	95.02	97.31	94.32	97.77	94.88	**98.33**	**95.69**	**98.18**	95.49
SPANSEGTAG (BERT)	98.67	96.77	**97.53**	**95.04**	97.30	**94.50**‡	**97.86**	**95.22**‡	98.06	95.59	98.12	**95.54**

Table 3: Experimental results on test sets of six Chinese benchmark datasets. The symbol ‡ denotes that the improvement is statistically significant at p < 0.01 compared with TwASP[5](ZEN) (Tian et al., 2020a) using paired t-test.

and UD1. Moreover, our improvements on CTB7 and CTB9 is statistically significant at $p < 0.01$ using paired t-test. We can explain our improvements by modeling all n-grams in the input sentence directly to the word segmentation and POS tagging task via span labeling rather than indirectly according to the work of Tian et al. (2020a). To sum up, our SPANSEGTAG does not achieve state-of-the-art performance on all six datasets. However, we obtained significant results on two of the largest joint CWS and POS tagging datasets, including CTB7 and CTB9. To explore the pros and cons of our SPANSEGTAG, we provide analysis on the section 4.

	$R_{POS\text{-}OOV}$			$R_{POS\text{-}iV}$		
	TwASP (BERT)	TwASP (ZEN)	Our (BERT)	TwASP (BERT)	TwASP (ZEN)	Our (BERT)
CTB5	**83.81**	**83.81**	82.73	97.54	**97.55**	97.54
CTB6	83.10	**84.22**	82.69	95.48	95.66	**95.68**
CTB7	79.94	79.39	**80.19**	95.20	95.25	**95.33**
CTB9	**79.93**	78.80	78.52	95.49	95.44	**95.80**
UD1	**88.67**	87.40	86.13	96.64	**96.92**	96.85

Table 4: Recall of out-of-vocabulary words and their POS tags ($R_{POS\text{-}OOV}$) and recall of in-vocabulary words and their POS tags ($R_{POS\text{-}iV}$). Notably, we do not provide scores on UD2 dataset since we can not reproduce result from the pre-trained model of Tian et al. (2020a).

4 Analysis

4.1 Recall of Out-of-vocabulary and in-vocabulary Words

Inspired by the research of Gao et al. (2005), we test the performance of detecting unknown words with POS tags ($R_{POS\text{-}OOV}$) and the performance of resolving ambiguities in word segmentation with POS tags ($R_{POS\text{-}iv}$), as shown in Table 4. The analysis reveals that our SPANSEGTAG tends to have the higher $R_{POS\text{-}iv}$ than $R_{POS\text{-}OOV}$. This analysis motivates us to research the multi-view model of sequence tagging and span labeling in future work.

4.2 Combination Ambiguity String Error

In addition to $R_{POS\text{-}iv}$ in subsection 4.1, we also follow (Gao et al., 2005) to analyze combination ambiguity string (CAS) errors, as shown in Table 5. The CAS detection requires a judgment of the syntactic and semantic sense of the segmentation. Hence, we only use the CAS measure in a pilot study. Inspired by (Gao et al., 2005), we test on a set of 70 high-frequency CASs of each dataset. The result tells that our SPANSEGTAG solves CASs slightly better than TwASP (Tian et al., 2020a) on the CTB6, CTB7 and CTB9 datasets. Hence, this error analysis will motivate the research community to improve the joint CWS and POS tagging task.

4.3 Model Size and Inference Speed

In theory, our SPANSEGTAG is a $O(n^2)$ algorithm due to computing of all possible span representations, which is equivalent to computing of mem-

	CTB5	CTB6	CTB7	CTB9	UD1
TwASP (BERT)	**96.43**	93.72	94.26	94.61	96.40
TwASP (ZEN)	**96.43**	94.88	94.23	95.47	**97.30**
Our (BERT)	95.71	**95.30**	**94.72**	**95.56**	**97.30**

Table 5: CWS accuracies of TwASP (Tian et al., 2020a) using BERT and ZEN versus our SPANSEGTAG on 70 high-frequency two-character CASs.

	CTB5	CTB6	CTB7	CTB9	UD1
TwASP (BERT)	514	699	716	650	435
TwASP (ZEN)	989	1,010	1,170	1,100	909
Our (BERT)	**433**	**434**	**435**	**441**	**413**

Table 6: Model sizes (MB) of TwASP (Tian et al., 2020a) using BERT and ZEN versus our SPANSEGTAG.

ory network for context features and corresponding knowledge instances from off-the-shelf toolkits in (Tian et al., 2020a). In practice, when use GPU Tesla V100 via Google Colaboratory, the inference speed of our SPANSEGTAG (BERT) and TwASP (BERT) are 264 and 239 (sentence/second), respectively. We notice that we did not count the time TwASP (Tian et al., 2020a) consuming by running off-the-shelf toolkits. Table 6 shows that the parameters of our SPANSEGTAG are independent of the datasets and significant smaller compared with TwASP (Tian et al., 2020a).

5 Related Work

The one-step approach for joint CWS and POS tagging was proved better than the two-step one by many prior studies (Jiang et al., 2008; Jiang et al., 2009; Sun, 2011; Zeng et al., 2013; Zheng et al., 2013; Kurita et al., 2017; Shao et al., 2017; Zhang et al., 2018). Besides, Tian et al. (2020a) confirmed the importance of context features and corresponding knowledge instances from off-the-shelf toolkits. Our work is related to (Chen et al., 2016) in view of using matrix for CWS and to (Sun and T'sou, 1995; Chen and Goodman, 1996; Li et al., 2003; Gao et al., 2005; Ma et al., 2014; Chen et al., 2016) concerning dealing with ambiguity for CWS.

6 Conclusion

In this paper, we propose a neural approach for joint CWS and POS tagging via span labeling. Our proposed approach uses the biaffine operation over the left and right boundary representations of consecutive characters to model the n-grams. Our experiments show that our BERT-based model SPANSEGTAG achieved competitive performances on the CTB5, CTB6, and UD, and significant improvements on the CTB7 and CTB9 benchmark datasets compared with the current state-of-the-art method TwASP using BERT and ZEN encoders. Our approach does not use any context features and corresponding knowledge instances from off-the-shelf toolkits and a significantly smaller model than TwASP. However, our SPANSEGTAG has the disadvantage of the complexity and time running. For future work, we will explore the architecture of the BERT model (Devlin et al., 2019) for joint CWS and POS tagging because the primitive of BERT also has the complexity of $O(n^2)$ and the self-attention mechanism over the input sentence may be related to span representation.

References

Stanley F. Chen and Joshua Goodman. 1996. An Empirical Study of Smoothing Techniques for Language Modeling. In *Proceedings of ACL*, pages 310–318.

Yanping Chen, Qinghua Zheng, Feng Tian, and Deli Zheng. 2016. A Segmentation Matrix Method for Chinese Segmentation Ambiguity Analysis. In *International Journal of Computational Linguistics & Chinese Language Processing, Volume 21, Number 1, June 2016*.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*, pages 4171–4186.

Shizhe Diao, Jiaxin Bai, Yan Song, Tong Zhang, and Yonggang Wang. 2020. ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations. In *Findings of EMNLP*, pages 4729–4740.

Timothy Dozat and Christopher D. Manning. 2017. Deep Biaffine Attention for Neural Dependency Parsing. In *Proceedings of ICLR*.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The Hitchhiker's Guide to Testing Statistical Significance in Natural Language Processing. In *Proceedings of ACL*, pages 1383–1392. Association for Computational Linguistics.

Jianfeng Gao, Mu Li, Andi Wu, and Chang-Ning Huang. 2005. Chinese Word Segmentation and Named Entity

Recognition: A Pragmatic Approach. *Computational Linguistics*, 31(4):531–574.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *Proceedings of ICLR*.

Wenbin Jiang, Liang Huang, Qun Liu, and Yajuan Lü. 2008. A Cascaded Linear Model for Joint Chinese Word Segmentation and Part-of-Speech Tagging. In *Proceedings of ACL*, pages 897–904.

Wenbin Jiang, Liang Huang, and Qun Liu. 2009. Automatic Adaptation of Annotation Standards: Chinese Word Segmentation and POS Tagging – A Case Study. In *Proceedings of ACL-IJCNLP*, pages 522–530.

Canasai Kruengkrai, Kiyotaka Uchimoto, Jun'ichi Kazama, Yiou Wang, Kentaro Torisawa, and Hitoshi Isahara. 2009. An Error-Driven Word-Character Hybrid Model for Joint Chinese Word Segmentation and POS Tagging. In *Proceedings of ACL-IJCNLP*, pages 513–521.

Shuhei Kurita, Daisuke Kawahara, and Sadao Kurohashi. 2017. Neural Joint Model for Transition-based Chinese Syntactic Analysis. In *Proceedings of ACL*, pages 1204–1214.

Mu Li, Jianfeng Gao, Chang-Ning Huang, and Jianfeng Li. 2003. Unsupervised Training for Overlapping Ambiguity Resolution in Chinese Word Segmentation. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 1–7.

Guojie Ma, Xingshan Li, and Keith Rayner. 2014. Word segmentation of overlapping ambiguous strings during Chinese reading. *Journal of Experimental Psychology: Human Perception and Performance*, 40(3):1046–1059.

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese Part-of-Speech Tagging: One-at-a-Time or All-at-Once? Word-Based or Character-Based? In *Proceedings of EMNLP*, pages 277–284.

Duc-Vu Nguyen, Linh-Bao Vo, Dang Van Thin, and Ngan Luu-Thuy Nguyen. 2021. Span Labeling Approach for Vietnamese and Chinese Word Segmentation. In *PRICAI 2021: Trends in Artificial Intelligence*, pages 244–258.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajič, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal Dependencies v1: A Multilingual Treebank Collection. In *Proceedings of LREC*, pages 1659–1666. European Language Resources Association (ELRA).

Yan Shao, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2017. Character-based Joint Segmentation and POS Tagging for Chinese using Bidirectional RNN-CRF. In *Proceedings of IJCNLP*, pages 173–183. Asian Federation of Natural Language Processing.

Mo Shen, Hongxiao Liu, Daisuke Kawahara, and Sadao Kurohashi. 2014. Chinese Morphological Analysis with Character-level POS Tagging. In *Proceedings of ACL*, pages 253–258.

Mitchell Stern, Jacob Andreas, and Dan Klein. 2017. A Minimal Span-Based Neural Constituency Parser. In *Proceedings of ACL*, pages 818–827. Association for Computational Linguistics.

Maosong Sun and Benjamin K. T'sou. 1995. Ambiguity Resolution in Chinese Word Segmentation. In *Proceedings of PACLIC*, pages 121–126. City University of Hong Kong.

Weiwei Sun. 2011. A Stacked Sub-Word Model for Joint Chinese Word Segmentation and Part-of-Speech Tagging. In *Proceedings of ACL*, pages 1385–1394.

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 ACL*, 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 ACL*, pages 8274–8285. Association for Computational Linguistics.

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 IJCNLP*, pages 309–317. 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. *Natural Language Engineering*, 11(2):207–238.

Yaqin Yang and Nianwen Xue. 2012. Chinese Comma Disambiguation for Discourse Analysis. In *Proceedings of ACL*, pages 786–794.

Zhixiu Ye and Zhen-Hua Ling. 2018. Hybrid semi-Markov CRF for Neural Sequence Labeling. In *Proceedings of ACL*, pages 235–240.

Xiaodong Zeng, Derek F. Wong, Lidia S. Chao, and Isabel Trancoso. 2013. Graph-based Semi-Supervised Model for Joint Chinese Word Segmentation and Part-of-Speech Tagging. In *Proceedings of ACL*, pages 770–779.

Meishan Zhang, Yue Zhang, Wanxiang Che, and Ting Liu. 2014. Type-Supervised Domain Adaptation for Joint Segmentation and POS-Tagging. In *Proceedings of EACL*, pages 588–597.

Meishan Zhang, Nan Yu, and Guohong Fu. 2018. A Simple and Effective Neural Model for Joint Word Segmentation and POS Tagging. *IEEE/ACM Transactions on Audio, Speech, and Language Processing (TASLP)*, 26:1528–1538.

Yu Zhang, Houquan Zhou, and Zhenghua Li. 2020. Fast and Accurate Neural CRF Constituency Parsing. In *Proceedings of IJCAI*, pages 4046–4053. International Joint Conferences on Artificial Intelligence Organization.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep Learning for Chinese Word Segmentation and POS Tagging. In *Proceedings of EMNLP*, pages 647–657.