The 18th Pacific Rim International Conference on Artificial Intelligence

Span Labeling Approach for Vietnamese and Chinese Word Segmentation

Authors: Duc-Vu Nguyen, Linh-Bao Vo, Dang Van Thin
Ngan Luu-Thuy Nguyen
Affiliations: University of Information Technology
Viet Nam National University Ho Chi Minh City

November 12, 2021

Outline

Outline

Introduction

1. **Word segmentation:** dividing a string of written language into its com-ponent words.

Introduction

1. **Word segmentation:** dividing a string of written language into its com-ponent words.
2. The input of Vietnamese word segmentation (VWS) is the sequence of **syllables** delimited by space.

Introduction

1. **Word segmentation:** dividing a string of written language into its com-ponent words.
2. The input of Vietnamese word segmentation (VWS) is the sequence of **syllables** delimited by space.
3. The input of Chinese word segmentation (CWS) is the sequence of **characters** WITHOUT explicit delimiter.

Introduction

1. **Word segmentation:** dividing a string of written language into its com-ponent words.
2. The input of Vietnamese word segmentation (VWS) is the sequence of **syllables** delimited by space.
3. The input of Chinese word segmentation (CWS) is the sequence of **characters** WITHOUT explicit delimiter.
4. The use of a Vietnamese syllable and a Chinese character are similar.

Introduction

1. **Word segmentation:** dividing a string of written language into its com-ponent words.
2. The input of Vietnamese word segmentation (VWS) is the sequence of **syllables** delimited by space.
3. The input of Chinese word segmentation (CWS) is the sequence of **characters** WITHOUT explicit delimiter.
4. The use of a Vietnamese syllable and a Chinese character are similar.
5. Vietnamese and Chinese have similar linguistic phenomena such as overlapping ambiguity

1. **Vietnamese example:**
   1. **Input:** học sinh học sinh học
      - học_sinh: student
      - học: learn
      - sinh_học: biology
   2. **Output:** [B]học_[E]sinh [S]học [B]sinh_[E]học
      (students learn biology) (correct)
      [B]học_[E]sinh [B]học_[E]sinh [S]học
      (student student learn) (incorrect)

Introduction

2. **Chinese example:**
   1. 他 /从小/ 学/ 电脑/ 技术
      (He learned computer techniques since childhood)
      1. 从小/ 学: learn since childhood
      2. 从/ 小学: from primary school
3. **Overlap ambiguity** makes VWS and CWS challenging

Outline

2 Motivation

Motivation

1. Most of VWS and CWS methods treat word segmentation as a **token-based** problem.

Motivation

1. Most of VWS and CWS methods treat word segmentation as a **token-based** problem.
2. The intersection of VWS and CWS approaches leverage the context to model n-gram of token information.

Motivation

1. Most of VWS and CWS methods treat word segmentation as a **token-based** problem.
2. The intersection of VWS and CWS approaches leverage the context to model n-gram of token information.
3. We get the inspiration of span representation in constituency parsing to propose our model.

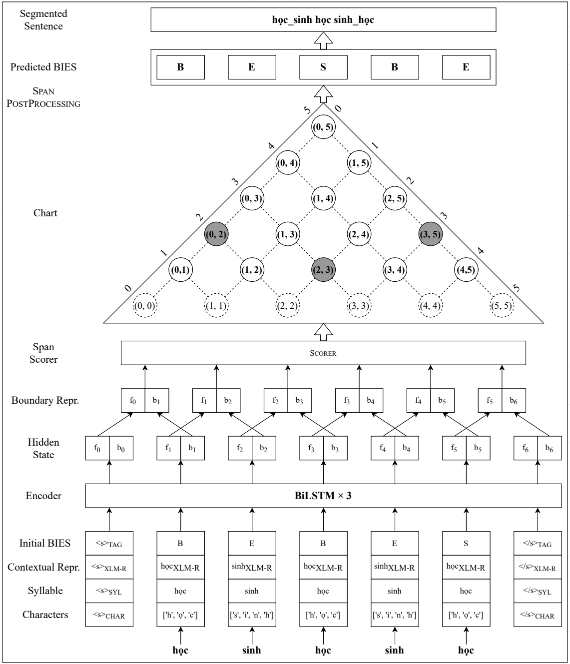
Motivation

1. Most of VWS and CWS methods treat word segmentation as a **token-based** problem.
2. The intersection of VWS and CWS approaches leverage the context to model n-gram of token information.
3. We get the inspiration of span representation in constituency parsing to propose our model.
4. Its main idea is to model n-grams in the input sentence and score them.

Outline

3 The Proposed Framework

The Proposed Framework for VWS



Word segmentation as span labeling task for Vietnamese and Chinese

- Input: a sequence of characters $\mathcal{X} = x_1 x_2 \ldots x_n$ with the length of n.
- Output: a sequence of words $\mathcal{W} = w_1 w_2 \ldots w_m$ with the length of m.
- Notations:
  1. We use $x_i x_{i+1} \ldots x_{i+k-1}$ to denote that the word w_j is constituted by k consecutive characters beginning at character x_i, where $1 \leq k \leq n$.
  2. We get the inspiration of span representation in constituency parsing[1] to use the span $(i-1, i-1+k)$ representing the word constituted by k consecutive characters $x_i x_{i+1} \ldots x_{i+k-1}$.

[1] [1] Stern et al. "A Minimal Span-Based Neural Constituency Parser". 2017.

Word segmentation as span labeling task for Vietnamese and Chinese (continue)

- Formally, the problem of our SpanSeg for VWS and CWS can be formalized as:

$$\hat{\mathcal{Y}}_{\text{novlp}} = \textsc{SpanPostProcessing}(\hat{\mathcal{Y}}) \quad (1)$$

  $\textsc{SpanPostProcessing}(\hat{\mathcal{Y}})$ solely is an algorithm for producing the word segmentation boundary guaranteeing non-overlapping between every two spans.

Word segmentation as span labeling task for Vietnamese and Chinese (continue)

- The $\hat{\mathcal{Y}}$ is the set of predicted spans as follows:

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

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

Decoding Algorithm for Predicted Span

- Our algorithm named SpanPostProcessing deals with overlapping ambiguity and missing spans from prediced spans.
  1. Keeping the spans with the highest score and eliminate the remainder among overlapping spans.
  2. Adding 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.

Algorithm 1 SPANPOSTPROCESSING

Require:

The input sentence $\mathcal{X}$ with the length of n;
The scoring module $\textsc{Scorer}(\cdot)$ for any span (l, r) in $\mathcal{X}$, where $0 \le l \le n-1$ and $l < r \le n$;
The set of predicted spans $\hat{\mathcal{Y}}$, sorted in ascending order.

Ensure:

The list of valid predicted spans $\hat{\mathcal{S}}$, satisfying non-overlapping between every two spans.

$\hat{\mathcal{S}}_{\text{novlp}} = [(0,0)]$ ▷ The list of predicted spans without overlapping ambiguity.
$\hat{\mathcal{S}} = []$ ▷ The final list of valid predicted spans.
for $\hat{y}$ **in** $\hat{\mathcal{Y}}$ **do** ▷ The $\hat{y}[0]$ is the left boundary and $\hat{y}[1]$ is the right boundary of each span $\hat{y}$.
  if $\hat{\mathcal{S}}_{\text{novlp}}[\text{-}1][1] < \hat{y}[0]$ **then** ▷ Check for missing boundary.
    $\hat{\mathcal{S}}_{\text{novlp}}$.**append**$\big((\hat{\mathcal{S}}_{\text{novlp}}[\text{-}1][1], \hat{y}[0])\big)$ ▷ Add the missing span to $\hat{\mathcal{S}}_{\text{novlp}}$
  end if
  if $\hat{\mathcal{S}}_{\text{novlp}}[\text{-}1][0] \le \hat{y}[0] < \hat{\mathcal{S}}_{\text{novlp}}[\text{-}1][1]$ **then** ▷ Check for overlapping ambiguity.
    if $\textsc{Scorer}(\mathcal{X}, \hat{\mathcal{S}}_{\text{novlp}}[\text{-}1][0], \hat{\mathcal{S}}_{\text{novlp}}[\text{-}1][1]) < \textsc{Scorer}(\mathcal{X}, \hat{y}[0], \hat{y}[1])$ **then**
      $\hat{\mathcal{S}}_{\text{novlp}}$.**pop**() ▷ Remove the span causing overlapping with the lower score than $\hat{y}$.
      $\hat{\mathcal{S}}_{\text{novlp}}$.**append**$\big((\hat{y}[0], \hat{y}[1])\big)$ ▷ Add the span $\hat{y}$ to $\hat{\mathcal{S}}_{\text{novlp}}$.
    end if
  else
    $\hat{\mathcal{S}}_{\text{novlp}}$.**append**$\big((\hat{y}[0], \hat{y}[1])\big)$ ▷ Add the span $\hat{y}$ to $\hat{\mathcal{S}}_{\text{novlp}}$.
  end if
end for
if $\hat{\mathcal{S}}_{\text{novlp}}[\text{-}1][1] < n$ **then** ▷ Check for missing boundary.
  $\hat{\mathcal{S}}_{\text{novlp}}$.**append**$\big((\hat{\mathcal{S}}_{\text{novlp}}[\text{-}1][1], n)\big)$ ▷ Add the missing span to $\hat{\mathcal{S}}_{\text{novlp}}$
end if
for $i, \hat{y}$ **in enumerate**($\hat{\mathcal{S}}_{\text{novlp}}$) **do** ▷ The $\hat{y}[0]$ is the left boundary and $\hat{y}[1]$ is the right boundary of each span $\hat{y}$, and i is the index of $\hat{y}$ in list $\hat{\mathcal{S}}_{\text{novlp}}$.
  if $0 < i$ **and** $\hat{\mathcal{S}}_{\text{novlp}}[i-1][1] < \hat{y}[0]$ **then** ▷ Check for missing boundary.
    missed_boundaries $= \big[\hat{\mathcal{S}}_{\text{novlp}}[i-1][1]\big]$
    for *bound* **in range**$\big(\hat{\mathcal{S}}_{\text{novlp}}[i-1][1], \hat{y}[0]\big)$ **do**
      if $\textsc{Scorer}(\mathcal{X}, \textit{bound}, \textit{bound}+1) > 0.5$ **then** ▷ Check for single word.
        missed_boundaries.**append**(*bound* $+ 1$)
      end if
    end for
    missed_boundaries.**append**($\hat{y}[0]$)
    for *j* **in range**$\big($**len**(*missed_boundaries*) $- 1\big)$ **do**
      $\hat{\mathcal{S}}$.**append**$\big((\textit{missed\_boundaries}[j], \textit{missed\_boundaries}[j+1])\big)$ ▷ Add the missing span to $\hat{\mathcal{S}}$
    end for
  end if
  $\hat{\mathcal{S}}$.**append**$\big(\hat{y}[0], \hat{y}[1]\big)$ ▷ Add the non-overlapping span to $\hat{\mathcal{S}}$
end for

Span Scoring for Word Segmentation

- We have the left $\mathbf{r}_i^{\text{left}}$ and right $\mathbf{r}_i^{\text{right}}$ boundary representations of token x_i as following:

$$\mathbf{r}_i^{\text{left}} = \text{MLP}^{\text{left}}(\mathbf{f}_{i-1} \oplus \mathbf{b}_i) \quad (3)$$

$$\mathbf{r}_i^{\text{right}} = \text{MLP}^{\text{right}}(\mathbf{f}_i \oplus \mathbf{b}_{i+1}) \quad (4)$$

- Finally, inspired by[2], given the input sentence $\mathcal{X}$, the span scoring module $\textsc{Scorer}(\cdot)$ for span (l, r) in our $\textsc{SpanSeg}$ model is computed by using a biaffine operation over the left boundary representation of token x_l and the right boundary representation of token x_r as following:

$$\textsc{Scorer}(\mathcal{X}, l, r) = \text{sigmoid}\left(\begin{bmatrix} \mathbf{r}_l^{\text{left}} \\ 1 \end{bmatrix}^{\mathrm{T}} \mathbf{W} \mathbf{r}_r^{\text{right}} \right) \quad (5)$$

  where $\mathbf{W} \in \mathbb{R}^{d \times d}$.
- To sum up, the $\textsc{Scorer}(\mathcal{X}, l, r)$ gives us a score to predict whether a span (l, r) is a word.

[2] [2] Zhang et al. "Fast and Accurate Neural CRF Constituency Parsing". 2020.

Outline

Statistics of the Vietnamese treebank dataset for word segmentation

Table 1: We provide the number of sentences, characters, syllables, words, character types, syllable types, word types. We also compute the out-of-vocabulary (OOV) rate as the percentage of unseen words in the development and test set.

	VTB		
	Train	Dev	Test
# sentences	74,889	500	2,120
# characters	6,779,116	55,476	307,932
# syllables	2,176,398	17,429	96,560
# words	1,722,271	13,165	66,346
# character types	155	117	121
# syllable types	17,840	1,785	2,025
# word types	41,355	2,227	3,730
OOV Rate	-	2.2	1.6

Statistics of five Chinese benchmark dataset for word segmentation

Table 2: We provide the number of sentences, characters, words, character types, word types. We also compute the out-of-vocabulary (OOV) rate as the percentage of unseen words in the test set.

	MSR		PKU		AS		CityU		CTB6		
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Dev	Test
# sentences	86,918	3,985	19,054	1,944	708,953	14,429	53,019	1,492	23,420	2,079	2,796
# characters	4,050,469	184,355	1,826,448	172,733	8,368,050	197,681	2,403,354	67,689	1,055,583	100,316	134,149
# words	2,368,391	106,873	1,109,947	104,372	5,449,581	122,610	1,455,630	40,936	641,368	59,955	81,578
# character types	5,140	2,838	4,675	2,918	5,948	3,578	4,806	2,642	4,243	2,648	2,917
# word types	88,104	12,923	55,303	13,148	140,009	18,757	68,928	8,989	42,246	9,811	12,278
OOV Rate	-	2.7	-	5.8	-	4.3	-	7.2	-	5.4	5.6

Outline

Main Result on the VWS dataset

Table 3: Performance (F-score) comparison between SpanSeg (with different configurations) and previous state-of-the-art models on the test set of VTB dataset. The initial word boundary tags (TAG) were predicted by RDRsegmenter [3].

	VTB			
	P	R	F	R_{OOV}
vnTokenizer [4]	96.98	97.69	97.33	-
JVnSegmenter-Maxent [5]	96.60	97.40	97.00	-
JVnSegmenter-CRFs [5]	96.63	97.49	97.06	-
DongDu [6]	96.35	97.46	96.90	-
UETsegmenter [7]	97.51	98.23	97.87	-
RDRsegmenter [3]	97.46	98.35	97.90	-
UITsegmenter [8]	97.81	**98.57**	98.19	-
BiLSTM-CRF	97.42	97.84	97.63	72.47
SpanSeg	97.58	97.94	97.76	**74.65**
BiLSTM-CRF (XLM-R)	97.69	97.99	97.84	72.66
SpanSeg (XLM-R)	97.75	98.16	97.95	70.01
BiLSTM-CRF (TAG)	97.91	98.28	98.10	69.16
SpanSeg (TAG)	97.67	98.28	97.97	65.94
BiLSTM-CRF (TAG+XLM-R)	97.94	98.44	98.19	68.87
SpanSeg (TAG+XLM-R)	**98.21**	98.41	**98.31**	72.28

Main Result on the CWS datasets

Table 4: Performance (F-score) comparison between SpanSeg (BERT and ZEN) and previous state-of-the-art models on the test set of five Chinese benchmark datasets. The symbol [★] denotes the methods learning from data annotated through different segmentation criteria, which means that the labeled training data are different from the rest.

	MSR		PKU		AS		CityU		CTB6	
	F	R_{OOV}	F	R_{OOV}	F	R_{OOV}	F	R_{OOV}	F	R_{OOV}
Chen et al. [9]	97.40	-	96.50	-	-	-	-	-	96.00	-
Xu and Sun [10]	96.30	-	96.10	-	-	-	-	-	95.80	-
Zhang et al. [11]	97.70	-	95.70	-	-	-	-	-	95.95	-
Chen et al. [12] [★]	96.04	71.60	94.32	72.64	94.75	75.34	95.55	81.40	-	-
Wang and Xu [13]	98.00	-	96.50	-	-	-	-	-	-	-
Zhou et al. [14]	97.80	-	96.00	-	-	-	-	-	96.20	-
Ma et al. [15]	98.10	80.00	96.10	78.80	96.20	70.70	97.20	87.50	96.70	85.40
Gong et al. [16]	97.78	64.20	96.15	69.88	95.22	77.33	96.22	73.58	-	-
Higashiyama et al. [17]	97.80	-	-	-	-	-	-	-	96.40	-
Qiu et al. [18] [★]	98.05	78.92	96.41	78.91	96.44	76.39	96.91	86.91	-	-
WMSeg (BERT-CRF) [19]	98.28	**86.67**	96.51	**86.76**	96.58	78.48	97.80	87.57	97.16	88.00
WMSeg (ZEN-CRF) [19]	**98.40**	84.87	96.53	85.36	**96.62**	**79.64**	97.93	**90.15**	**97.25**	**88.46**
MetaSeg [20] [★]	98.50	-	96.92	-	97.01	-	98.20	-	97.89	-
SpanSeg (BERT)	98.31	85.32	**96.56**	85.53	**96.62**	79.36	97.74	87.45	**97.25**	87.91
SpanSeg (ZEN)	98.35	85.66	96.35	83.66	96.52	78.43	**97.96**	90.11	97.17	87.76

Anlysis I: Practical complexity

Table 5: Statistics of model size (MB) and inference time (minute) of WMSEG [19] and our SPANSEG dealing with the training set of the AS dataset on Chinese. We use the same batch size as the work of Tian et al. [19]. The inference time is done by using Tesla P100-PCIE GPU with memory size of 16,280 MiB via Google Colaboratory.

	BERT Encoder		ZEN Encoder	
	WMSeg	**SpanSeg**	**WMSeg**	**SpanSeg**
Size (MB)	704	397	1,150	872
Inference Time (minute)	28	15	46	32

Anlysis II: Error statistics of the overlapping ambiguity problem involving three consecutive tokens on VWS dataset

Table 6: The symbols ✓ and ✗ denote predicting correctly and incorrectly, respectively.

BiLSTM-CRF	SpanSeg	Configuration			
		Defalut	XLM-R	TAG	TAG+XLM-R
✗	✗	15	5	19	7
✓	✗	**7**	**0**	4	0
✗	✓	**7**	**0**	**18**	**1**

Anlysis III: Error statistics of the overlapping ambiguity problem involving three consecutive tokens on five Chinese benchmark datasets

Table 7: The symbols ✓ and ✗ denote predicting correctly and incorrectly, respectively.

WMSeg [19]	**SpanSeg**	**MSR**	**PKU**	**AS**	**CityU**	**CTB6**
✗	✗	14	13	12	2	3
✓	✗	**2**	**2**	2	**1**	**2**
✗	✓	**2**	1	**5**	0	0

Outline

Conclusion

1. This paper proposes a span labeling approach, namely SpanSeg, for VWS. Straightforwardly, our approach encodes the n-gram information by using span representations.

Conclusion

1. This paper proposes a span labeling approach, namely SpanSeg, for VWS. Straightforwardly, our approach encodes the n-gram information by using span representations.
2. The experimental results on VWS show that our SpanSeg is better than BiLSTM-CRF when utilizing the predicted word boundary and contextual information with the state-of-the-art F-score of 98.31%.

Conclusion

1. This paper proposes a span labeling approach, namely SpanSeg, for VWS. Straightforwardly, our approach encodes the n-gram information by using span representations.
2. The experimental results on VWS show that our SpanSeg is better than BiLSTM-CRF when utilizing the predicted word boundary and contextual information with the state-of-the-art F-score of 98.31%.
3. On CWS, our SpanSeg achieves competitive or higher F-scores through experimental results, fewer parameters, and faster inference time than the previous state-of-the-art method, WMSeg.

Conclusion

1. This paper proposes a span labeling approach, namely SpanSeg, for VWS. Straightforwardly, our approach encodes the n-gram information by using span representations.
2. The experimental results on VWS show that our SpanSeg is better than BiLSTM-CRF when utilizing the predicted word boundary and contextual information with the state-of-the-art F-score of 98.31%.
3. On CWS, our SpanSeg achieves competitive or higher F-scores through experimental results, fewer parameters, and faster inference time than the previous state-of-the-art method, WMSeg.
4. For the future work, we will explore the architecture of BERT for span representations for word segmentation to reduce time complexity caused by biaffine operation.

Outline

Thank you for your time!

References I

[1] Mitchell Stern, Jacob Andreas, and Dan Klein. "A Minimal Span-Based Neural Constituency Parser". In: *Proceedings of ACL*. 2017, pp. 818–827.

[2] Yu Zhang, Houquan Zhou, and Zhenghua Li. "Fast and Accurate Neural CRF Constituency Parsing". In: *Proceedings of IJCAI*. 2020, pp. 4046–4053.

[3] Dat Quoc Nguyen et al. "A Fast and Accurate Vietnamese Word Segmenter". In: *Proceedings of LREC*. 2018, pp. 2582–2587.

[4] Hong-Phuong Le et al. "A Hybrid Approach to Word Segmentation of Vietnamese Texts". In: *Language and Automata Theory and Applications*. Springer Berlin Heidelberg, 2008, pp. 240–249.

[5] Cam-Tu Nguyen et al. "Vietnamese Word Segmentation with CRFs and SVMs: An Investigation". In: *Proceedings of PACLIC*. Tsinghua University Press, 2006, pp. 215–222.

References II

[6] T. A. Luu and K. Yamamoto. *Ứng dụng phương pháp Pointwise vào bài toán tách từ cho tiếng Việt*. 2012. URL: http://www.vietlex.com/xu-li-ngon-ngu/117-Ung_dung_phuong_phap_Pointwise_vao_bai_toan_tach_tu_cho_tieng_Viet.

[7] T. P. Nguyen and A. C. Le. "A hybrid approach to Vietnamese word segmentation". In: *Proceeding of IEEE-RIVF*. 2016, pp. 114–119.

[8] Duc-Vu Nguyen et al. "Vietnamese Word Segmentation with SVM: Ambiguity Reduction and Suffix Capture". In: *Proceedings of PACLING*. 2019, pp. 400–413.

[9] Xinchi Chen et al. "Long Short-Term Memory Neural Networks for Chinese Word Segmentation". In: *Proceedings of EMNLP*. 2015, pp. 1197–1206.

[10] Jingjing Xu and Xu Sun. "Dependency-based Gated Recursive Neural Network for Chinese Word Segmentation". In: *Proceedings of ACL)*. 2016, pp. 567–572.

References III

[11] Meishan Zhang, Yue Zhang, and Guohong Fu. "Transition-Based Neural Word Segmentation". In: *Proceedings of ACL*. 2016, pp. 421–431.

[12] Xinchi Chen et al. "Adversarial Multi-Criteria Learning for Chinese Word Segmentation". In: *Proceedings of ACL*. 2017, pp. 1193–1203.

[13] Chunqi Wang and Bo Xu. "Convolutional Neural Network with Word Embeddings for Chinese Word Segmentation". In: *Proceedings of IJCNLP*. 2017, pp. 163–172.

[14] Hao Zhou et al. "Word-Context Character Embeddings for Chinese Word Segmentation". In: *Proceedings of EMNLP*. 2017, pp. 760–766.

[15] Ji Ma, Kuzman Ganchev, and David Weiss. "State-of-the-art Chinese Word Segmentation with Bi-LSTMs". In: *Proceedings of EMNLP*. 2018, pp. 4902–4908.

References IV

[16] Jingjing Gong et al. “Switch-LSTMs for Multi-Criteria Chinese Word Segmentation”. In: *Proceedings of AAAI*. 2019, pp. 6457–6464.

[17] Shohei Higashiyama et al. “Incorporating Word Attention into Character-Based Word Segmentation”. In: *Proceedings of NAACL*. 2019, pp. 2699–2709.

[18] Xipeng Qiu et al. “A Concise Model for Multi-Criteria Chinese Word Segmentation with Transformer Encoder”. In: *Findings of EMNLP*. 2020, pp. 2887–2897.

[19] Yuanhe Tian et al. “Improving Chinese Word Segmentation with Wordhood Memory Networks”. In: *Proceedings of ACL*. 2020, pp. 8274–8285.

[20] Zhen Ke et al. “Pre-training with Meta Learning for Chinese Word Segmentation”. In: *Proceedings of NAACL*. 2021, pp. 5514–5523.