That Slepen Al the Nyght with Open Ye! Cross-era Sequence Segmentation with Switch-memory

Anonymous ACL submission

Abstract

Language evolution follows the rule of gradual change. Grammar, vocabulary, and lexical semantics shift took place over time, resulting in the diachronic linguistic gap. However, a considerable amount of texts are written in languages of different eras, which brings obstacles to natural language processing tasks, such as word segmentation and machine translation. Chinese is a language with a long history, but previous Chinese natural language processing works mainly focused on tasks in a specific era. Therefore, in this paper, we propose a cross-era learning framework for Chinese word segmentation (CWS), CROSSWISE, which uses the Switch-memory (SM) module to incorporate era-specific linguistic knowledge. Experiments on four corpora with different eras show that the performance of each corpus obtains a significant improvement. Further analyses also demonstrate that the SM can effectively integrate the knowledge of the eras into the neural network.

1 Introduction

As a human-learnable communication system, language is by no means static but evolve over time. Various aspects of language, such as grammar, vocabulary and word meaning change at different rates due to language contact and many other factors, a fact that led to the diachronic linguistic gap. For example, "That slepen al the nyght with open ye (That sleep all the night with open eye)" is a sentence from The Canterbury Tales, written in Middle English by Geoffrey Chaucer at the end of the 14th century. It's difficult for people without a background of Middle English knowledge to understand the sentence. However, some discourses may consist of modern English and Old English because of citation or rhetorical need. For instance, Shakespeare's fourteen lines of poetry is often quoted in contemporary novels. This kind of era-hybrid text brings barriers to natural language processing tasks such as word segmentation and machine translation.

Sample from MSR								
Golds	(wait)		(who)	(come)	(slove)		(ne)	?
	等待		谁	来	解决		呢	?
PKUSeg	等待		谁	来	解决		呢	?
JiaYan	等	待	谁	来	解	决	呢	?
Sample from AWIKI								
Golds	(Qi)	(Cui Shu)		(leads army)		(attack)	(Lv)	。
	齐	崔杼		帅师		伐	莒	。
PKUSeg	齐崔		杼帅		师伐莒			。
JiaYan	齐	崔杼		帅师		伐	莒	。

Table 1: Illustration of the different segmentation results for a modern Chinese sentence and an ancient Chinese sentence with different segmentation toolkits.

Having the honour of being listed as one of the oldest languages of the world, the Chinese language has seen several changes over its long history. It has undergone various incarnations, which is recognized as Archaic (Ancient) Chinese, Middle Ancient Chinese, Near Ancient Chinese, and Modern Chinese. Notably, most Chinese NLP tasks skew towards Modern Chinese. Take Chinese Word Segmentation (CWS) as an example, many previous methods mainly focused on addressing the CWS problem on Modern Chinese and achieved satisfying results (Zheng et al., 2013; Chen et al., 2015; Zhang et al., 2016; Xu and Sun, 2016; Shao et al., 2017; Yang et al., 2017; Zhang et al., 2018; Tian et al., 2020b,a). Although CWS for ancient Chinese has been noticed in recent years, the processing of language-hybrid texts is still an open question. As shown in Table 1, PKUSeg (Luo et al., 2019a) is a Chinese segmenter trained with modern Chinese corpus, which can segment the modern Chinese sentence correctly, but its accuracy drops sharply when applied to ancient Chinese. And the ancient Chinese segmenter JiaYan[1] achieves good performance on ancient Chinese text, but fails to perform well on the Modern Chinese. Therefore, it is necessary to develop appropriate models to exploit cross-era NLP tasks.

To this end, we propose CROSSWISE (CROsS-ear Segmentation WIth Switch-mEmory), a learning framework that deals with cross-era Chinese word segmentation (CECWS) task. The framework integrates era-specific knowledge with the Switch-memory mechanism to improve CWS for era-hybrid texts. More specifically, we jointly train CWS and sentence classification task in order to predict both segmentation result and era label. We utilize the Switch-memory module to incorporate knowledge of different eras, which consists of key-value memory networks (Miller et al., 2016) and a switcher. The key-value memory networks are used to store era-specific knowledge by several memory cells. And the sentence discriminator is considered as a switcher governing how much information in each memory cell will be integrated into the model. For each memory cell, we map candidate words from dictionary and word boundary information to keys and values.

The main contributions of this paper could be summarized as follows.

- Cross-era learning is first introduced for CWS, in which we share all the parameters with a multi-task architecture. The shared encoder is used to capture the common information between several datasets with different eras. This single model can produce different words segmentation granularity according to the different era.

- The Switch-memory mechanism is used to integrate era-specific knowledge into the neural network, which can help improve the performance of out of vocabulary (OOV) words. And two switcher modes (*hard-switcher* and *soft-switcher*) are proposed to control how much information in each cell will be feed into the model.

- Experimental results from four CWS datasets with different eras confirm that the performance of each corpus obtains a significant improvement. Further analyses also demonstrate that our model is flexible for cross-era Chinese word segmentation.

[1] `http://github.com/jiayan/Jiayan/`

2 Related Work

Chinese word segmentation is generally considered as a sequence labeling task, i.e. to assign a label to each character in a given sentence. In recent years, many deep learning methods have been applied to CWS successfully (Zheng et al., 2013; Chen et al., 2015; Zhang et al., 2016; Xu and Sun, 2016; Shao et al., 2017; Yang et al., 2017; Kurita et al., 2017; Liu et al., 2018; Zhang et al., 2018; Ye et al., 2019a; Higashiyama et al., 2019; Huang et al., 2020b; Tian et al., 2020b,a,c; Liu et al., 2021). Among these studies, some point out that context features and external knowledge can improve the CWS accuracy (Kurita et al., 2017; Yang et al., 2017; Zhang et al., 2018; Liu et al., 2018; Tian et al., 2020b,a,c). The studies from Liu et al. (2018) and Zhang et al. (2018) leveraged dictionary to improve the task; n-gram are also an effective context feature for CWS (Kurita et al., 2017; Tian et al., 2020b; Shao et al., 2017). Tian et al. (2020b) utilized syntactic knowledge generated by existing NLP toolkits to improve CWS and part-of-speech (POS). Tian et al. (2020c) incorporated wordwood information for neural segmenter and achieved state-of-the-art performance at that time.

It is a common practice to jointly train CWS and other related tasks based on a multi-task framework. Chen et al. (2017) took each segmentation criterion as a single task, and proposed an adversarial multi-task learning framework for multi-criteria CWS by extracting shared knowledge from multiple segmentation datasets. Yang et al. (2017) investigated the effectiveness of several external sources for CWS by a globally optimized beam-search model. They considered each type of external resource as an auxiliary classification, then leveraged multi-task learning to pre-train the shared parameters used for the context modeling of Chinese characters. Liu et al. (2018) jointly trained the CWS and word classification task by a unified framework model. Inspired by these successful studies, we also borrow ideas from the multi-task framework, and jointly train the CWS task and the sentence classification task to boost the performance of cross-era CWS.

Recently, some studies have noticed the linguistic gap due to the differences in eras. Ceroni et al. (2014) proposed a time-aware re-contextualization approach to bridge the temporal context gap. Chang et al. (2021) reframed the translation of ancient Chinese text as a multi-label prediction task, then predicted both translation and its particular

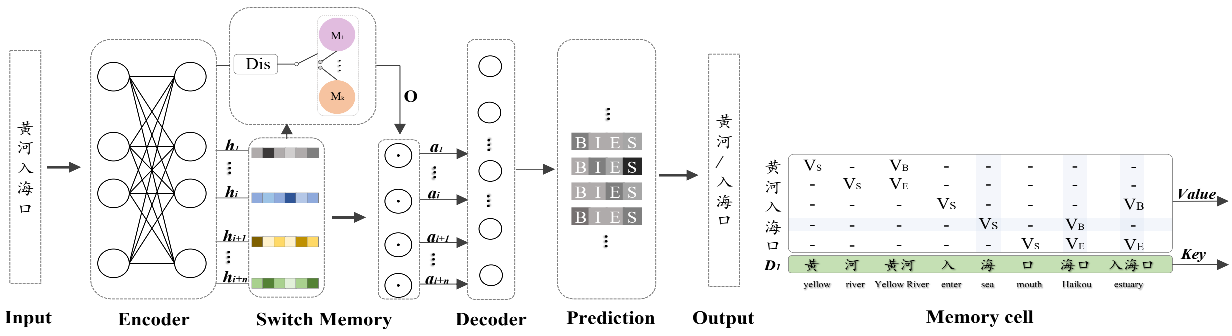


Figure 1: CROSSWISE for cross-era Chinese word segmentation. "Dis" represents the discriminator, namely sentence classifier. "M_1" is the first memory cell, its internal structure as shown at the right of the figure. For each character, the first memory cell extracts all candidate words from the input sentence and only keeps ones that appeared in the first dictionary (candidate words as keys, words' boundary information as value).

era by dividing ancient Chinese into three periods.

Key-value memory networks were introduced to the task of directly reading documents and answering questions by Miller et al. (2016), which helped bridge the gap between direct methods and using human-annotated or automatically constructed Knowledge Bases. Tian et al. (2020c) utilized this mechanism to incorporate n-grams into the neural model for CWS.

Encouraged by the above works, we design a multi-task model for cross-era CWS, jointly train the sentence classification task and CWS by a unified framework model. Key-value memory networks are used to integrate era-specific knowledge into the neural network follow Tian et al. (2020c).

3 The Proposed Framework

3.1 BERT-CRF model for Chinese word Segmentation

Chinese word segmentation is generally viewed as a character-based sequence labeling task. In detail, given a sentence $X = \{x_1, x_2, ...x_T\}$, each character in the sequence is labeled as one of $\mathcal{L} = \{B, M, E, S\}$, indicating the character is at the beginning, middle, end of a word, or the character is a single-character word. CWS aims to figure out the ground truth of labels $Y^* = \{y_1^*, y_2^*, \dots y_T^*\}$:

$$Y^* = arg\ max\ P\left(Y|X\right)_{Y \in \mathcal{L}^T} \tag{1}$$

The universal end-to-end neural CWS architecture usually contains an encoder and a decoder.

Encoding layer. According to Fu et al. (2020), although BERT-based (Devlin et al., 2019) models for CWS are not impeccable, BERT is inferior to un-pre-training models in many aspects, such as BERT is more suitable for dealing with long sentences. therefore, we utilize BERT released by Devlin et al. (2019) as the shared encoder, which is pre-trained with a large number of unlabeled Chinese data.

$$\{\mathbf{h}_1, ..., \mathbf{h}_i, ..., \mathbf{h}_T\} = Encoder(\{x_1, ..., x_i, ...x_T\}) \tag{2}$$

where $\mathbf{h}_i$ is the representation for x_i from the encoder.

Decoding layer. In this work, we use a shared decoder for different eras' samples, since we combined era-aware representation for each character by the Switch-memory module. There are different algorithms that can be implemented as decoders, such as random conditional fields (CRF) (Lafferty et al., 2001) and softmax. In our framework, we use CRF as the decoder.

In CRF layer, $P\left(Y|X\right)$ in Eq. 1 could be represented as:

$$P(Y|X) = \frac{\emptyset(Y|X)}{\sum_{Y' \in \mathcal{L}^T} \emptyset(Y'|X)} \tag{3}$$

where, $\emptyset(Y|X)$ is the potential function, and we only consider interactions between two successive labels.

$$\emptyset(Y|X) = \prod_{i=2}^{T} \sigma(X, i, y_{i-1}, y_i) \tag{4}$$

$$\sigma(\mathbf{x}, i, y', y) = exp(s(X, i)_y + b_{y'y}) \tag{5}$$

where $b_{y'y} \in \mathbf{R}$ is trainable parameters respective to label pair (y', y). The score function $s(X, i) \in \mathbb{R}^{|\mathcal{L}|}$ calculate the score of each lable for i_{th} character:

Rule	$V_{i,j}$
x_i is the beginning character of $w_{i,j}$.	V_B
x_i is the ending character of $w_{i,j}$.	V_E
x_i is a single word, $w_{i,j}$.	V_S

Table 2: the rules for assigning different values to x_i according to its position in word $w_{i,j}$.

$$s(X\ i) = \mathbf{W}_s^\top \mathbf{a}_i + b_s \quad (6)$$

where $\mathbf{a}_i$ is the final representation for i_{th} character. $\mathbf{W}_s \in \mathbb{R}^{d_a \times L}$ and $b_s \in \mathbb{R}^{|\mathcal{L}|}$ are trainable parameters.

3.2 Switch-memory mechanism

The Switch-memory consists of d memory cells and a switcher. For an input sentence, there are d memory cells for each character. The switcher govern how much information in each cell will be integrated into the network. And the state of the switcher depends on sentence classification task.

3.2.1 Memory cells

Dictionary has been used as an useful external source to improve the performance for CWS in many studies(Yang et al., 2017; Liu et al., 2018; Zhang et al., 2018). However, the method of incorporating dictionary for previous studies is limited in either concatenating candidate words and character embeddings or requiring handcrafted templates. In this work, we utilize key-value memory networks to incorporate dictionary information, which is initially applied to the Question Answering(QA) task for better storage of prior knowledge required by QA. Intuitively, we can also use this network structure to store the prior knowledge required by cross-era CWS.

At a fine-grained view, the notion of "ancient Chinese" may not be considered a single language with a static word-meaning mapping. Ancient Chinese has three development stages: Ancient, Middle Ancient, and Near Ancient. Each stage has specific lexicon and word segmentation granularity. Therefore, we construct four dictionaries $\mathcal{D} = \{D_0, D_1, D_2, D_3\}$, associating with the four development stages of Chinese respectively, and each dictionary is era-related. Given a sentence, four memory cells are generated for each character in the sentence according to the four dictionaries, and each memory cell will map candidate words and word boundary information to keys and values.

Candidate words as keys. Following Miller et al., for each x_i in the input sentence, each dictionary has many words containing x_i, we only keep the n-grams from x_i's context and appear in each dictionary, resulting $w_i^d = \{w_{i,1}^d,\ w_{i,2}^d, ...w_{i,j}^d...w_{i,m_i}^d\}$, x_i is a part of word $w_{i,j}^d \in D_d$, $d \in [0, 3]$. We use an example to illustrate our idea. For the input sentence show in Figure 1, there are many n-grams for x_3 = "海(sea)", we only keep ones that appear in D_0 for the first memory cell, thus, $w_3^0 =$ {"海口(HaiKou)", "入海口(estuary)", "海(sea)"}. Similarly, we can generate w_3^1, w_3^2, w_3^3 for the second, third and fourth memory cell according to D_1, D_2, D_3. Then, the memory cell compute the probability for each key (which are denoted as $e_{i,j}^w$ for each $w_{i,j}^d$), here $\mathbf{h}_i$ is the embedding for x_i, which is encoded by the encoder.

$$p_{i,j}^d = \frac{exp(\mathbf{h}_i \cdot e_{i,j}^w)}{\sum_{j=1}^{m_i} exp(\mathbf{h}_i \cdot e_{i,j}^w)} \quad (7)$$

Word boundary information as values. As we know, CWS aims to find the best segment position. However, each character x_i may have different position in each $w_{i,j}^d$. For example, x_i may be at the beginning, middle, ending of $w_{i,j}^d$, or x_i may form a single word. Different positions convey different information. Therefore, we use the boundary information of candidate words as values for key-value networks. As shown in Table 2, a set of word boundary value $\{V_B, V_E, V_S\}$ with embeddings $\{e_{V_B}, e_{V_E}, e_{V_S}\}$ represent the x_i's different positions in $w_{i,j}^d$, and we map x_i to different value vectors according to its positions. As a result, each w_i^d for x_i has a values list $\mathcal{V}_i^d = [v_{i,1}^d, v_{i,2}^d, ...v_{i,j}^d, ...v_{i,m_i}^d]$. In Figure 1, x_3 = "海(sea)", for the first memory cell, we can map candidate word boundary information to the value list $\mathcal{V}_3^0 = [V_S, V_B]$. Four cells for x_i has a values list $\mathcal{V}_i = [v_i^0, v_i^1, v_i^2, v_i^3]$. Then the d_{th} memory cell embedding for x_i is computed from the weighted sum of all keys and values as follow.

$$\mathbf{o}_i^d = \sum_{j=1}^{m_i} p_{i,j}^d e_{i,j}^{v^d} \quad (8)$$

where $e_{i,j}^{v^d}$ is the embedding for $v_{i,j}^d$. Next, the final character embedding is the element-wise sum of $\mathbf{o}_i$ and $\mathbf{h}_i$, or their concatenation, passing through a fully connected layer as follow:

$$\mathbf{a}_i = \mathbf{W}_o \cdot (\mathbf{o}_i \odot \mathbf{h}_i) \quad (9)$$

where $\odot$ operation could be sum or concatenate, $\mathbf{W}_o$ is a trainable parameter and the output $\mathbf{a}_i \in \mathbb{R}^T$ is the final representation for the i_{th} character. $\mathbf{o}_i$ is the final memory embedding for the i_{th} character, and can be calculated as follow.

$$\mathbf{o}_i = Switcher([\mathbf{o}_i^0, \mathbf{o}_i^1, \mathbf{o}_i^2, \mathbf{o}_i^3]) \qquad (10)$$

The Switcher is used to control how much information in each memory cell will be combined with the output of the encoder.

3.2.2 The switcher

Inspired by the efforts of multi-task, we add a discriminator network on the top of the source encoder to predict the era label of the input sentence. The discriminator predicts the probability of the correct era label z conditioned on the hidden states of the encoder $\mathbf{H}$. The loss function of the discriminator is $\mathcal{J}_{disc} = -logP(z|\mathbf{H})$, through minimizing the negative cross-entropy loss to maximizes $P(z|\mathbf{H})$. The predicted result is not only used to switch memory cells, intuitively, but it also forces the encoder to encode era-related information into the features it generates.

For our work, we feed $\mathbf{H}$ into a fully-connected layer and let it pass through a softmax layer to obtain probabilities for each era label.

Switch mode. For the switcher, we propose two switcher modes, *hard-switcher* and *soft-switcher*. *Hard-switcher* switches memory cells according to the final predict result from the discriminator. For the input sentence in Figure 1, if the predict result is the modern era, the switcher will switch to the memory cell associated with modern Chinese, and $\mathbf{o}_i = \mathbf{o}_i^d$. *Soft-switcher* switches memory cells according to the predict probability, which will be taken as the weight for each memory cell. *Soft-switcher* means there are some other datasets corresponding dictionary information that will be incorporated into the current sentence representation. For example, the predict probability list is $[0.1, 0.2, 0.1, 0.6\ ]$, therefore, the final memory representation for i_{th} character is $\mathbf{o}_i = \mathbf{o}_i^0 * 0.1 + \mathbf{o}_i^1 * 0.2 + \mathbf{o}_i^2 * 0.1 + \mathbf{o}_i^3 * 0.6$.

3.2.3 Objective

In our framework, the discriminator is optimized jointly with the CWS task, both sharing the same encoding layer. We assign different weights to the loss of the two tasks, the final loss function is:

$$\mathcal{J} = \alpha \mathcal{J}_{cws} + (1 - \alpha)\, \mathcal{J}_{disc} \qquad (11)$$

where α is the weight that controls the interaction of the two losses. $\mathcal{J}_{cws}$ is the negative log likelihood of true labels on the training set.

$$\mathcal{J}_{cws} = -\sum_{n=1}^{N} log(P(Y_n|X_n)) \qquad (12)$$

where N is the number of samples for training set, and Y_n is the ground truth tag sequence of the n_{th} sample.

4 Experiment

4.1 Datasets

We evaluate our proposed architecture on four CWS datasets from Academia Sinica Ancient Chinese Corpus[2] (ASACC) and SIGHAN 2005 (Emerson, 2005). Table 3 lists the statistics of all datasets. Among these datasets, PKIWI, DKIWI, AKIWI from ASACC, corresponding to near ancient Chinese, middle ancient Chinese, ancient Chinese respectively, and MSR from SIGHAN 2005 is a modern Chinese CWS dataset. Note that PKIWI, DKIWI, and AKIWI are traditional Chinese, and we translate them into simplified Chinese before segmentation.

For PKIWI, DKIWI, and AKIWI, we randomly pick 5K examples as test set, and randomly pick 10% instances from training set as the development set for all the datasets. Similar to previous work (Chen et al., 2017), we preprocess all datasets by replacing Latin characters, digits, and punctuation with a unique token.

4.2 Experimental Configurations

In our experiments, for the encoder BERT, we follow the default setting of the BERT (Devlin et al., 2019). The key embedding size and value embedding size are the same as the output of the encoder, and we random initialize them. For the baseline model Bi-LSTM, we set character embedding size to 300 and set the hidden state to 100. For the transformer, we follow the settings as Qiu et al. (2020). The loss weight coefficient α is a hyper-parameter to balance the classification loss and segmentation loss, we searched for α from 0 to 1 by setting an equal interval to 0.1, and the model achieves the best performance when α is set to 0.7.

We use the words from the training set as the internal dictionary, and each training set generates

[2] http://lingcorpus.iis.sinica.edu.tw/ancient

Datasets			Words	Chars	Word types	Char Types	Sents	OOV Rate
ASACC	AKIWI	Train	2.8M	3.2M	65.3K	7.5K	59.7K	-
		Test	0.2M	0.3M	15.7K	4.4K	5K	4.35%
	DKIWI	Train	2.2M	2.8M	44.3K	6.0K	50.1K	-
		Test	0.2M	0.3M	13.0K	3.8K	5K	4.91%
	PKIWI	Train	6.4M	7.8M	117.0K	7.2K	144.1K	-
		Test	0.2M	0.3M	18.6K	4.4K	5K	1.71%
SIGHAN05	MSR	Train	2.4M	4.1M	88.1K	5.2K	86.9K	-
		Test	0.1M	0.2M	12.9K	2.8K	4.0K	2.60%

Table 3: Detail of the four datasets.

a dictionary. The simplified Chinese dictionary sourced from jieba [3] is used as the external dictionary for MSR, and we extract words from *The ErYa* (an ancient dictionary) and ancient Chinese textbooks as the external dictionary for AWIKI. In particular, for PWIKI and DWIKI, we use high-frequency bi-grams and tri-grams extracted from the corresponding period corpus [4] as external dictionaries.

4.3 Overall results

In this section, we first give the experimental results of the proposed model on test sets of four cross-era CWS datasets. The experimental results on the aforementioned four datasets are shown in Table 4, where the F1 score and the OOV recall rate are reported.

There are several observations drawn from the results. First, we compare BERT-CRF in single-era scenario (ID:1 in Table 4) and cross-era learning without the SM module (ID:6). As can be seen from the table, when mixing four datasets, the average F1 value on all datasets slightly drops. Single-era dataset learning obtains 97.61 in average F1 value, while cross-era learning without the Switch-memory module obtains 97.32 average F1 value. It shows that the performance cannot be improved by merely mixing several datasets.

Second, these models with the SM mechanism (ID:3,5,7) outperform those baseline models (ID:2,4,6) in terms of F1 value and R_{oov} on all datasets. For instance, BERT-CRF with SM module (ID:7) gains 1.09% improvement on the average F1 score compared with BERT-CRF(ID:6), and the average R_{oov} improves from 76.15 to 82.37. It indicates that the Switch-memory can help improve segmentation and R_{oov} performance by integrating era-specific knowledge.

Third, among different encoders, the improvement of pre-trained encoder BERT on F1 value is still decent. When using Bi-LSTM as the encoder (ID:2,3), the average F1 value and the R_{oov} is 89.15, 90.66, respectively. When using BERT as the encoder (ID:6,7), the F1 value obtains about 8% improvement. The reason may be that the pre-training processing supplements some effective external knowledge.

To further illustrate the validity and the effectiveness of our model, we compare our best result on four datasets with some previous state-of-the-art works. Multi-domain and multi-criteria Chinese word segmentation are very similar to our task in some aspects, and therefore we also reproduce experiments on several previous word segmentation models with four datasets (Luo et al., 2019b; Qiu et al., 2020; Huang et al., 2020a). For the multi-domain segmenter PKUSeg (Luo et al., 2019b), we train four datasets with pre-trained mixed model, respectively. The comparison is shown in Table 5, and our model outperforms previous methods.

4.4 Ablation study

Table 6 shows the effectiveness of each component in the SM module.

The first ablation study is to verify the effectiveness of memory cells. In this experiment, the sentence classification task is no longer a switcher, it's simply a joint training task with word segmentation. We can see that ancient Chinese datasets (AWIKI, DWIKI, PWIKI) are more sensitive to the memory cells than MSR. This may be explained by the fact that the encoder is pre-trained with a large number of modern Chinese data, and our memory cells incorporate some ancient era knowledge into the model, and help boost the performance on the

[3] github.com/fxsjy/jieba/tree/master/jieba/dict.txt
[4] http://core.xueheng.net/

NO.	En-De		AWIKI	PWIKI	DWIKI	MSR	Avg.
Single-era learning							
1	BT-CRF	F	97.62	97.58	97.19	98.03	97.61
		R_{oov}	68.85	76.58	74.80	**86.85**	76.77
Cross-era learning							
2	BL-CRF	F	89.78	85.98	87.04	93.81	89.15
		R_{oov}	45.55	46.43	37.51	58.06	46.89
3	BL-CRF+SM	F	90.66	87.41	89.18	95.42	90.66
		R_{oov}	43.48	44.40	32.78	78.74	49.76
4	TR-CRF	F	95.89	95.43	95.87	92.68	94.97
		R_{oov}	57.87	58.01	47.07	72.24	58.80
5	TR-CRF+SM	F	96.69	97.04	96.87	96.71	96.82
		R_{oov}	64.22	57.23	50.42	71.34	60.80
6	BT-CRF	F	97.04	97.51	96.96	97.75	97.32
		R_{oov}	68.78	75.39	73.94	86.48	76.15
7	BT-CRF+SM(CROSSWISE)	F	**98.46**	**98.04**	**98.42**	**98.73**	**98.41**
		R_{oov}	**83.88**	**81.86**	**77.25**	86.50	**82.37**

Table 4: Experimental results of the proposed model on the tests of four CWS datasets with different configurations, "+SM" indicates this model uses the Switch-memory module. There are two blocks. The first block is results of the baseline model (BERT - CRF) on the single-era dataset. The second block consists of the results of cross-era learning model with different encoders ("BL" for Bi-LSTM, "TR" for Transformer, "BT" for BERT). Here, F, R_{oov} represent the F1 value and OOV recall rate respectively. The maximum F1 values are highlighted for each dataset.

Models	AWIKI		PWIKI		DWIKI		MSR	
	F	R_{oov}	F	R_{oov}	F	R_{oov}	F	R_{oov}
Chen et al. (2017)	-	-	-	-	-	-	96.04	71.60
Gong et al. (2019)	-	-	-	-	-	-	97.78	64.20
Luo et al. (2019b)	91.25	56.32	97.01	48.09	97.00	43.18	97.09	75.19
Ye et al. (2019b)	-	-	-	-	-	-	98.40	84.87
Qiu et al. (2020)	96.44	65.06	95.83	63.75	96.31	57.03	98.05	78.92
Huang et al. (2020a)	98.16	78.97	97.70	75.69	98.12	74.28	98.29	81.75
Tian et al. (2020c)	-	-	-	-	-	-	98.28	**86.67**
CROSSWISE	**98.46**	**83.88**	**98.04**	**81.86**	**98.42**	**77.25**	**98.73**	86.50

Table 5: Performance (F1 value) comparison between CROSSWISE and previous state-of-the-art models on the test sets of four datasets.

three ancient Chinese datasets.

The second ablation study is to evaluate the effect of the switcher. For this experiment, we use the average of four memory cells embedding as the final memory representation. The comparison between the second and the third line indicates that the switcher is an important component when integrating era-specific information.

In summary, in terms of average performance, the switcher and memory cells both can boost the performance on OOV considerably.

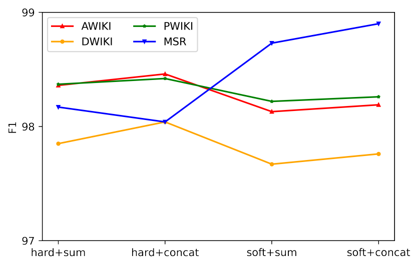


Figure 2: The F1 values of CROSSWISE using four pair settings, "hard+sum" means hard-switcher and sum the memory embedding and the character embedding from encoder as the final character representation.

ID	Switcher	Memory	AWIKI		DWIKI		PWIKI		MSR	
			F	R_{oov}	F	R_{oov}	F	R_{oov}	F	R_{oov}
1	✓	×	98.00	80.62	97.87	80.69	97.52	74.69	98.01	86.48
2	×	✓	98.28	76.58	97.85	74.80	98.32	74.85	98.63	**86.85**
3	✓	✓	**98.46**	**83.88**	**98.04**	**81.86**	**98.42**	**77.25**	**98.73**	86.50

Table 6: Ablation experiments.

Sample from MSR (Modern Chinese): 从大乱走向大治，中经雍正承前启后。 (From chaos to prosperity, through Yongzheng connects the past and the future.)													
Golds	从	大	乱	走	向	大	治	，	中	经	雍正	承前启后	。
	from	big	chaos	go	to	big	prosperity	，	middle	through	Yongzheng	connect	。
w/o SM	从	大	乱	走	向	大	治	，	中经		雍正	承前启后	。
Ours	从	大	乱	走	向	大	治	，	中	经	雍正	承前启后	。
Mixed sample from DWIKI (Near Ancient Chinese): 古人诗中有“水流花谢两无情”。 (In ancient poems, there are “two merciless things: water flowing and flowers fading.)													
Golds	古	人	诗	中	有	“	水	流	花	谢	两	无情	”。
	ancient	people	poem	in	have	“	water	flow	flower	fade	two	merciless	”。
w/o SM	古	人	诗	中	有	“	水	流	花	谢	两	无 | 情	”。
Ours	古	人	诗	中	有	“	水	流	花	谢	两	无情	”。

Table 7: Segmentation cases from the test sets of MSR and DWIKI datasets.

4.5 Mode selection

In this section, we investigate the influence of the switcher mode and the combination mode (concatenate or sum) of the memory embedding and the character embedding.

To better understand the effect of the different configurations. We study four pair settings to train our model on four datasets, the results as shown in Figure 2, where different color poly lines represent different dataset. As we see, *soft-switcher* significantly improves the F1 value on MSR comparing to *hard-switcher*, while other three datasets perfer *hard-switcher*, which implies that the forward direction of knowledge dissemination from ancient to modern can help modern Chinese word segmentation, and the reverse knowledge dissemination will have a negative impact on ancient Chinese word segmentation. Concatenating the memory embedding and the character embedding from the encoder outperforms summing both.

4.6 Case study

We further explore the benefits of the SM mechanism by comparing some cases from BERT-CRF and CROSSWISE. Table 7 lists two examples from the test sets of MSR and DWIKI datasets. According to the results, in the first sentence, BERT-CRF gives the wrong prediction of boundary in “中(middle)” and “经(through)”. However, our CROSSWISE achieves exact segmentation of this instance. The second sample is a sentence written in both ancient and modern Chinese, we could observe that CROSSWISE also can split the words correctly. This investigation indicates that our model is flexible for era-hybrid texts Chinese word segmentation, and can produce the different segmentation granularity of words according to the era of the sentence. At the same time, it also shows that our model is effective to integrate the era-specific linguistic knowledge according to different samples.

5 Conclusion

In this paper, we propose a flexible model, called CROSSWISE, for cross-era Chinese word segmentation, which can improve the performance of every single dataset by fully integrating the era-specific knowledge. Experiments on four corpora show the effectiveness of our model. In the future, we are also planning to incorporate other labeling tasks into the CROSSWISE, such as POS tagging and named entity recognition.

References

Andrea Ceroni, Nam Khanh Tran, Nattiya Kanhabua, and Claudia Niederée. 2014. Bridging temporal context gaps using time-aware re-contextualization. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, page 1127–1130. ACM.

Ernie Chang, Yow-Ting Shiue, Hui-Syuan Yeh, and Vera Demberg. 2021. Time-aware Ancient Chinese text translation and inference. In *Proceedings of the 2nd International Workshop on Computational Approaches to Historical Language Change 2021*, pages 1–6, Online. 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)*, page 1193–1203. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

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

Jinlan Fu, Pengfei Liu, Qi Zhang, and Xuanjing Huang. 2020. Rethink cws: Is chinese word segmentation a solved task? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5676–5686, Online. Association for Computational Linguistics.

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.

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

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

Shuhei Kurita, Daisuke Kawahara, and Sadao Kurohashi. 2017. Neural joint model for transition-based chinese syntactic analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 1204–1214. 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.

Junxin Liu, Fangzhao Wu, Chuhan Wu, Yongfeng Huang, and Xing Xie. 2018. Neural chinese word segmentation with dictionary knowledge. In *Natural Language Processing and Chinese Computing*, pages 80–91, Cham. Springer International Publishing.

Yang Liu, Yuanhe Tian, Tsung-Hui Chang, Song Wu, Xiang Wan, and Yan Song. 2021. Exploring word segmentation and medical concept recognition for Chinese medical texts. In *Proceedings of the 20th Workshop on Biomedical Language Processing*, pages 213–220, Online. Association for Computational Linguistics.

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

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

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston.

2016. Key-value memory networks for directly reading documents. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1400–1409, Austin, Texas. 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*, page 2887–2897. Association for Computational Linguistics.

Yan Shao, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2017. Character-based joint segmentation and pos tagging for chinese using bidirectional rnn-crf.

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

Yuanhe Tian, Yan Song, and Fei Xia. 2020b. Joint Chinese word segmentation and part-of-speech tagging via multi-channel attention of character n-grams. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 2073–2084, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yuanhe Tian, Yan Song, Fei Xia, Tong Zhang, and Yonggang Wang. 2020c. 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.

Jingjing Xu and Xu Sun. 2016. Dependency-based gated recursive neural network for Chinese word segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–572, Berlin, Germany. Association for Computational Linguistics.

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, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 839–849. Association for Computational Linguistics.

Yuxiao Ye, Weikang Li, Yue Zhang, Likun Qiu, and Jian Sun. 2019a. Improving cross-domain Chinese word segmentation with word embeddings. 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 2726–2735, Minneapolis, Minnesota. Association for Computational Linguistics.

Yuxiao Ye, Weikang Li, Yue Zhang, Likun Qiu, and Jian Sun. 2019b. Improving cross-domain Chinese word segmentation with word embeddings. 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 2726–2735, Minneapolis, Minnesota. Association for Computational Linguistics.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Transition-based neural word segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 421–431, Berlin, Germany. Association for Computational Linguistics.

Qi Zhang, Xiaoyu Liu, and Jinlan Fu. 2018. Neural networks incorporating dictionaries for chinese word segmentation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5682–5689. AAAI Press.

Xiaoqing Zheng, Hanyang Chen, and Tianyu Xu. 2013. Deep learning for Chinese word segmentation and POS tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 647–657, Seattle, Washington, USA. Association for Computational Linguistics.

A Appendix

A.1 Extra Case Study

We further explore the benefits of the SM mechanism by comparing some cases from BERT-CRF and CROSSWISE. Table 8 lists three examples from the test sets of Ancient Chinese, modern Chienae, and Near Ancient Chinese datasets. According to the results, in the first sentence, “靡(swept)” and “草(grass)” are two words in ancient Chinese, BERT-CRF treats these two words as a single word; BERT-CRF gives the second sentence the wrong prediction of boundary in “中(middle)” and “经(through)”. However, our CROSSWISE achieves all exact segmentation of these instances. The third sample is a sentence written in both ancient and modern Chinese, we could observe that CROSSWISE also can split the words correctly. This investigation indicates that our model is flexible for era-hybrid texts Chinese word segmentation, and can produce the different segmentation granularity of words according to the era of the sentence. At the same time, it also shows that the SM mechanism is effective to integrate the era-specific linguistic knowledge according to different samples.

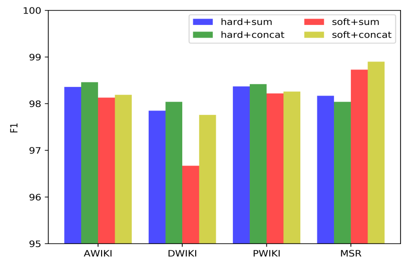
Sample from AWIKI (Ancient Chinese): 故上化下，犹风之靡草也。 (Therefore, the superior civilizes and the subordinate, like the winds swept the grass)													
Golds	故	上	之	化	下	，	犹	风	之	靡	草	也	。
	so	superior	zhi	enlighten	subordinate	,	like	wind	zhi	swept	grass	ye	。
w/o SM	故	上	之	化	下	，	犹	风	之	靡草		也	。
Ours	故	上	之	化	下	，	犹	风	之	靡	草	也	。
Sample from MSR (Modern Chinese): 从大乱走向大治，中经雍正承前启后。 (From chaos to prosperity, through Yongzheng connects the past and the future.)													
Golds	从	大	乱	走	向	大	治	，	中	经	雍正	承前启后	。
	from	big	chaos	go	to	big	prosperity	,	middle	through	Yongzheng	connect	。
w/o SM	从	大	乱	走	向	大	治	，	中经		雍正	承前启后	。
Ours	从	大	乱	走	向	大	治	，	中	经	雍正	承前启后	。
Mixed sample from DWIKI (Near Ancient Chinese): 古人诗中有“水流花谢两无情”。 (In ancient poems, there are “two merciless things: water flowing and flowers fading.)													
Golds	古	人	诗	中	有	“	水	流	花	谢	两	无情	”。
	ancient	people	poem	in	have	“	water	flow	flower	fade	two	merciless	”。
w/o SM	古	人	诗	中	有	“	水	流	花	谢	两	无 情	”。
Ours	古	人	诗	中	有	“	水	流	花	谢	两	无情	”。

Table 8: Segmentation cases from the test sets of MSR, AWKI and DWIKI datasets.

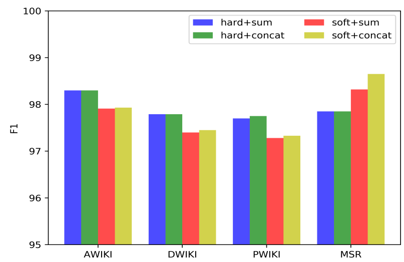
A.2 Effect on Dataset Imbalance

In this section, we investigate the influence of the switcher mode and the combination mode. Our model is a multi-task framework, imbalanced datasets will bring some sentence classification errors, we expect to use different switcher modes to minimize the negative effect of these errors.

We study four pair settings to train our model on four intact datasets, the results as shown in Figure 3(a). According to Table 3, the data of the four datasets are unbalanced. In order to explore the relationship between the data balance and experiment configurations. We randomly keep 50K training samples for MSR and PKIWI in the training set respectively, then conduct experiments with different settings. The experimental results as shown in Figure 3(b). Although less than half of the training data has been reduced, MSR is still sensitive to the "soft-concat" setting and keeps a competitive F1 value. The results of the other three datasets drop slightly. Moreover, the comparison between Figure 3(a) and Figure 3(b) indicates that although the data are imbalanced, hybrid training is also a strategy to increase the scale of training samples in disguise. As we know, the scale of training samples is the key to improve the performance with neural methods.



(a) The F1 values of SMSeg using four pair settings on four datasets, the data of the four datasets are unbalanced.



(b) The F1 values of SMSeg using four pair settings on four datasets, the data of the four datasets are balanced. MSR and PKIWI only keep about 50K training samples.

Figure 3: The F1 values of SMSeg using four pair settings, "hard+sum" means hard-switcher and sum the memory embedding and the character embedding from encoder as the final character representation.