Delving Deep into Regularity: A Simple but Effective Method for Chinese Named Entity Recognition

Yingjie Gu[1], **Xiaoye Qu**[1]*, **Zhefeng Wang**[1]†, **Yi Zheng**[1]
Baoxing Huai[1], **Nicholas Jing Yuan**[1]
[1]Huawei Cloud, China
{guyingjie4,quxiaoye,wangzhefeng,zhengyi29}@huawei.com
huaibaoxing@huawei.com, nicholas.jing.yuan@gmail.com

Abstract

Recent years have witnessed the improving performance of Chinese Named Entity Recognition (NER) from proposing new frameworks or incorporating word lexicons. However, the inner composition of entity mentions in character-level Chinese NER has been rarely studied. Actually, most mentions of regular types have strong name regularity. For example, entities end with indicator words such as "公司 (company) " or "银行 (bank)" usually belong to organization. In this paper, we propose a simple but effective method for investigating the regularity of entity spans in Chinese NER, dubbed as **R**egularity-**I**nspired re**CO**gnition **N**etwork (RICON). Specifically, the proposed model consists of two branches: a regularity-aware module and a regularity-agnostic module. The regularity-aware module captures the internal regularity of each span for better entity type prediction, while the regularity-agnostic module is employed to locate the boundary of entities and relieve the excessive attention to span regularity. An orthogonality space is further constructed to encourage two modules to extract different aspects of regularity features. To verify the effectiveness of our method, we conduct extensive experiments on three benchmark datasets and a practical medical dataset. The experimental results show that our RICON significantly outperforms previous state-of-the-art methods, including various lexicon-based methods.

1 Introduction

Named entity recognition (NER) aims at identifying text spans pertaining to specific entity types. It plays an important role in many downstream tasks such as relation extraction (Cheng et al., 2021), entity linking (Gu et al., 2021), co-reference resolution (Clark and Manning, 2016), and knowledge graph (Ji et al., 2020). Due to the complex composition (Gui et al., 2019), character-level Chinese NER is more challenging compared to English NER. As shown in Figure 1 (a), the middle character "流" can constitute words with the characters to both their left and their right, such as "河流 (River)" and "流经 (flows)", leading to ambiguous character boundaries.

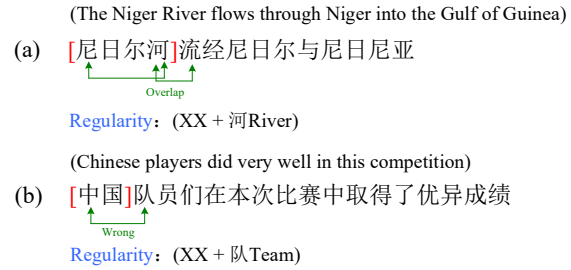


Figure 1: (a) Complex composition of Chinese NER and regularity. (b) Excessive focusing on regularity leads to wrong entity boundary.

There are two typical frameworks for NER. The first one conceptualizes NER as a sequence labeling task (Huang et al., 2015; Lample et al., 2016; Ma and Hovy, 2016), where each character is assigned to a special label (e.g., B-LOC, I-LOC). The second one is span-based method (Li et al., 2020a; Yu et al., 2020), which classifies candidate spans based on their span-level representations. However, despite the success of these two types of methods, they do not explicitly take the complex composition of Chinese NER into consideration. Recently, several works (Zhang and Yang, 2018; Gui et al., 2019; Li et al., 2020b) utilize external lexicon knowledge to help connect related characters and promote capturing the local composition. Nevertheless, building the lexicon is time-consuming and the quality of the lexicon may not be satisfied.

In contrast to previous works, we observe that the regularity exists in the common NER types (e.g., ORG and LOC). As shown in Figure 1 (a), "尼日尔河 (Niger River)" follows the specific composition pattern "XX+河 (XX + River)" which ends

*Equal contribution
†Corresponding author

Findings of the Association for Computational Linguistics: NAACL 2022, pages 1863 - 1873

with indicator character “河" and mostly belongs to location type, and the ambiguous character “流” can properly constitute “流经” with the right character “经”. Thus, the regularity information serves as important clues for entity type recognition and identifying the character composition. Formally, we refer to regularity as specific internal patterns contained in a type of entity (Lin et al., 2020). However, too immersed regularity leads to unfavorable boundary detection of entities and disturbing character composition. As shown in Figure 1 (b), “中国队 (Chinese team)” conforms to the pattern “XX+队 (XX + Team)”, but the correct entity boundary should be “中国 (Chinese)” and “队员 (players)” according to the context. Therefore, the context also plays a key role in determining the character boundary.

In this paper, we introduce a simple but effective method to explore the regularity information of entity spans for Chinese NER, dubbed as **R**egularity-**I**nspired re**CO**gnition **N**etwork (RICON). The proposed model consists of two branches named regularity-aware module and regularity-agnostic module, where each module has task-specific encoder and optimization object. Concretely, the regularity-aware module aims at analyzing the internal regularity of each span and integrates the significant regularity information into the corresponding span-level representation, leading to precise entity type prediction. Meanwhile, the regularity-agnostic module is devised to capture context information and avoid excessive focus on intra-span regularity. Furthermore, we adopt an orthogonality space restriction to encourage two branches to extract different features with regard to the regularity. To verify the effectiveness of our method, we conduct extensive experiments on three large-scale benchmark datasets (OntoNotes V4.0, OntoNotes V5.0, and MSRA). The results show that RICON achieves considerable improvements compared to the state-of-the-art models, even outperforming existing lexicon-based models. Moreover, we experiment on a practical medical dataset (CBLUE) to further demonstrate the ability of RICON.

Our contributions can be summarized as follows:

- This is the first work that explicitly explores the internal regularity of entity mentions for Chinese NER.
- We propose a simple but effective method for Chinese NER, which effectively utilizes regularity information while avoiding excessive focus on intra-span regularity.
- Extensive experiments on three large-scale benchmark datasets and a practical medical dataset demonstrate the effectiveness of our proposed method.

2 Related Work

Traditional methods treat NER as a sequence labeling task, where each word or character in the sentence is assigned to a special label. As a representative, Huang et al. (2015) utilized the BiLSTM as an encoder to learn the contextual representation, and then exploited Conditional Random Field (CRF) as a decoder to label the tokens. The BiLSTM-CRF architecture achieved superior performance on various datasets, hence many following works (Lample et al., 2016; Ma and Hovy, 2016) adopt such architecture. More recently, strong pre-trained language models such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) are incorporated to further enhance the performance of NER. Although the sequence labeling framework achieves decent performance on flat NER, it struggles for nested NER. As a result, span-based models are proposed to solve the nested problem by classifying all possible spans into predefined types (e.g. PER, LOC) in the sentence. For example, Yu et al. (2020) adopted a biaffine attention model to assign scores for all potential spans and achieved the state-of-the-art performance on both flat and nested English NER datasets. Shen et al. (2021) also employed span-based framework on Chinese NER datasets. In this paper, we adopt span-based method as our basic framework for two reasons. Firstly, the span-based method considers each span and naturally suits analyzing inner-span character composition. Secondly, the span-based framework can easily extend our method from flat NER to nested NER.

Recently, for Chinese NER, researchers proposed various lexicon-based models that incorporate the external lexicon information and obtained better results. Zhang and Yang (2018) investigated Lattice-LSTM for incorporating word lexicons into the character-based NER model. However, the lattice structure fails to compute in parallel. To address this problem, Gui et al. (2019) introduced a lexicon-based graph neural network that recasts Chinese NER as a node classification task. There are also several works that focus on incorporating all matched words from the lexicon into the charac-

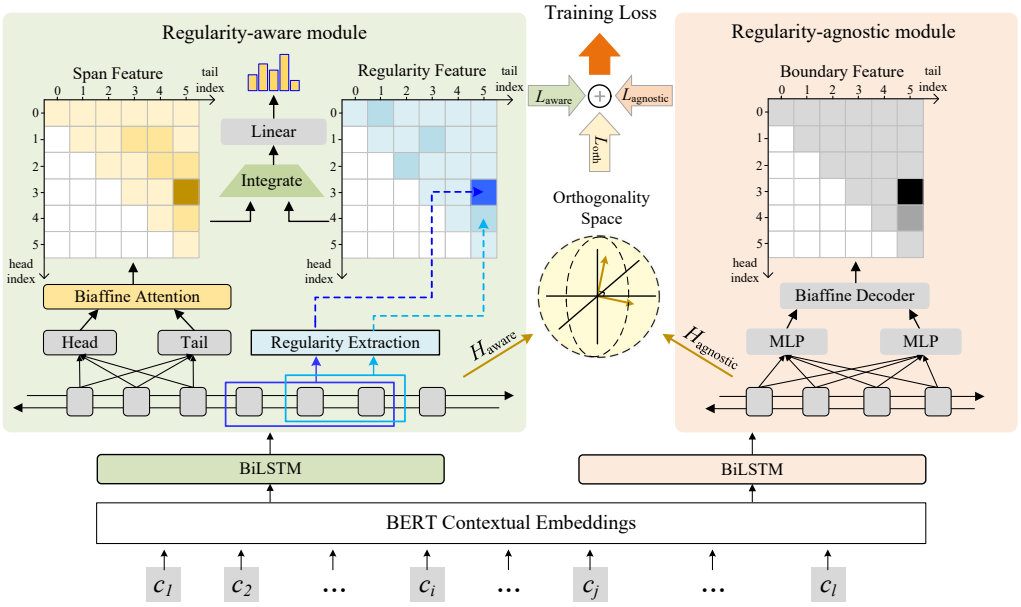


Figure 2: Overall structure of RICON. Each character in the sentence is first embedded by BERT. Then, two separate Bi-LSTM layers are adopted to encode representations for the regularity-aware module and regularity-agnostic module. An orthogonality space is further utilized to encourage extracting different features for each module.

ter embeddings (Ma et al., 2020; Liu et al., 2021). Different from the aforementioned lexicon-based works that incorporate external resources, in this paper, we focus on exploring the internal regularity information of spans.

3 Method

The overall architecture of our RICON is shown in Figure 2, which mainly consists of two branches: the regularity-aware module and the regularity-agnostic module.

3.1 Embedding and Task-specific Encoder

First of all, each character of the input sequence is embedded into a dense vector. Then the character vectors are separately fed into two task-specific bidirectional LSTM (BiLSTM) layers to extract the corresponding hidden states for each module respectively. Formally, given a sentence with l characters $s = \{c_1, c_2, ..., c_l\}$. We use a standard BERT (Devlin et al., 2019) to obtain the context dependent embeddings for a target token:

$$x_i = \text{BERT}(c_i) \tag{1}$$

Then, the sequence of character embeddings will be fed to two separate BiLSTM layers for regularity-aware module and regularity-agnostic module. The hidden state of BiLSTM is expressed as follows:

$$\overrightarrow{h}_{i,\tau} = \overrightarrow{\text{LSTM}}(x_i, \overrightarrow{h}_{i-1,\tau}) \tag{2}$$

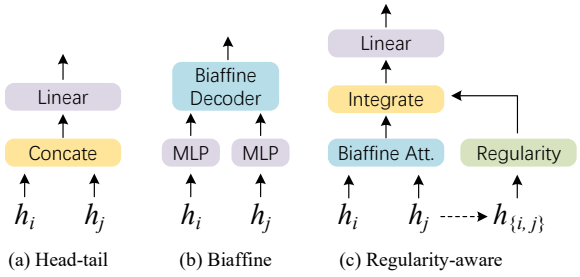


Figure 3: Conceptual comparison of three architectures for span-based NER. $\{,\}$ denotes the representations across from ith to jth character of the span $s_{i,j}$.

$$\overleftarrow{h}_{i,\tau} = \overleftarrow{\text{LSTM}}(x_i, \overleftarrow{h}_{i-1,\tau}) \tag{3}$$

$$h_{i,\tau} = [\overrightarrow{h}_{i,\tau}; \overleftarrow{h}_{i,\tau}] \tag{4}$$

where $\tau \in \{\text{aware}, \text{agnostic}\}$, $[;]$ denotes concatenation, and the dimension of $h_{i,\tau}$ is $2d$. The character sequence representation can be denoted as $H_\tau = \{h_{1,\tau}, ..., h_{i,\tau}, ..., h_{l,\tau}\}$.

3.2 Regularity-aware Module

In this module, we aim to explore the internal regularity of each span. As shown in Figure 3 (a), typical span-based NER methods (Sohrab and Miwa, 2018; Xia et al., 2019; Li et al., 2020a) represent each entity span via concatenating corresponding head and tail features, and use a linear classifier

to predict the type of this span. In this way, the span features are coarse-grained. Then, as denoted in Figure 3 (b), Yu et al. (2020) propose a biaffine decoder to enhance the interaction between head and tail representations after two MLPs and predict span types simultaneously. Nevertheless, the internal regularity among characters in the span is still neglected in this biaffine method.

Consequently for this, our regularity-aware module is devised to capture the internal regularity feature for each span $s_{i,j}$, as demonstrated in Figure 3 (c). It is worth noting that span representations are obtained by the head and tail characters of the span, while the regularity representations stem from each character in the span. To achieve this goal, we utilize a linear attention to obtain the regularity representation of each span as follows:

$$a_t = W_{\text{reg}}^{\top} h_t + b_{\text{reg}} \tag{5}$$

$$\alpha_t = \frac{exp(a_t)}{\sum_{k=i}^{j} exp(a_k)} \tag{6}$$

$$h_{s_{i,j}}^{(\text{reg})} = \sum_{t=i}^{j} \alpha_t \cdot h_t \tag{7}$$

where $h_t = h_{t,\text{aware}}$ and $t \in \{i, i+1, ..., j\}$ is the index of the span, $W_{\text{reg}} \in \mathbb{R}^{2d \times 1}$ and $b_{\text{reg}} \in \mathbb{R}^1$ are learnable weights and bias respectively. For a span whose length is 1, we do not extract extra features but use the hidden representation $h_{i,\text{aware}}$ to denote its regularity. The regularity feature $H^{(\text{reg})} \in \mathbb{R}^{l \times l \times 2d}$ will be used for the subsequent entity type prediction.

To predict the type of an entity, our model integrates the regularity feature of each span into the span representations. Firstly, we acquire the span representation via a biaffine attention mechanism by interacting head and tail features:

$$h_{s_{i,j}}^{(\text{span})} = h_i^{\top} U^{(1)} h_j + (h_i \oplus h_j) U^{(2)} + b_1 \tag{8}$$

where $h_i, h_j \in H_{\text{aware}}$ are the head and tail representations of span $s_{i,j}$. $U^{(1)}$ is a $2d \times 2d \times 2d$ tensor, $U^{(2)}$ is a $4d \times 2d$ matrix, and b_1 is the bias. It is worth noting that here we do not apply two separate MLPs like Figure 3 (b) to generate different representations for the head and tail features of the spans, as different MLPs will project the head, tail, and regularity representation into distinct spaces. The experiment also verifies that such space inconsistency degrades the recognition performance. Then a gated network is devised to integrate the span and regularity representation as below:

$$g_{s_{i,j}} = \sigma(U^{(3)}[h_{s_{i,j}}^{(\text{span})}; h_{s_{i,j}}^{(\text{reg})}] + b_2) \tag{9}$$

$$h_{s_{i,j}} = g_{s_{i,j}} \odot h_{s_{i,j}}^{(\text{span})} + (1 - g_{s_{i,j}}) \odot h_{s_{i,j}}^{(\text{reg})} \tag{10}$$

where $U^{(3)} \in \mathbb{R}^{4d \times 1}$ is a trainable parameter and b_2 is the bias. σ denotes the sigmoid function and $\odot$ mean the element-wise dot multiplication. Finally, we adopt a standard linear classifier with a softmax function to predict the entity type for each span.

$$\tilde{y}_{s_{i,j}} = \text{Softmax}({W_{\text{type}}}^{\top} h_{s_{i,j}} + b_3) \tag{11}$$

where $W_{\text{type}} \in \mathbb{R}^{2d \times c}$ is a trainable parameter and b_3 is the bias. The loss function of the regularity-aware module is defined as cross-entropy:

$$\mathcal{L}_{\text{aware}} = -\frac{1}{N} \sum_{n=1}^{N} \sum_{i=1}^{l} \sum_{j=1}^{l} y_{s_{i,j}}^{(n)} \log(\tilde{y}_{s_{i,j}}^{(n)}), i \leq j \tag{12}$$

where $\tilde{y}_{s_{i,j}}$ denotes the prediction and $y_{s_{i,j}}$ is the the ground truth type of the span. N is the number of training samples in the regularity-aware module.

3.3 Regularity-agnostic Module

By considering regularity, above regularity-aware module makes the model stricter in terms of predicting the entity type, thus improving the precision of entity prediction. Nevertheless, too immersed regularity may result in inaccurate word boundaries. To get rid of it, we propose to erase the concrete form of golden entities and relieve the excessive learning of structural pattern by regularity-aware module. In this scenario, the head and tail features which determine boundary become more significant, thereby we first apply two multi-layer perceptrons (MLPs) on the hidden states from BiLSTM to get separate representations for head and tail. Then a biaffine decoder is leveraged for obtaining entity probability of the span $s_{i,j}$ as follows:

$$\bar{h}_i = \text{MLP}_{\text{head}}(h_i) \quad \bar{h}_j = \text{MLP}_{\text{tail}}(h_j) \tag{13}$$

$$\bar{y}_{ij} = \sigma([\bar{h}_i; 1]^{\top} U_m [\bar{h}_j; 1]) \tag{14}$$

where $h_i = h_{i,\text{agnostic}}$, $h_j = h_{j,\text{agnostic}}$, U_m is a $(2d+1) \times 1 \times (2d+1)$ trainable parameter, σ

is the sigmoid function. Finally, we adopt binary cross-entropy loss to train this task.

$$\mathcal{L}_{\text{agnostic}} = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{l}\sum_{j=1}^{l}[y_{ij}^{(n)}\log(\bar{y}_{ij}^{(n)}) + (1-y_{ij}^{(n)})\log(1-\bar{y}_{ij}^{(n)})], i \leq j \tag{15}$$

where $\bar{y}_{ij}$ denotes the prediction and y_{ij} is the binary target indicating whether the span is an entity or not. N is the number of training samples in the regularity-agnostic module.

3.4 Orthogonality Space Restriction

As regularity-aware module aims to capture the regularity information while regularity-agnostic module pays no attention to the concrete regularity, we expect to learn different features for these two modules. To this end, we construct an orthogonality space on the top of two BiLSTM layers to encourage encoding different aspects of the input embeddings. The loss is calculated as follows:

$$H_{\text{orth}} = {H_{\text{aware}}}^{\top} H_{\text{agnostic}} \tag{16}$$

$$\mathcal{L}_{\text{orth}} = \|H_{\text{orth}}\|_F^2 = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i=1}^{l}\sum_{j=1}^{l}|h_{ij}^{(n)}|^2 \tag{17}$$

where $\|\cdot\|_F^2$ is the squared Frobenius norm and N is the number of training elements.

3.5 Training and Inference

During training, our RICON can be trained by joint optimizing above three sub-tasks, so we define the total loss as below:

$$\mathcal{L} = \lambda_1\mathcal{L}_{\text{aware}} + \lambda_2\mathcal{L}_{\text{agnostic}} + \lambda_3\mathcal{L}_{\text{orth}} \tag{18}$$

where λ_1, λ_2, and λ_3 are hyperparameter. During inference, we directly use regularity-aware module to predict the entity type for each span and apply a post-processing constraint for two overlapped entity candidates E_1 and E_2 that if $E_{1i} < E_{2i} \leq E_{1j} < E_{2j}$, where i and j are start and end indexes, we only select the entity with the higher type score.

4 Experiments

4.1 Datasets

OntoNotes V4.0 (Weischedel et al., 2011). It is a multilingual corpus in the news domain. This dataset has 4 entity types. We use the same split as (Zhang and Yang, 2018).

OntoNotes V5.0 (Pradhan et al., 2013). Compared with V4.0, this version has more news data and contains 18 types of entities. We use the same split as (Jie and Lu, 2019).

MSRA (Levow, 2006). It contains 3 types of named entities collected from the news domain. We use the same split as (Gui et al., 2019).

CBLUE-CMeEE (Hongying et al., 2020). CBLUE is Chinese biomedical language understanding evaluation which consists of 10 sub-tasks. Among them, CMeEE focuses on Chinese medical entity extraction and has 9 types of entities. We use the official train and dev split.

In addition, all types of OntoNotes V4.0, OntoNotes V5.0, MSRA, and 8 types of CBLUE-CMeEE are flat NER, while the symptom type of CBLUE-CMeEE is nested NER.

Due to the space limitation, the statistics of all datasets are listed in the appendix.

4.2 Implementation Details

In our experiments, we use the same settings for all datasets. Specifically, we adopt the standard pre-trained Chinese BERT-base model with 768 dimensions hidden representation to obtain character embeddings. We use Adam optimizer with 2e-5 learning rate for BERT embedding fine-tuning and 0.001 learning rate for other parts. The number of layer and dropout rate of BiLSTM encoders are set to 3 and 0.4. The hidden state size of BiLSTM encoders is set to 200. For the regularity-agnostic module, the output dimension of MLPs and the dropout rate are set to 150 and 0.2. To avoid overfitting, we also apply 0.1 dropout rate for the BERT output embeddings. For the hyper-parameters in loss, we set $\lambda_1 = \lambda_2 = 1$ and $\lambda_3 = 0.5$. For all experiments including ablation study, we adopt an average of performance over five different runs to reduce randomness.

4.3 Comparison Methods

In our experiments, we compare our RICON with recent state-of-the-art methods, where part of them contain pre-trained language model BERT or external Chinese lexicon information. Here we briefly describe five typical methods:

(1) **Star-GAT** (Chen and Kong, 2021) propose a Star-transformer based NER system. They utilize explicit head and tail boundary information and

Models	Lexicon	OntoNotes V4.0			OntoNotes V5.0			MSRA		
		P	R	F1	P	R	F1	P	R	F1
Lattice LSTM (Zhang and Yang, 2018)	✓	76.35	71.56	73.88	-	-	-	93.57	92.79	93.18
Collaborative Graph Network (Sui et al., 2019)	✓	75.06	74.52	74.79	-	-	-	94.01	92.93	93.47
LGN (Gui et al., 2019)	✓	76.13	73.68	74.89	-	-	-	94.19	92.73	93.46
DGLSTM-CRF (Jie and Lu, 2019)		-	-	-	77.40	77.41	77.40	-	-	-
WC-GCN (Tang et al., 2020)	✓	76.59	75.17	75.87	-	-	-	94.82	93.98	94.40
Star-GAT (Chen and Kong, 2021)		79.25	80.66	79.95	78.22	80.88	79.53	-	-	-
with Pre-trained Language Model										
BERT-Tagger		76.01	79.96	77.93	73.59	80.55	76.91	93.40	94.12	93.76
BERT+LSTM+CRF		81.99	81.65	81.82	77.12	79.81	78.44	95.06	94.61	94.83
BERT+PLTE (Mengge et al., 2020)	✓	79.62	81.82	80.60	-	-	-	94.91	94.15	94.53
BERT+Biaffine (Yu et al., 2020)		81.06	84.03	82.52	78.79	80.07	79.43	**96.65**	94.75	95.20
BERT+FLAT (Li et al., 2020b)	✓	-	-	81.82	-	-	-	-	-	96.09
BERT+SoftLexicon (Ma et al., 2020)	✓	**83.41**	82.21	82.81	-	-	-	95.75	95.10	95.42
LEBERT (Liu et al., 2021)	✓	-	-	82.08	-	-	-	-	-	95.70
RICON (Ours)		81.95	**84.78**	**83.33**	**79.26**	**81.64**	**80.43**	95.94	**96.33**	**96.14**

Table 1: We compare our RICON with recent state-of-the-art models on three Chinese benchmark datasets.

Dependency GAT-based implicit boundary information to improve the performance. It is the SOTA model on the OntoNotes V5.0 dataset.

(2) **BERT+Biaffine** (Yu et al., 2020) recast NER as a task of identifying start and end positions and assigning a type to each span by a biaffine attention.

(3) **BERT+FLAT** (Li et al., 2020b) devise a FLAT model for Chinese NER, which converts the lattice structure into a flat structure consisting of spans to overcome the shortage of lattice-based model (Zhang and Yang, 2018). They also equipped with BERT embeddings and achieved the SOTA performance on the MSRA dataset.

(4) **BERT+SoftLexicon** (Ma et al., 2020) incorporate the word lexicon into the character features. They leverage Chinese lexicon to match every character in the sentence with word appeared in the lexicon to improve the performance, which achieves the SOTA performance on OntoNotes V4.0.

(5) **LEBERT** (Liu et al., 2021) introduce a Lexicon Adapter layer to integrate external lexicon knowledge into BERT layers directly.

4.4 Results

We present the results on three benchmark datasets in Table 1. From this table, we can observe that our RICON achieves the state-of-the-art performance on these datasets. Moreover, RICON even outperforms recent methods with Chinese lexicon significantly. Concretely, on OntoNotes V4.0, RICON achieves 0.81 absolute F1 improvement over the strong method BERT+Biaffine and 0.52 absolute improvement compared with the SOTA lexicon-based method BERT+SoftLexicon. On OntoNotes V5.0, we obtain a decent improvement compared to the SOTA approach Star-GAT by 0.90 F1 score. In addition, on MSRA, although the improvement of our model over the SOTA model BERT-FLAT is limited, our model still surpasses the other two lexicon-based models LEBERT and BERT+SoftLexicon by 0.44 and 0.72 respectively.

	All Types			Symptom Type		
Model	P	R	F1	P	R	F1
BERT-Tagger	53.41	63.32	57.95	40.57	45.38	42.84
BERT-CRF	58.34	64.08	61.07	46.01	**47.51**	46.75
BERT-Biaffine	64.17	61.29	62.29	**63.17**	33.91	44.14
RICON	**66.25**	**64.89**	**65.57**	57.93	43.99	**50.01**

Table 2: Performance of models on CBLUE-CMeEE, including all types and symptom type.

In addition, we present the model performance on CBLUE-CMeEE in Table 2. Considering there are no available lexicons for this task, we only compare RICON with typical models. As shown in this table, RICON outperforms the strong BERT-Biaffine model with a 3.28 F1 score improvement over 9 types. It is remarkable progress in this challenging dataset. Meanwhile, we provided the result of nested symptom type. RICON performs much better than BERT-Biaffine with a 5.81 F1 improvement. This observation also denotes that our RICON also applies to nested NER.

4.5 Ablation Study

We conduct abundant ablation studies on OntoNotes V4.0 and V5.0 from module and implementation perspectives in Table 3 and 4. Vanilla in tables is built from RICON by removing orthogonality space and regularity-agnostic module, and omitting to capture regularity features and integrate it in the regularity-aware module.

From the results in Table 3, we can observe that: (1) When applying regularity-agnostic module to the vanilla, the performances improve by

Module	OntoNotes V4.0			OntoNotes V5.0		
	P	R	F1	P	R	F1
Vanilla	81.08	84.17	82.59	77.87	82.04	79.42
+Reg-agnostic	81.18	84.77	82.80	78.32	81.48	79.90
+Reg-aware	82.49	83.86	83.16	79.28	80.90	80.07
+Reg-aware & agnostic	81.72	84.89	83.28	79.24	81.48	80.33
RICON (Ours)	81.95	84.78	83.33	79.26	81.64	80.43

Table 3: Performance of modules on OntoNotes.

Implementation	Dataset (F1)	
	OntoNotes V4.0	OntoNotes V5.0
Vanilla+Reg-aware	83.16	80.07
Reg. feature by Mean-pooling	83.06 (-0.10)	79.97 (-0.10)
Reg. feature by Max-pooling	82.82 (-0.34)	79.79 (-0.28)
Reg. feature by Multi-Head	83.10 (-0.06)	79.86 (-0.21)
Gate replaced with Add	82.96 (-0.20)	79.94 (-0.13)
Gate replaced with Cat	82.80 (-0.36)	79.77 (-0.30)
Apply MLPs to head and tail	82.90 (-0.26)	79.67 (-0.40)
Vanilla	82.59 (-0.57)	79.42 (-0.65)

Table 4: Performance of variants on OntoNotes datasets.

0.21 and 0.48 respectively, showing the effectiveness of this module. (2) When the vanilla equips with regularity-aware module, the F1 scores significantly improve by 0.57 and 0.65 respectively, which verifies that regularity plays a significant role in entity recognition. (3) After combining regularity-aware and regularity-agnostic modules, we achieve further improvements, which indicates that two modules can mutually reinforce each other. (4) The orthogonality space is a valid method according to the further F1 score improvements.

Furthermore, we notice that adding the regularity-aware module significantly increases the **Precision** (1.41 on both datasets, Vanilla vs Vanilla+Reg-aware) but reduces the **Recall** (0.31 and 1.04 respectively), which conforms to that focusing on regularity feature would reinforce the type prediction, while missing several spans that are supposed to be entities. Nevertheless, this situation can be remedied by the regularity-agnostic module and the **Recall** improved 1.03 and 0.58, respectively (Vanilla+Reg-aware vs Vanilla+Reg-aware & agnostic). This result also meets our motivation that regularity-agnostic module can reinforce the entity boundary detection.

As shown in Table 4, there are several alternative ways to extract regularity information instead of linear attention used in this paper, such as mean-pooling, max-pooling, or more complex multi-head self-attention (Vaswani et al., 2017), but these methods all perform worse. It is one future direction to explore how to obtain regularity by a more sophisticated architecture. However, considering the model complexity and performance, we choose linear attention to capture regularity. In addition, replacing our devised gate mechanism with a simple concatenate or add operation both degrades the performance, denoting that gate mechanism is more efficient to integrate span feature and regularity feature. We also explored adding two MLPs separately to head and tail features when generating span features in the regularity-aware module. The experimental results prove that different feature space for span feature and regularity feature leads to worse performance.

4.6 Analysis

In this section, We deeply analyze our proposed RICON from the following aspects.

4.6.1 Regularity: A Latent Adaptive Lexicon.

The lexicon-based methods focus on incorporating external word lexicons to improve the performance of character-based NER. The core concept of them is preserving all words which match a specific character and let the subsequent NER model determine which word to apply (Zhang and Yang, 2018; Ma et al., 2020). In our model, we calculate the regularity for each span, namely, all words containing a specific character are considered, and then the best word and corresponding regularity will be determined. In this sense, our explored regularity can be seen as a latent adaptive lexicon. Furthermore, this latent adaptive lexicon is more complete than external lexicons because all spans matching the specific character are considered, while lexicon-based methods only match a limited number of words. As shown in Table 1, the previous SOTA method BERT+Biaffine performs worse than lexicon-based methods, but our regularity-based method RICON outperforms the lexicon-based methods. Actually, our regularity-based method can further be combined with lexicon-based methods.

4.6.2 Performance vs. Entity Type.

We examine how regularity affects each entity type. As Figure 4 shows, 12 types of entities achieve better performance with the regularity. This result conforms to the fact that types like GPE, ORG, and DATE have strong regularity. Nevertheless, for the types with little regularity information, such as WORK_OF_ART and PERSON, immersed regularity leads to performance degradation. We notice that the MONEY type typically contains regularity but we do not observe an improvement in

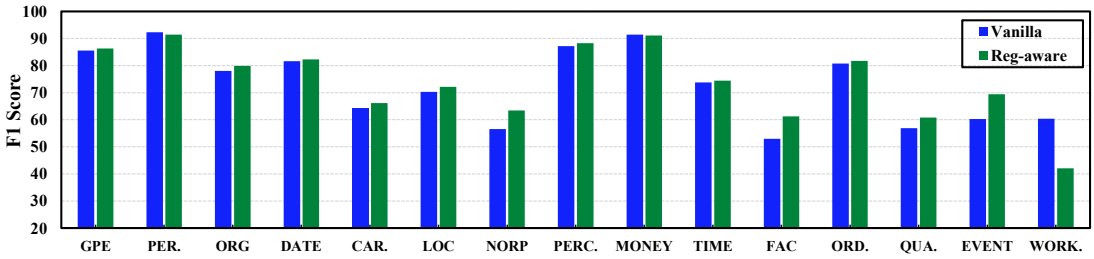
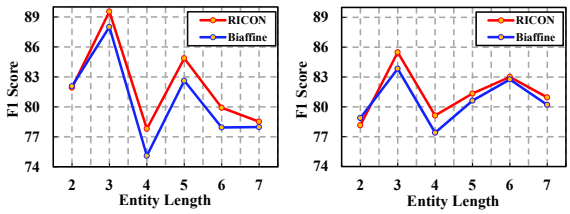


Figure 4: The performance of 15 types of entities on OntoNotes V5.0. The types are sorted in descending order based on the proportion of entities of that type to the total. As the remaining 3 types on OntoNotes V5.0 only have less than 35 entities (0.05%) among all entities. To avoid the impact of labeling errors, we do not present them here.

#1 Sentence (Truncated)	据报道，从波罗的海三国撤回的俄罗斯军队.....			
	(Reportedly, Russian Army withdrawn from the three countries around Baltic Sea...)			
Characters (Entity Included)	波	罗	的	海
Gold Label	B-LOC	M-LOC	M-LOC	E-LOC
Vanilla	B-GPE	M-GPE	M-GPE	M-GPE
Vanilla + Reg-aware	B-LOC	M-LOC	M-LOC	E-LOC
Regularity weight	0.04	0.06	0.07	**0.83**
#2 Sentence (Truncated)	新闻分析：美国公司兼并为何愈演愈烈?			
	(News analysis: why the mergers of American companies are intensifying?)			
Characters (Entity Included)	美	国	公	司
Gold Label	B-GPE	E-GPE	O	O
Vanilla + Reg-aware	B-ORG	M-ORG	M-ORG	M-ORG
Reg-aware + Reg-agnostic	B-GPE	E-GPE	O	O

Table 5: There examples from the Ontonotes V4.0 dataset. The label is organized in the form of BMES.



(a) F1 on OntoNotes V4.0 (b) F1 on OntoNotes V5.0

Figure 5: Performance vs. Entity Length

this category. This is, due to inconsistencies between the training and test dataset. For instance, the training data contains the abundant pattern "number+dollar", while only numbers exist in the test set. To remedy the excessive regularity, our RICON further utilizes a regularity-agnostic module to rectify the captured regularity. The above observations also inspire us to devise more elaborate NER for different entity types with various degree regularity properties in the future. Our regularity-aware module may also serve as a potential tool for evaluating the intensity of regularity.

4.6.3 Performance vs. Entity Length.

Figure 5 depicts the performance on the OntoNotes V4.0 and V5.0 datasets with different length of entities. From this figure, we can observe that our RICON consistently outperforms BERT-Biaffine (Yu et al., 2020) when the entity length is longer than 2, which illustrates that the regularity information is helpful to predict the types for long entities. In contrast, BERT-Biaffine performs comparable to RICON when entity length is 2 as there are no additional character information except the head and tail representations.

4.6.4 Case study.

Table 7 shows two examples from OntoNotes V4.0. In the first example, the Vanilla misidentifies the entity type, while Vanilla+reg-aware learns regularity “XX+海" by the greatest weight 0.83 on “海", thus obtaining the accurate entity type. It is worth noting that regularity can capture more complex character compositions besides explicit patterns in the first example. More complex examples are presented in the appendix. In the second example, “美国公司" conforms to the regularity "XX +公司" and is recognized as organization type by our Vanilla+Reg-aware model. After equipping with the regularity-agnostic module, we obtain the precise character boundary and relieve the excessive attention to regularity.

5 Conclusion

In this paper, we proposed a simple but effective method to explore the regularity information

for Chinese NER, dubbed as Regularity-Inspired reCOgnition Network (RICON). It contains a regularity-aware module to capture the internal regularity feature of each span, and a regularity-agnostic module to reinforce the entity boundary detection while avoid imposing excessive attention on regularity. The features of two modules are encouraged to be dissimilar by an orthogonality space restriction. Evaluation shows that RICON achieves the state-of-the-art performance on four datasets.

References

Chun Chen and Fang Kong. 2021. Enhancing entity boundary detection for better Chinese named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 20–25, Online. Association for Computational Linguistics.

Qiao Cheng, Juntao Liu, Xiaoye Qu, Jin Zhao, Jiaqing Liang, Zhefeng Wang, Baoxing Huai, Nicholas Jing Yuan, and Yanghua Xiao. 2021. Hacred: A large-scale relation extraction dataset toward hard cases in practical applications. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2819–2831.

Kevin Clark and Christopher D. Manning. 2016. Improving coreference resolution by learning entity-level distributed representations. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 643–653, Berlin, Germany. 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, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yingjie Gu, Xiaoye Qu, Zhefeng Wang, Baoxing Huai, Nicholas Jing Yuan, and Xiaolin Gui. 2021. Read, retrospect, select: An mrc framework to short text entity linking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 12920–12928.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuanjing Huang. 2019. A lexicon-based graph neural network for Chinese NER. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1040–1050, Hong Kong, China. Association for Computational Linguistics.

Zan Hongying, Li Wenxin, Zhang Kunli, Ye Yajuan, Chang Baobao, and Sui Zhifang. 2020. Building a pediatric medical corpus: Word segmentation and named entity annotation. In *Workshop on Chinese Lexical Semantics*, pages 652–664.

Z. Huang, X. Wei, and Y. Kai. 2015. Bidirectional lstm-crf models for sequence tagging. *Computer Science arXiv preprint arXiv:1508.01991*.

S. Ji, S. Pan, E. Cambria, P. Marttinen, and P. S. Yu. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *Computer Science arXiv preprint arXiv:2002.00388*.

Z. Jie and W. Lu. 2019. Dependency-guided lstm-crf for named entity recognition. *arXiv preprint arXiv:1909.10148*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

Gina-Anne Levow. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pages 108–117, Sydney, Australia. Association for Computational Linguistics.

J. Li, A. Sun, and Y. Ma. 2020a. Neural named entity boundary detection. *IEEE Transactions on Knowledge and Data Engineering*, PP(99):1–1.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. 2020b. FLAT: Chinese NER using flat-lattice transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6836–6842, Online. Association for Computational Linguistics.

Hongyu Lin, Yaojie Lu, Jialong Tang, Xianpei Han, Le Sun, Zhicheng Wei, and Nicholas Jing Yuan. 2020. A rigorous study on named entity recognition: Can fine-tuning pretrained model lead to the promised land? *arXiv preprint arXiv:2004.12126*.

W. Liu, X. Fu, Y. Zhang, and W. Xiao. 2021. Lexicon enhanced chinese sequence labelling using bert adapter. *arXiv preprint arXiv:2105.07148*.

Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuanjing Huang. 2020. Simplify the usage of lexicon in Chinese NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5951–5960, Online. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Xue Mengge, Bowen Yu, Tingwen Liu, Yue Zhang, Erli Meng, and Bin Wang. 2020. Porous lattice transformer encoder for Chinese NER. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3831–3841, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Sameer Pradhan, Alessandro Moschitti, Nianwen Xue, Hwee Tou Ng, Anders Björkelund, Olga Uryupina, Yuchen Zhang, and Zhi Zhong. 2013. Towards robust linguistic analysis using OntoNotes. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 143–152, Sofia, Bulgaria. Association for Computational Linguistics.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021. Locate and label: A two-stage identifier for nested named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

Mohammad Golam Sohrab and Makoto Miwa. 2018. Deep exhaustive model for nested named entity recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2843–2849, Brussels, Belgium. Association for Computational Linguistics.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 2019. Leverage lexical knowledge for Chinese named entity recognition via collaborative graph network. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3830–3840, Hong Kong, China. Association for Computational Linguistics.

Z. Tang, B. Wan, and L. Yang. 2020. Word-character graph convolution network for chinese named entity recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, PP(99):1–1.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

R. Weischedel, M. Palmer, M. Marcus, E. Hovy, S. Pradhan, L. Ramshaw, N. Xue, A. Taylor, J. Kaufman, and M. Franchini. 2011. Ontonotes release 4.0. *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium.*

Congying Xia, Chenwei Zhang, Tao Yang, Yaliang Li, Nan Du, Xian Wu, Wei Fan, Fenglong Ma, and Philip Yu. 2019. Multi-grained named entity recognition. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1430–1440, Florence, Italy. Association for Computational Linguistics.

Juntao Yu, Bernd Bohnet, and Massimo Poesio. 2020. Named entity recognition as dependency parsing. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6470–6476, Online. Association for Computational Linguistics.

Yue Zhang and Jie Yang. 2018. Chinese NER using lattice LSTM. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1554–1564, Melbourne, Australia. Association for Computational Linguistics.

A Data Statistics

Table 6 shows the detailed statistics of each dataset.

Datasets	Type	Train	Dev	Test
OntoNotes V4.0	Sentence	15.7K	4.3K	4.3K
	Char	491.9K	200.5K	208.1K
	Entity	12.8K	6.5K	7.2K
OntoNotes V5.0	Sentence	36K	6.1K	4.5K
	Char	1197.5K	173.3K	147.4K
	Entity	58.1K	8.5K	7.0K
MSRA	Sentence	46.4K	-	4.4K
	Char	2169.9K	-	172.6K
	Entity	69.7K	-	5.2K
CBLUE-CMeEE	Sentence	15.3K	5.0k	-
	Char	825.0K	270.4K	-
	Entity	62.0K	20.3K	-

Table 6: Statistics of datasets.

B More Case Study

B.1 Complex Regularity

Besides explicit patterns like the first example in Table 5, Table 7 shows a more complex form of regularity that our model can capture. In this example, the Vanilla+Reg-aware model pays highest attention weight 0.92 to important character "和" (and), and recognize that A and B are independent entities according to the regularity "A 和 B (A and B)". For comparison, the vanilla fails to

Sentence (Truncated)	铼德和年兴纺织，是台湾唯二上榜的公司。 (RITEK and Nien Hsing Textiles are the only two companies on the list in Taiwan.)						
Characters (Entity Included)	铼	德	和	年	兴	纺	织
Gold Label	B-ORG	E-ORG	O	B-ORG	M-ORG	M-ORG	E-ORG
Vanilla	O	B-ORG	M-ORG	M-ORG	M-ORG	M-ORG	E-ORG
Vanilla + Reg-aware	B-ORG	E-ORG	O	B-ORG	M-ORG	M-ORG	E-ORG
Regularity weight	7.5e-2	2.5e-6	**9.2e-1**	2.1e-6	8.9e-6	9.5e-4	3.1e-5

Table 7: An example on the Ontonotes V4.0 dataset. The label is organized in the form of BMES.

Sentence (Truncated)	Golden Entity / Type	Biaffine Prediction	RICON Prediction	Regularity
(1) 肺多叶病变显示...	肺多叶病变，Symptom	未识别	肺多叶病变，Symptom	XX+病变
Multi-lobed lung lesions showed that...	Multi-lobed lung lesions	N/A	Multi-lobed lung lesions	
(2) 大片状融合性病变为主...	大片状融合性病变，Symptom	未识别	大片状融合性病变，Symptom	
Massive fusion lesions were the main...	Massive fusion lesion	N/A	Massive fusion lesion	
(3) 增加肝脏病变...多数属...	肝脏病变，Symptom	未识别	肝脏病变，Symptom	XX+lesion
Increase liver lesions, most of which...	liver lesions	N/A	liver lesions	
(4) ...患儿可并发肝损害。	肝损害，Symptom	肝损害，Disease	肝损害，Symptom	XX+损害
...can be complicated with liver damage.	liver damage	liver damage	liver damage	
(5) SARS患儿有部分出现心脏损害...	心脏损害，Symptom	心脏损害，Disease	心脏损害，Symptom	
SARS children suffer from heart damage...	heart damage	heart damage	heart damage	
(6) ...合并多脏器损害	多脏器损害，Symptom	多脏器损害，Disease	多脏器损害，Symptom	XX+damage
...complicated with multi-organ damage	multi-organ damage	multi-organ damage	multi-organ damage	

Table 8: Cases study on the domain CBLUE-CMeEE dataset.

distinguish these two entities. This example further reveals that our regularity-aware module can discover more complex character compositions.

B.2 Case Study in Medical Domain

To further demonstrate the effectiveness of our RICON in Chinese NER, we present six examples of the CBLUE-CMeEE dataset from the medical domain. As shown in the first three examples in Table 8, the biaffine model fails to identify the accurate boundary of the entities, thus leading to unrecognized entity type. However, our RICON achieves detecting the correct span boundary as well as predicting golden type type (Symptom) of the entities according to the regularity "XX+病变" (XX+lesion). In the last three examples, both biaffine model and our RICON successfully detect the correct span boundary of the entities. For entity type prediction, the biaffine model assigns a wrong type (Disease) to these entities, but our RICON predicts types correctly as a result of it captures the regularity feature "XX+损害" (XX+damage) from "Symptom" type. To sum up, our RICON is also beneficial for domain datasets.