LADA-Trans-NER: Adaptive Efficient Transformer for Chinese Named Entity Recognition using Lexicon-Attention and Data-Augmentation

Jiguo Liu[1*], Chao Liu[1,2], Nan Li[1,2*], Shihao Gao[1,2], Mingqi Liu[1], Dali Zhu[1,2]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
{liujiguo, liuchao, linan, gaoshihao, liumingqi, zhudali}@iie.ac.cn

Abstract

Recently, word enhancement has become very popular for Chinese Named Entity Recognition (NER), reducing segmentation errors and increasing the semantic and boundary information of Chinese words. However, these methods tend to ignore the semantic relationship before and after the sentence after integrating lexical information. Therefore, the regularity of word length information has not been fully explored in various word-character fusion methods. In this work, we propose a Lexicon-Attention and Data-Augmentation (LADA) method for Chinese NER. We discuss the challenges of using existing methods in incorporating word information for NER and show how our proposed methods could be leveraged to overcome those challenges. LADA is based on a Transformer Encoder that utilizes lexicon to construct a directed graph and fuses word information through updating the optimal edge of the graph. Specially, we introduce the advanced data augmentation method to obtain the optimal representation for the NER task. Experimental results show that the augmentation done using LADA can considerably boost the performance of our NER system and achieve significantly better results than previous state-of-the-art methods and variant models in the literature on four publicly available NER datasets, namely Resume, MSRA, Weibo, and OntoNotes v4. We also observe better generalization and application to a real-world setting from LADA on multi-source complex entities.

1 Introduction

Named Entity Recognition (NER) plays an essential role in structuring of unstructured text. It is a sequence tagging task that extracts named entities from unstructured text. The main task of NER is to automatically identify named entities such as Person (PER), Location (LOC), Organization (ORG), etc. in given text (Zhao et al. 2020). NER is a basic task of many NLP systems including relation extraction (Takanobu et al. 2019; Wei et al. 2020; Cheng et al. 2021), entity linking (Le and Titov 2018; Hou et al. 2020; Gu et al. 2021), knowledge graph (Ji et al. 2020; Chawla et al. 2021), etc.

Due to the additional word segmentation process of Chinese (Zhao et al. 2019), Chinese NER is more difficult compared to English NER. In particular, Chinese NER has some ambiguity in many cases, and the boundaries of new words are vaguely grasped. Besides, the task also has many other challenges, such as complex combinations, entity nesting, and indefinite length. As shown in Figure 1(a), there is a scenario where a shorter entity is completely contained within another longer entity. For example, "邮件(Mail)" and "邮件钓鱼攻击(Email Phishing Attack)" are two entities: mail and attack event. In Figure 1(b), " 360安全大脑(360 Security Brain)" is easily misidentified as " 大脑(Brain)", and "2020全球高级持续性威胁研究报告(2020 Global Advanced Persistent Threat Research Report)" is a longer entity that is easily misidentified as "研究报告(Research Report)".

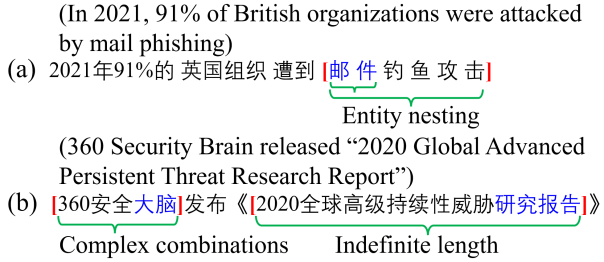

Figure 1: (a) An example to show entity nesting for Chinese NER. (b) An example to show the complex combination and entity indefinite length for Chinese NER.

Traditionally, the task of Chinese NER is decoupled into a pipeline of two separated subtasks, namely word segmentation and word sequence labeling (Yang et al. 2016; Zhao et al. 2021). The major disadvantage of this method is error propagation: word segmentation errors negatively impact the identification of named entities (Peng and Dredze 2015; Sun and He 2017). With the development of deep learning, neural networks have been introduced to the NER task and achieved impressive results (Huang et al. 2015; Lample et al. 2016; Habibi et al. 2017; Gregoric et al. 2018; Lin and Lu 2018). To avoid the segmentation errors, most of neural Chinese NER models are character-based. Although character-based model has achieved good performance than word-based model, it does not exploit word information in character sequence. To explicitly inform each character about its related word information, previous works (Zhang and Yang 2018; Liu et al. 2019; Yan et al. 2019) have proposed to integrate word information into character sequences via word-character lattice structure.

*Corresponding author

As the number of matched words for each character is dynamically changed, such lattice structure is deprived of batch training, which makes the model inefficient and difficult to deploy. Then, the soft-lexicon feature strategy (Peng et al. 2020) is used to overcome the problems of low reasoning efficiency and poor portability of sequence structure. However, this method mainly relies on the weighting of word frequency to embed the word information. The problem is that it ignores the length information of each word and does not fully explore the regularity of the word length in the semantic information before and after the sentence.

To address the above issue, we propose a novel **L**exicon-**A**ttention and **D**ata-**A**ugmentation (LADA) method to integrate word information into character-based model for Chinese NER. The key insight comes from multimodal learning in computer vision (Gao et al. 2019; Yu et al. 2019), where the character and word sequences are viewed as two different modalitie. In order to utilize word information accurately according to sentence semantics, we first design lexicon-attention mechanism to capture the local composition and potential word boundaries by using the lexicon knowledge. We construct a directed graph to search and update edges to obtain optimal lexical information. Then, we further choose to concatenate the representations of the four word sets to represent them as a whole and add it to the character representation. Since entity-labeled data is much smaller than non-entity-labeled data, the available valid data is relatively small. Therefore, we introduce an Adaptive Rank Generative Adversarial Network (AR-GAN) for data augmentation to alleviate the problem of entity label data imbalance.

Finally, we conducted extensive experiments on four public datasets and unique Cyber Threat Intelligence (CTI) datasets to evaluate the proposed model. We find that our framework is quite effective for various NER, which achieves state-of-the-art (SoTA) performances for widely-used benchmark datasets. In particular, we obtain 98.70%, 98.50%, 70.18%, 85.91%, and 95.23% F1 on Resume, MSRA, Weibo, OntoNotes, and CTI datasets respectively.

In summary, our contributions of this paper are summarized as follows:

- We propose a Lexicon-Attention and Data-Augmentation (LADA) framework for Chinese NER, which adds vocabulary information to the character representation layer and effectively integrates word-character information.
- We propose a lexicon attention mechanism that constructs a directed graph with dictionary words. In this work, we make full use of the length information of each word in the dictionary, and effectively realize lexicon enhancement through character graph.
- We introduce an AR-GAN method for data augmentation to alleviate the problem of entity-labeled data imbalance and improve the performance of NER.
- The experimental results show that our method can considerably boost the performance of our NER system, and achieve significantly better results than previous SoTA methods. In particular, we constructed the CTI datasets, and also observed that LADA has better generalization and application on multi-source complex entities.

2 Related Work

Based on the level of granularity, most of the models can be divided into three categories: word-based models, character-based models, and hybrid models.

Word-Based Models Collobert and Weston (Collobert and Weston 2008) proposed one of the first word-based models for NER, with feature constructed from orthographic features, dictionaries and lexicons (Yadav and Bethard 2018). Later, Collobert et al. (Collobert et al. 2011) replaced the hand-crafted features with word embeddings, which improved the automation of NER tasks. The landmark BiLSTM-CRF model (Huang et al. 2015) was proposed and achieved good performance. Ma et al. (Ma and Hovy 2016; Chiu and Nichols 2016) used CNN to capture spelling characteristics, and Lample et al. (Lample et al. 2016) used LSTM instead. The above models all have segmentation errors when applied to Chinese NER, because Chinese word segmentation is compulsory for those models.

Character-Based Models Peng and Dredze (Peng and Dredze 2015) first proposed to add segmentation features for better recognition of entity boundary. Later, Dong et al. (Dong et al. 2016) integrated radical-level features into character-based model. To eliminate the ambiguity of character, Sun and He (Sun and He 2017) took the position of character into account. Although the above models have achieved good results, they all ignore word information in character sequence.

Hybrid Models Some efforts have been made to integrate word boundary information into character-based models. Motivated by the success of multi-task learning for Natural Language Processing, Peng and Dredze (Peng and Dredze 2016) first proposed to jointly train Chinese NER with Chinese word segmentation task. Cao et al. (Cao et al. 2018) applied adversarial transfer learning framework to integrate the task-shared word boundary information into the Chinese NER task. Zhang and Yang (Zhang and Yang 2018) proposed another way to obtain word boundary information, which uses lattice LSTM to integrate word information into character-based model. Gui et al. (Gui et al. 2019a) proposed a CNN-based NER model (LR-CNN) that encoded matched words at different window sizes. In addition, Gui et al. (Gui et al. 2019b) converted lattice into graph and used graph neural networks (GNNs) for encoding. However, NER is very sensitive to sentence structure, and these methods still need to use LSTM as the backbone encoder, which makes the model complex. Later, Yan et al. (Yan et al. 2019) adapted Transformer Encoder to model the character-level features and word-level features by incorporating the direction-aware, distance-aware and un-scaled attention. Zhu et al. (Zhu et al. 2019) proposed a Convolutional Attention Network (CAN) to improve the performance of Chinese NER, which makes the model more efficient and robust. Xue et al. (Xue et al. 2020) proposed Porous Lattice Transformer Encoder (PLTE), which models all characters and matching lexical words in parallel with batch processing.

Recently, Li et al. (Li et al. 2020b) devised a FLAT model for Chinese NER, which converts the lattice structure into a

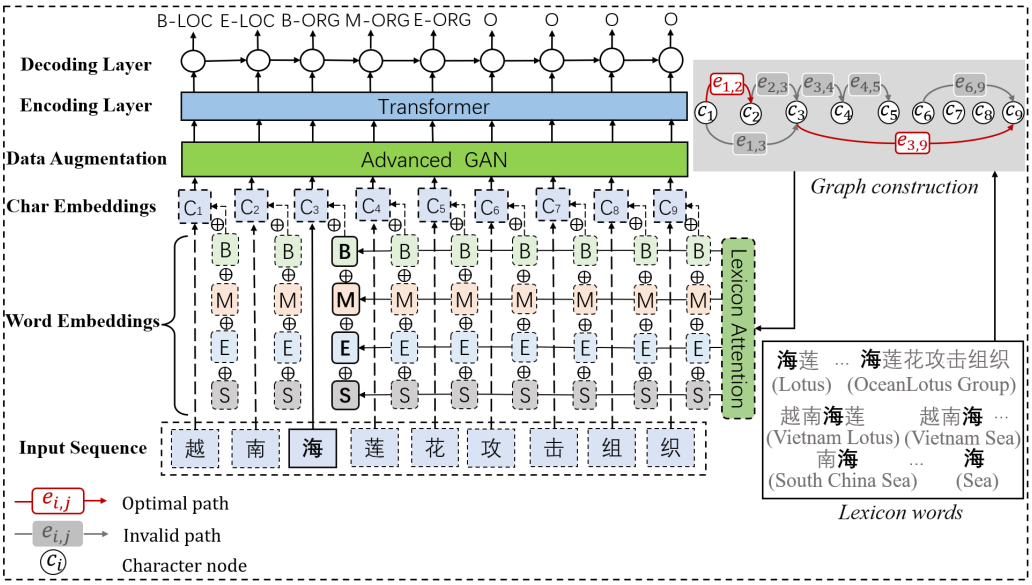

Figure 2: Overview of our overall architecture. Our network mainly consists of five components, *i.e.*, word embeddings, char embeddings, data augmentation, sequence encoding layer, and decoding layer.

flat structure consisting of spans to overcome the shortage of lattice-based model. ZEN 2.0 (Song et al. 2021; Diao et al. 2020) adopt a large volume of data and advanced training technology to integrate n-gram representations. Lattice-BERT (Lai et al. 2021) adopt the multi-granularity structures in lattices to aggregate the coarse-grained word information. Wu et al. (Wu et al. 2021) proposed a novel multi-metadata embedding based cross-transformer (MECT) to improve the performance of Chinese NER by fusing the structural information of Chinese characters. In the latest research, RICON-NER (Gu et al. 2022) adopt a simple but effective method to investigate the regularity of entity spans in Chinese NER, and achieved good performance.

3 Methodology

In this section, we introduce the proposed model for the task of NER (LADA-Trans-NER) in details, and the architecture of our proposed model is shown in Figure 2. We first introduce the character representation layer and incorporate lexicon information. Next, we utilize AR-GAN for data augmentation to alleviate the problem of entity label data imbalance and encode through the sequence Transformer Encoder. Finally, we apply a Conditional Random Field (CRF) (Lafferty, McCallum, and Pereira 2001) layer to perform the decoding for Chinese NER.

3.1 Character Representation Layer

Character embeddings are used to map discrete characters into continuous input vectors. Given a Chinese sentence as $s = \{c_1, c_2, \ldots, c_n\}$, where c_i denotes the i-th character. Each character c_i is represented using a dense vector (embedding):

$$x_i^c = \mathbf{e}^c(c_i), x_i^c \in \mathbb{R}^d, \tag{1}$$

where $\mathbf{e}^c$ denotes the character embedding lookup table[1]. The character feature representations can be formulated as:

$$X = [x_1^c, x_2^c, \ldots, x_n^c], X \in \mathbb{R}^{n \times d}. \tag{2}$$

3.2 Incorporating Lexicon Information

The problem with the purely soft-lexicon NER model (Peng et al. 2020) is that it fails to exploit the regularity of the word length. To address this issue, we proposed a lexical attention mechanism, as described below, to capture the local composition and potential word boundaries by using the lexicon knowledge. In particular, we first introduce the concept of Lexicon-based Character Graphs (LCG) for NER.

Lexicon Attention The whole sentence is converted into a directed graph $g = (\nu, \varepsilon)$, where each character $c_i \in \nu$ is a graph node, and the connection between the first and last characters in a lexicon word can be regarded as an edge ε, as shown in Figure 3. The potential words in the lexicon that match a character subsequence can be formulated as $w_{b,e} = \{c_b, c_{b+1}, \ldots, c_{e-1}, c_e\}$, where the index of the first and last letters are b and e, respectively. Once a character subsequence matches a potential word $w_{b,e}$, we construct one edge $e_{b,e} \in \varepsilon$, pointing from the beginning character c_b to the ending character c_e.

For edge update, we first calculate the in-degree $d^+(c_i)$ and out-degree $d^-(c_i)$ of each node, which can be formulated as:

$$d^+(c_i) = \sum_{j=1}^{i} \mathbb{I}_{\{c_j \to c_i\}} \quad (1 \le j \le i), \tag{3}$$

$$d^-(c_i) = \sum_{k=i}^{n} \mathbb{I}_{\{c_i \to c_k\}} \quad (i \le k \le n). \tag{4}$$

[1]The lookup table is a matrix of embedded vectors for each character in the vocabulary.

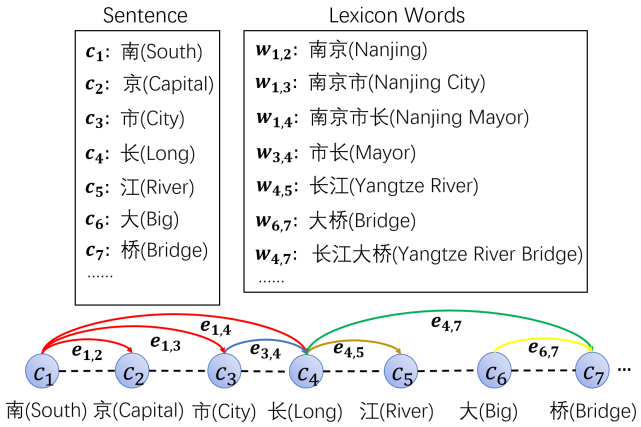

Figure 3: Illustration of lexicon-based character graphs construction. The sentences are composed of characters, which connect to form the vertices. The words in the lexicon form the edges of the graph.

Then the optimal forward path p_λ and the optimal successor path p_γ function can be formulated as:

$$e_{best} \leftarrow p_\lambda = \arg\max_{c_i \in \mathbb{R}^s}(e_{j,i}) \ \ \exists d^+(c_i), \tag{5}$$

$$e_{best} \leftarrow p_\gamma = \arg\max_{c_i \in \mathbb{R}^s}(e_{i,k}) \ \ \exists d^-(c_i), \tag{6}$$

where e_{best} is the optimal path.

The entire training process is described at Algorithm 1 (see Appendix A for details).

Categorizing Matched Words In this work, each character c of a sentence s corresponds to four word sets marked by the four segmentation labels "BMES". For each character c_i in the input sequence, the four set is constructed by:

$$\begin{aligned}
&\mathrm{B}(\mathrm{c_i}) = c_{i,k} \ \ \exists c_{i,k} \in \mathrm{L} : \mathrm{e_{best}} \ (i < k \leq n), \\
&\mathrm{M}(\mathrm{c_i}) = c_{j,k} \ \ \exists c_{j,k} \in \mathrm{L} : \mathrm{e_{best}} \ (1 \leq j < i < k \leq n), \\
&\mathrm{E}(\mathrm{c_i}) = c_{j,i} \ \ \exists c_{j,i} \in \mathrm{L} : \mathrm{e_{best}} \ (1 \leq j < i), \\
&\mathrm{S}(\mathrm{c_i}) = c_i \ \ \exists c_i \in \mathrm{L} : \mathrm{e_{best}},
\end{aligned} \tag{7}$$

where $\mathrm{L} : \mathrm{e_{best}}$ denotes the lexicon used in the optimal path.

The word set $\mathrm{B}(\mathrm{c_i})$ consists of all lexicon matched words on s that begin with c_i. Similarly, $\mathrm{M}(\mathrm{c_i})$ consists of all lexicon matched words in the middle of which c_i occurs, $\mathrm{E}(\mathrm{c_i})$ consists of all lexicon matched words that end with c_i, and $\mathrm{S}(\mathrm{c_i})$ is the single-character word comprised of c_i. In addition, if a word set is empty, a special word "NONE" is added to the empty word set.

Word-Character Fusion The key to word-character fusion is to condense the four word sets of each character into a fixed-dimensional vector. In order to retain information as much as possible, we choose to concatenate the representations of the four word sets to represent them as a whole and add it to the character representation:

$$\mathbf{e}^c(\mathrm{B},\mathrm{M},\mathrm{E},\mathrm{S}) = [\mathrm{v}^c(\mathrm{B}) \oplus \mathrm{v}^c(\mathrm{M}) \oplus \mathrm{v}^c(\mathrm{E}) \oplus \mathrm{v}^c(\mathrm{S})], \tag{8}$$

$$\mathbf{x}^c \leftarrow [\mathbf{x}^c; \mathbf{e}^c(\mathrm{B},\mathrm{M},\mathrm{E},\mathrm{S})], \tag{9}$$

where v^c denotes the function that maps a single word set to a dense vector.

3.3 Data Augmentation

Since entity-labeled data is much smaller than non-entity-labeled data, the available valid data is relatively small. Therefore, we introduce an AR-GAN method to alleviate the problem of entity-labeled data imbalance and improve the performance of NER.

As shown in Figure 4, the inputs of the ranker R_ϕ consist of one synthetic sequence and multiple raw word-character fusion sentences. Given the reference sentence $\mathbf{U}$, we rank the input sentences according to the relative scores. It is illustrated that the generator tries to fool the ranker and let the synthetic sentence to be ranked at the top with respect to the reference sentence.

G_θ's learning goal is to generate a synthetic sequence that gets a higher score than real data. However, the goal of R_ϕ is to rank the synthetic sentence lower than word-character fusion sentences. Thus, this can be treated as G_θ and R_ϕ play a minimax game with the objective function ψ:

$$\begin{aligned}
\min_\theta \max_\phi \psi(G_\theta, R_\phi) = &\mathop{\mathbb{E}}_{s\sim\Upsilon_h}[\log R_\phi(s|\mathbf{U}, C^-)] \\
&+ \mathop{\mathbb{E}}_{s\sim G_\theta}[\log(1 - R_\phi(s|\mathbf{U}, C^+))],
\end{aligned} \tag{10}$$

where θ and ϕ are the variable parameters in $\mathbf{G}$ and $\mathbf{S}$, respectively. $\mathbb{E}$ is the expectation operator. $s \sim \Upsilon_h$ and $s \sim G_\theta$ denote that s is from word-character fusion sentences and synthesized sentences, respectively. $\mathbf{U}$ is the reference set used for estimating relative ranks. C^+ and C^- are the comparison set with regard to different input sentences s.

To avoid trivializing description, we put the details of rank score and policy gradient in Appendix B.

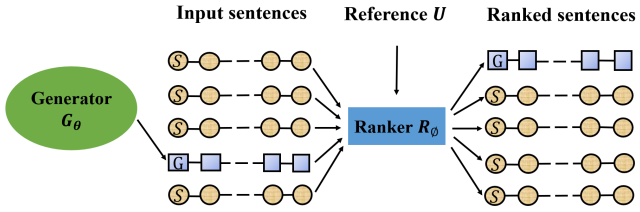

Figure 4: The illustration of AR-GAN. $\mathbf{S}$ denotes the raw data sampled from the word-character aggregation sentences. $\mathbf{G}$ is the sentence generated by the generator G_θ, $\mathbf{U}$ is fused by word-character.

3.4 Adaptive Transformer Encoding

The canonical self-attention in (Vaswani et al. 2017) is defined based on the tuple inputs, i.e, query, key and value, which performs the scaled dot-product as $\mathcal{A}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}(\mathbf{Q}\mathbf{K}^T/\sqrt{d_k})\mathbf{V}$, where $\mathbf{Q} \in \mathbb{R}^{l_q \times d}$, $\mathbf{K} \in \mathbb{R}^{l_k \times d}$, $\mathbf{V} \in \mathbb{R}^{l_v \times d}$ and d is the input dimension.

Self-lattice attention with relative position encoding is designed to model character-level self-correlations, which takes the character features after data augmentation F and relative position encoding P as inputs. This module is a variant of the multi-head attention mechanism, which can be formulated as:

$$head_i = \mathrm{softmax}((\mathbf{Q}W_i^{\mathbf{Q}})\mathbf{K}[i]^T + P[i])(\mathbf{V}W_i^{\mathbf{V}}), \tag{11}$$

$$O = [head_1; ...; head_z]W^o, \tag{12}$$

where $W_i^{\mathbf{Q}} \in \mathbb{R}^{d \times d/z}$, $W_i^{\mathbf{V}} \in \mathbb{R}^{d \times d/z}$, $W_i^{\mathbf{o}} \in \mathbb{R}^{d \times d}$ are trainable parameters, $K[i] \in \mathbb{R}^{n \times d/z}$ is the i-th partition of K, and $P[i] \in \mathbb{R}^{n \times n}$ contains relative position information of the i-th partition.

3.5 Decoding and Training

Considering the dependency between successive labels, we use a CRF layer to make sequence labeling. Given the sequence of final node states $c_1^T, c_2^T, \ldots, c_n^T$, the probability of a label sequence $\widehat{y} = \widehat{y}_1, \widehat{y}_2, \ldots, \widehat{y}_n$ can be formulated as:

$$p(\widehat{y}|s) = \frac{exp(\sum_{i=1}^{n} \Phi(\widehat{y}_{i-1}, \widehat{y}_i, c_i^T))}{\sum_{y' \in \mathrm{Y(s)}} exp(\sum_{i=1}^{n} \Phi(\widehat{y'}_{i-1}, \widehat{y'}_i, c_i^T))}, \tag{13}$$

where $\Phi(y_{i-1}, y_i, c_i^T) = W_{(y_{i-1}, y_i)} c_i^T + b_{(y_{i-1}, y_i)}$ is the scoring function, and $W_{(y_{i-1}, y_i)} c_i^T$ and $b_{(y_{i-1}, y_i)}$ are the weight vector and bias, Y(s) is the set of all arbitrary label sequences.

Given N manually labeled data $(s_i, y_i)|_{i=1}^N$, we minimize the sentence-level log-likelihood loss to train the model:

$$\mathcal{L}_{ner} = -\sum_{i=1}^{n} log(p(y_i|s_i)). \tag{14}$$

4 Experiments

In this section, we evaluate our method on manually annotated and public datasets, and show that our system outperforms baselines. Precision (Prec.), recall and F1 are used as evaluation metrics for this work.

4.1 Datasets

CTI Datasets In order to solve the issue of scarcity and imbalance of entity categories in existing datasets, we have collected some CTI datasets for comprehensive experiments. In particular, we have used crawler tools to collect data from open network threat intelligence such as the web, security community blogs, security reports, etc. The label of each sentence is manually marked and contains 17 entity types (see Appendix C.1 for details).

Most of public datasets ($i.e.$, OntoNotes, MSRA, Resume, Weibo, etc.) only focus on entity class: Person, Location, Organization and Misc. However, the entity categories covered by these existing NER datasets are not comprehensive enough, so we expand some network entity categories. As shown in Table 1, we split the dataset into three parts: training set, testing set, and development set, which contain 2,417, 1,050, and 3,671 entities, respectively.

Public Datasets We conducted experiments on four mainstream Chinese NER benchmarking datasets.

- **Resume** (Zhang and Yang 2018): It is composed of resumes collected from Sina Finance[2] and is annotated with 8 types of named entities, $i.e.$, CONT, EDU, LOC, PER, ORG, PRO, RACE, and TITLE.

[2]https://finance.sina.com.cn/stock/

Datasets	Types	Train	Test	Dev
Resume	Sentences	3.82k	0.48k	0.46k
	Entities	1.34k	0.15k	0.16k
MSRA	Sentences	46.36k	4.37k	-
	Entities	74.80k	6.20k-	-
Weibo	Sentences	1.35k	0.27k	0.27k
	Entities	1.89k	0.42k	0.39k
OntoNotes	Sentences	15.72k	4.31k	4.30k
	Entities	4.32k	7.70k	6.95k
CTI	Sentences	0.22k	0.13k	1.18k
	Entities	2.42k	1.05k	3.67k

Table 1: Statistics of five benchmarking datasets.

- **MSRA** (Levow 2006): It is a manually annotated multilingual corpus in the news domain and contains 3 types of entities, $i.e.$, ORG, PER, and LOC.
- **Weibo** (Peng and Dredze 2015; He and Sun 2016): It consists of annotated NER messages drawn from the social media Sina Weibo[3] and the corpus contains 4 types of entities, $i.e.$, PER, ORG, LOC, and GPE.
- **OntoNotes 4.0** (Weischedel et al. 2011): It is also a dataset in the news domain and contains 4 types of entities, $i.e.$, PER, ORG, LOC, and GPE.

4.2 Baselines

In this work, the baselines mainly include two groups of models: previous SoTA models and the variant models of our model. The models are listed as follows:

Previous SoTA Methods To illustrate how well our model can handle NER tasks, we compare our proposed model with the following existing SoTA models, including Lattice LSTM (Zhang et al. 2018), LR-CNN (Gui et al. 2019a), LGN (Gui et al. 2019b), TE-NER (Yan et al. 2019), CAN-NER (Zhu et al. 2019), PLTE-NER (Xue et al. 2020), Soft-Lexicon-LSTM (Peng et al. 2020), FLAT (Li et al. 2020b), ZEN 2.0 (Song et al. 2021), Lattice-BERT (Lai et al. 2021), MECT-NER (Wu et al. 2021) and RICON-NER (Gu et al. 2022). Due to space limitation, the details of baseline settings are given in Appendix C.2.

Variant Models To analyze the contribution of each component in our model, we ablate the full model and demonstrate the effectiveness of each component.

- **SoftLexicon-Trans-NER:** We conducted an ablation experiment to verify the effectiveness of lexical attention, using the SoftLexicon-Trans-NER model to compare the LA-Trans-NER model.
- **LA-Trans-NER:** This model is a part of our model without the data augmentation component. Obviously, we are to verify the effectiveness of this component on model improvement.

[3]https://www.weibo.com

Models	Resume			MSRA			Weibo			OntoNotes		
	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1	Prec.	Recall	F1
Lattice LSTM[1]	94.81	94.11	94.46	73.88	92.79	93.18	52.71	53.92	53.92	74.89	71.56	73.88
LR-CNN[2]	95.37	94.84	95.11	94.50	92.93	93.71	65.06	50.00	56.54	76.40	72.60	74.45
LGN[3]	95.28	95.46	95.37	94.19	92.73	92.73	-	-	-	76.13	73.68	74.89
TE-NER[4]	-	-	95.00	-	-	92.74	-	-	58.17	-	-	74.43
CAN-NER[5]	95.05	94.82	94.94	93.53	92.42	92.97	-	-	-	75.05	72.29	73.64
PLTE-NER[6]	95.34	95.46	95.40	94.25	92.30	93.26	62.21	49.54	55.15	76.78	72.54	74.60
SoftLexicon (LSTM)[7]	95.53	**95.64**	95.59	93.56	93.44	93.50	56.99	61.41	61.24	77.31	73.85	75.54
FLAT[8]	95.86	-	94.93	**96.09**	-	94.35	**68.55**	-	**63.42**	81.82	-	75.70
ZEN 2.0[9]	-	-	-	-	-	96.20	-	-	-	-	-	**88.81**
Lattice-BERT[10]	-	-	-	-	-	**97.10**	-	-	-	-	-	-
MECT-NER[11]	**96.40**	95.39	**95.89**	94.55	94.09	94.32	61.91	**62.51**	63.30	77.57	76.27	76.92
RICON-NER[12]	-	-	-	95.94	**96.33**	96.14	-	-	-	**81.95**	**84.78**	83.33
SoftLexicon-Trans-NER	93.20	93.37	93.29	90.81	90.83	90.10	52.72	54.64	54.38	75.01	70.72	71.91
LA-Trans-NER	96.61	95.85	95.23	96.62	96.99	96.80	66.58	62.85	67.39	82.82	79.28	84.92
LADA-CNN-NER	90.28	93.50	91.86	91.12	92.58	91.85	60.89	54.49	54.10	75.29	74.10	75.85
LADA-LSTM-NER	97.10	96.37	97.12	97.25	96.58	**98.89**	66.80	63.81	68.10	83.15	85.90	85.36
LADA-Trans-NER (Ours)	**98.89**	**97.78**	**98.70**	**98.90**	**97.21**	98.50	**68.89**	**65.90**	**70.18**	**84.19**	**86.56**	**85.91**

Table 2: Main results (Prec., Recall, and F1) on Resume, MSRA, Weibo and OntoNotes datasets. Zhang et al. (2018)[1], Gui et al.(2019a)[2], Gui et al. (2019b)[3], Yan et al. (2019)[4] , Zhu et al. (2019)[5], Xue et al. (2020)[6], Peng et al. (2020)[7], Li et al. (2020b)[8], Song et al. (2021)[9], Lai et al. (2021)[10], Wu et al. (2021)[11], Gu et al. (2022)[12].

- **LADA-CNN-NER:** This model is a variant of our model, but we utilize CNN in the sequence encoding layer.
- **LADA-LSTM-NER:** This model is another variant of our model, but we utilize LSTM instead of Transformer in the sequence encoding layer.

4.3 Results and Discussion

The results on public datasets and manually annotated datasets are shown in Table 2 and Table 3 respectively. We have gathered several experiment findings from the results.

Discussion on SoTA Methods First, compared with other SoTA methods, FLAT has the highest precision on MSRA and Weibo datasets, reaching 96.09% and 68.55% respectively. ZEN 2.0 performs best on OntoNotes, the F1 score has reached 88.81%. Meanwhile, RICON-NER has higher precision than other SoTA methods on OntoNotes, reaching 81.95%. Lattice-BERT has performed well on MSRA, with F1 score of 97.10%. MECT-NER has higher precision of 96.40% on Resume, but it has poor performance on CTI. The reason is that CTI datasets contain many non-Chinese special characters, but the MECT-NER model mainly integrates the radical information of Chinese character structure.

Second, compared with other SoTA methods, the Softlexicon model shows a better effect on CTI datasets of multi-source entities. From Table 3, the maximum accuracy is 93.84%, which is only 1.77% lower than our model. Therefore, the lexicon-based model can greatly improve the performance on CTI datasets.

Third, from Table 2 and Table 3, we can observe that: **(1)** Compared with other models, LADA-Trans-NER model has the best performance in precision (98.89% on Resume, 98.90% on MSRA, 68.89% on Weibo, 84.19% on OntoNotes and 95.61% on CTI). **(2)** Our model is more suitable for multi-source complex datasets or other sequence labeling tasks. LADA-Trans-NER model achieved competitive performance by training on our training set, then evaluating on our testing set.

Models	Prec.	Recall	F1
Lattice LSTM	85.50	80.19	81.02
LR-CNN	83.02	87.91	88.00
LGN	84.50	83.41	82.47
TE-NER	77.10	77.25	73.18
CAN-NER	79.02	78.30	88.15
SoftLexicon (LSTM)	93.84	90.95	91.50
FLAT	93.17	-	91.95
ZEN 2.0	-	-	93.75
MECT-NER	54.19	53.01	52.50
SoftLexicon-Trans-NER	90.95	89.62	90.10
LA-Trans-NER	92.90	92.95	91.64
LADA-CNN-NER	89.20	89.42	89.01
LADA-LSTM-NER	93.89	93.61	93.73
LADA-Trans-NER (Ours)	**95.61**	**95.85**	**95.23**

Table 3: Main results (Prec., Recall, and F1) on CTI datasets.

Ablation Study All the components of our model play an important role in improving performance. If any component is missing, then the performance will decrease. We also conducted additional experiments on LADA-Trans-NER with ablation consideration.

The performance of SoftLexicon-Trans-NER: Compared with the LA-Trans-NER model, the precision of the

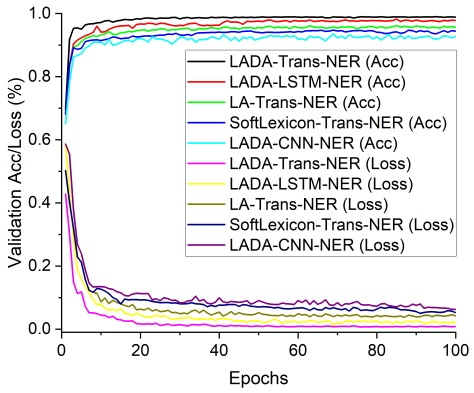

Figure 5: Performances of variant models on CTI datasets.

LA-Trans-NER model have decreased in different degrees (3.41%↓ on Resume, 5.81%↓ on MSRA, 13.86%↓ on Weibo, 7.81%↓ on OntoNotes, 1.95%↓ on CTI). Experiments on five datasets found that lexical attention mechanism plays a key role in improving the performance of NER system.

The performance of LA-Trans-NER: In this study, we removed the data augmentation module as a reference. Compared with our model, the F1 score have decreased in different degrees (3.47%↓ on Resume, 1.70%↓ on MSRA, 2.79%↓ on Weibo, 0.99%↓ on OntoNotes). Among them, the F1 score dropped the most by 3.59% on CTI datasets. Therefore, the AR-GAN data augmentation component can well solve the issue of unbalanced entity labels, and it also greatly promotes the performance of NER system.

The performance of LADA-CNN-NER: In this study, we replaced the sequence encoding layer with CNN to verify the performance of Transformer. From Figure 5, the performance of this variant model is worse than others, with low accuracy and relatively large loss on CTI datasets. Similarly, it can be seen that LADA-CNN-NER performs the worst than other variant models. Compared with our model, the precision decreased by 8.9% on OntoNotes (see Table 2).

The performance of LADA-LSTM-NER: In this study, we used LSTM as the sequence encoding layer to continuously verify the performance of Tranformer. It can be seen from Table 2 that the effectiveness of this model is better than other variant models, only inferior to our model. Most obviously, the F1 score of LADA-LSTM-NER model is 98.89% on MSRA, which exceeded our model in this indicator (↑ 0.49%). Therefore, we found that Transformer can play a positive role in improving the performance of NER system by combining lexical-attention and data-augmentation module.

4.4 Case Study

To intuitively verify that our model can better utilize fine-grained correlations in word-character spaces, we analyze two examples from CTI datasets (see Appendix for details).

In the first case, due to the inherently sequential nature, the character "国(Country)" has only access to its self-matched words "英国(Britain)" in the lattice LSTM. Hence, the lattice LSTM incorrectly recognizes "英国(Britain)" as a geopolitical entity. Similarly, although LR-CNN introduced the mechanism of lexical rethinking, it failed to identify accurately. Since Softlexicon uses lexical weighting, the final entity is also " 英国(Britain)". In the latest RICON-NER model, " 英国组织(Britain Organization)" conforms to the regularity "XX + 组织(Organization)" and is recognized as organization type. However, in the latter half of the sentence, "邮件钓鱼攻击(Mail Phishing Attack)" divides "邮件(Mail)" and "钓鱼攻击(Phishing Attack)" into two different entities. LADA-Trans-NER can correctly detects the attack entity "邮件钓鱼攻击(Mail Phishing Attack)". The reason is that LADA-Trans-NER can fully capture the longest lexical information "邮件钓鱼攻击(Mail Phishing Attack)", eliminating the interference of other irrelevant lexical information.

In the second case, there is an organization entity "360 安全大脑(360 Security Brain)" and a report entity "2020 全球高级持续性威胁研究报告(2020 Global Advanced Persistent Threat Research Report)". It is difficult for lattice LSTM to detect the uncommon entity, but can only recognize simple entities " 大脑(Brain)", "全球(Global)" and "研究报告(Research Report)". Compared with lattice LSTM, LR-CNN performs better, and can detect the organizational entity "安全大脑(Security Brain)" and the report entity "全球高级持续性威胁研究报告(Global Advanced Persistent Threat Research Report)", but the entity name is not fully identified. Softlexicon can accurately detect organizational entities "360 安全大脑(360 Security Brain)", but the latter half of the sentence is incorrectly recognized as "研究报告(Research Report)". In RICON-NER model, "2020 全球高级持续性威胁研究报告(2020 Global Advanced Persistent Threat Research Report)" follows the specific pattern "XX + 报告(XX + Report)" which ends with indicator word "XX + 报告(XX + Report)" and mostly belongs to report type. However, this model incorrectly recognizes "360 安全大脑(360 Security Brain)" as non-entity. LADA-Trans-NER can accurately exploit vocabulary information and filter out irrelevant words. These results show that the exact matching between each pair of character and word is critical, and our model can better understand the context semantics.

4.5 Parameter Sensitivity

In this part, we evaluate our model on different settings of the parameters. Specifically, we are concerned about the impact of dropout, learning rate decay and the dimensions of the parameters. Due to the limited space, more details about the hyper-parameter settings can be found in Appendix C.3.

	Resume	MSRA	Weibo	OntoNotes	CTI
No	96.15	96.12	68.53	84.89	90.45
YES	98.70	98.50	70.18	85.91	95.23

Table 4: Results with and without dropout on Resume, MSRA, Weibo, OntoNotes and CTI datasets (F1 score).

Dim.	Resume	MSRA	Weibo	OntoNotes	CTI
10	92.13	92.83	65.59	80.50	89.45
30	94.82	94.36	66.63	82.33	91.72
50	98.70	98.50	70.18	85.91	95.23
100	97.12	97.01	68.92	84.18	92.21
150	96.95	96.63	68.36	83.01	91.83
200	95.23	95.52	68.90	83.19	90.10

Table 5: Results of our proposed model influenced by different word embedding dimension (F1 score).

First, we compared the results achieved by our model with and without dropout layers, and show those results in Table 4. All other hyper-parameters remain the same as our best model. After using dropout, the F1 score has improved in each dataset. This demonstrates the effectiveness of dropout in reducing overfitting. Dropout is essential for state of the art performance, and the improvement is statistically significant. Our model achieved an essential and improved performance, because of introducing dropout.

Second, we analyzed the parameter sensitivity of learning rate decay, and compared the results achieved by our model with and without learning rate decay. Similarly, all other hyper-parameters remain the same as our best model. After using learning rate decay, the accuracy has improved on each dataset (see Appendix for details). Therefore, learning rate decay is very effective in finding global optimization.

Third, we evaluated our model on different parameter's dimensions. From Table 5, we listed the result our model achieved on different word embedding dimension. In our work, we discovered that when the dimension equals 50, we get the best results in our model.

4.6 Computation Efficiency

Table 6 lists the running speeds during training and inference of four baselines and our model. For fair comparison, all of these models are implemented using PyTorch and tested using the NVIDIA GeForce MX250 GPU. Due to the restriction of variable-sized set of matched words, Lattice LSTM (Zhang et al. 2018) and LR-CNN (Gui et al. 2019a) are non-batch parallel, while LGN (Gui et al. 2019b), LA-Trans-NER and LADA-Trans-NER can leverage parallel computation of GPU. First, LADA-Trans-NER (batch size=16) runs 4.07, 3.87, and 2.13 times faster than lattice LSTM, LR-CNN, and LGN (batch size=16) on the training speed, respectively. Furthermore, the inference speed of our model are about 4.53, 4.05, and 2.23 times faster than the transition-based model lattice LSTM, LR-CNN, and LGN respectively, which verify the efficiency of our model.

To further investigate the influence of sentence length, we analyze the performance of our LADA-Trans-NER model and other baseline approaches with respect to different grouped sentence lengths, as shown in Figure 6. We partition the sentence length into five groups ([0-14], [15-29], [30-44], [45-59], [$\geq$60]). We can observe that LADA-Trans-NER consistently runs faster than compared baselines under different sentence lengths. Especially, when the sentence length is less than 15, LADA-Trans-NER (batch size=16) runs 4.53, 4.05, and 2.23 times faster than Lattice LSTM, LR-CNN, and LGN (batch size=16) respectively. However, the speed gap becomes smaller as the sentence length increases. In summary, the LADA-Trans-NER model firmly outperforms current LSTM-based, CNN-based, and Graph-based methods in terms of efficiency.

Models	Training (sent/s)	Inference (sent/s)
Lattice LSTM	28.70±2.62	32.01±0.25
LR-CNN	30.20±1.81	35.82±0.64
LGN	26.35±1.56	34.90±1.02
LGN*	54.91±1.78	65.01±0.94
LADA-Trans-NER	35.05±2.13	36.11±1.43
LADA-Trans-NER*	116.95±1.24	145.02±1.08

Table 6: Running speed of different models, compared with Lattice LSTM, LR-CNN, LGN. The default batch size is 1, while * denotes the model is run with 16 batch size.

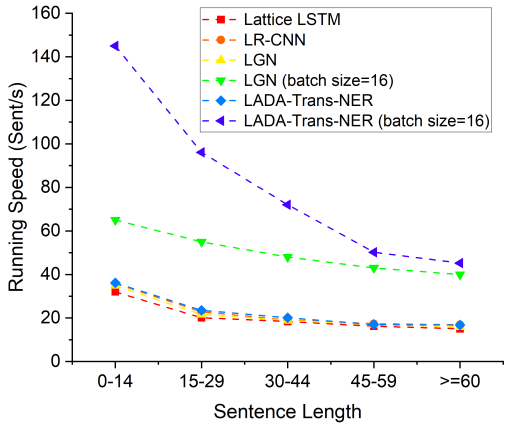

Figure 6: Running speed against sentence length. Sen/s denotes the number of sentences processed per second.

5 Conclusions

In this work, we proposed a novel Lexicon-Attention and Data-Augmentation (LADA) method for Chinese NER, which effectively integrates word-character information. We introduced a lexicon-attention mechanism to capture the local composition and potential word boundaries by using the lexicon knowledge. We further choose to concatenate the representations of the four word sets to represent them as a whole and add it to the character representation. Specially, we introduced an adaptive data augmentation method to alleviate the problem of entity label data imbalance. The model was trained and tested in an NER setting. Experimental results show that LADA is superior to most NER systems in the literature and it can be applied to scenarios with imperfect entity labeling on CTI datasets.

Acknowledgments

We sincerely thank the anonymous reviewers for their insightful comments and suggestions that helped improve the paper. This work was partly supported by the National Key Research and Development Program of China (No. 2019YFB1005204), and partly by the Key Deployment Projects of the Chinese Academy of Sciences (No. E1X0081104, No. KGFZD-145-21-03).

References

Cao, P.; Chen, Y.; Liu, K.; Zhao, J.; and Liu, S. 2018. Adversarial Transfer Learning for Chinese Named Entity Recognition with Self-Attention Mechanism. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 182-192.

Chawla, A.; Mulay, N.; Bishnoi, V.; and Dhama, G. 2021. KARL-Trans-NER: Knowledge Aware Representation Learning for Named Entity Recognition using Transformers. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 15436-15445.

Cheng, Q.; Liu, J.; Qu, X.; Zhao, J.; Liang, J.; Wang, Z.; Huai, B.; Yuan, N.; and Xiao, Y. 2021. HacRED: A Large-Scale Relation Extraction Dataset Toward Hard Cases in Practical Applications. In *Findings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP)*, 2819-2831.

Chiu, J. P.; and Nichols, E. 2016. Named Entity Recognition with Bidirectional LSTM-CNNs. In *Transactions of the Association for Computational Linguistics (TACL)*, 357-370.

Collobert, R.; and Weston, J. 2008. A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Machine Learning, Proceedings of the 25th International Conference (ICML)*, 160-167.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. P. 2011. Natural Language Processing (Almost) from Scratch. In *Journal of Machine Learning Research*, 12(1): 2493-2537.

Diao, S.; Bai, J.; Song, Y.; Zhang, T.; and Wang, Y. 2020. ZEN: Pre-training Chinese Text Encoder Enhanced by N-gram Representations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 4729-4740.

Dong, C.; Zhang, J.; Zong, C.; Hattori, M.; and Di, H. 2016. Character-Based LSTM-CRF with Radical-Level Features for Chinese Named Entity Recognition. In *Natural Language Understanding and Intelligent Applications*, 239-250. Springer.

Gregoric, A. Z.; Bachrach, Y.; and Coope, S. 2018. Named Entity Recognition With Parallel Recurrent Neural Networks. In *Proceedings of the 56th Annual Meeting on Association for Computational Linguistics*, 69-74.

Gui, T.; Ma, R.; Zhang, Q.; Zhao, L.; Jiang, Y.-G.; and Huang, X. 2019a. CNN-Based Chinese NER with Lexicon Rethinking. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 4982-4988.

Gui, T.; Zou, Y.; Zhang, Q.; Peng, M.; Fu, J.; Wei, Z.; and Huang, X.-J. 2019b. 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)*, 1039-1049.

Gu, Y.; Qu, X.; Wang, Z.; Huai, B.; Yuan, N. J.; and Gui, X. 2021. Read, Retrospect, Select: An MRC Framework to Short Text Entity Linking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 12920-12928.

Gu, Y.; Qu, X.; Wang, Z.; Zheng, Y.; Huai, B.; and Yuan, N. J. 2022. Delving Deep into Regularity: A Simple but Effective Method for Chinese Named Entity Recognition. In *Proceedings of the 2022 North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 1863-1873.

Habibi, M.; Weber, L.; Neves, M. L.; Wiegandt, D. L.; and Leser, U. 2017. Deep Learning with Word Embeddings improves Biomedical Named Entity Recognition. In *Bioinformatics (Oxford, England)*, 33(14): 37-38.

He, H.; and Sun, X. 2017. F-Score Driven Max Margin Neural Network for Named Entity Recognition in Chinese Social Media. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 713-718.

Hou, F.; Wang, R.; He, J.; and Zhou, Y. 2020. Improving Entity Linking through Semantic Reinforced Entity Embeddings. In *Proceedings of the 58th Annual Meeting on Association for Computational Linguistics*, 6843-6848.

Huang, Z.; Xu, W.; and Yu, K. 2015. Bidirectional LSTM-CRF Models for Sequence Tagging. In *arXiv preprint arXiv:1508.01991*.

Ji, S.; Pan, S.; Cambria, E.; Marttinen, P.; and Yu, P. S. 2020. A Survey on Knowledge Graphs: Representation, Acquisition, and Applications. In *Computer Science arXiv preprint arXiv:2002.00388*.

Lafferty, J.; McCallum, A.; and Pereira, F. C. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Machine Learning, Proceedings of the 18th International Conference (ICML)*, 282-289.

Lai, Y.; Liu, Y.; Feng, Y.; Huang, S.; and Zhao, D. 2021. Lattice-BERT: Leveraging Multi-Granularity Representations in Chinese Pre-trained Language Models. In *Proceedings of the 59th Annual Meeting on Association for Computational Linguistics*, 1716-1731.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 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 (NAACL)*, 260-270.

Le, P.; and Titov, I. 2018. Improving Entity Linking by Modeling Latent Relations between Mentions. In *Proceedings of the 56th Annual Meeting on Association for Computational Linguistics*, 1595-1604.

Levow, G. A. 2006. The third international Chinese language processing bakeoff: Word segmentation and named entity recognition. In *Proceedings of the 5th SIGHAN Workshop on Chinese Language Processing*, 108-117.

Li, X.; Yan, H.; Qiu, X.; and Huang, X. 2020b. FLAT: Chinese NER Using Flat-Lattice Transformer. In *Proceedings of the 58th Annual Meeting on Association for Computational Linguistics*, 6836-6842.

Lin, B. Y.; and Lu, W. 2018. Neural Adaptation Layers for Cross-domain Named Entity Recognition. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2012-2022.

Lin, K.; Li, D.; He, X.; Zhang, Z.; and Sun, M. T. 2017. Adversarial Ranking for Language Generation. In *Proceedings of the 31th Annual Conference on Neural Information Processing Systems (NIPS)*, 1-11.

Liu, W.; Xu, T.; Xu, Q.; Song, J.; and Zu, Y. 2019. An Encoding Strategy Based Word-Character LSTM for Chinese NER. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 2379-2389.

Ma, X.; and Hovy, E. H. 2016. End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting on Association for Computational Linguistics*, 4345-4357.

Peng, M.; Ma, R.; Zhang, Q.; and Huang, X. 2020. Simplify the Usage of Lexicon in Chinese NER. In *Proceedings of the 58th Annual Meeting on Association for Computational Linguistics*, 3827-3838.

Peng, N.; and Dredze, M. 2015. Named Entity Recognition for Chinese Social Media with Jointly Trained Embeddings. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 548-554.

Peng, N.; and Dredze, M. 2016. Improving Named Entity Recognition for Chinese Social Media with Word Segmentation Representation Learning. In *Proceedings of the 54th Annual Meeting on Association for Computational Linguistics*, 4321-4332.

Song, Y.; Zhang, T.; Wang, Y.; and Lee, K. F. 2021. ZEN 2.0: Continue Training and Adaption for N-gram Enhanced Text Encoders. In *arXiv preprint arXiv:2105.01279*.

Srivastava, N.; Hinton, G. E.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In *Journal of Machine Learning Research*, 15(1):1929-1958.

Sun, X; and He, H. 2017. F-score Driven Max Margin Neural Network for Named Entity Recognition in Chinese Social Media. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 713-718.

Takanobu, R.; Zhang, T.; Liu, J.; and Huang, M. 2019. A Hierarchical Framework for Relation Extraction with Reinforcement Learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7072-7079.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention Is All You Need. In *Advances in Neural Information Processing Systems*, 5998-6008.

Wei, Z.; Su, J.; Wang, Y.; Tian, Y.; and Chang, Y. 2020. A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. In *Proceedings of the 58th Annual Meeting on Association for Computational Linguistics*, 1476-1488.

Weischedel, R.; Pradhan, S.; Ramshaw, L.; Palmer, M.; Xue, N.; Marcus, M.; Taylor, A.; Greenberg, C.; Hovy, E.; Belvin, R.; et al. 2011. OntoNotes Release 4.0. In *LDC2011T03, Philadelphia, Penn.: Linguistic Data Consortium*.

Wu, S.; Song, X.; and Feng, Z. 2021. MECT: Multi-Metadata Embedding based Cross-Transformer for Chinese Named Entity Recognition. In *Proceedings of the 59th Annual Meeting on Association for Computational Linguistics*, 1529-1539.

Xue, M.; Yu, B.; Liu, T.; Zhang, Y.; Meng, E.; and Wang, B. 2019. Porous Lattice-based Transformer Encoder for Chinese NER. In *arXiv preprint arXiv:1911.02733*.

Yadav, V.; and Bethard, S. 2018. A Survey on Recent Advances in Named Entity Recognition from Deep Learning models. In *Proceedings of the 56th Annual Meeting on Association for Computational Linguistics*, 2145-2158.

Yan, H.; Deng, B.; Li, X.; and Qiu, X. 2019. TENER: Adapting Transformer Encoder for Name Entity Recognition. In *arXiv preprint arXiv:1911.04474*.

Yang, J.; Teng, Z.; Zhang, M.; and Zhang, Y. 2016. Combining Discrete and Neural Features for Sequence Labeling. In *International Conference on Intelligent Text Processing and Computational Linguistics*, 140-154.

Zhang, Y.; and Yang, J. 2018. Chinese NER Using Lattice LSTM. In *Proceedings of the 56th Annual Meeting on Association for Computational Linguistics*, 1554-1564.

Zhao, S.; Cai, Z.; Chen, H.; Wang, Y.; Liu, F.; and Liu, A. 2019. Adversarial Training Based Lattice LSTM for Chinese Clinical Named Entity Recognition. In *Journal of Biomedical Informatics* 99: 103290.

Zhao, S.; Hu, M.; Cai, Z.; and Liu, F. 2020. Modeling Dense Cross-Modal Interactions for Joint Entity-Relation Extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence (IJCAI-PRICAI)*, 4032-4038.

Zhao, S.; Hu, M.; Cai, Z.; Chen, H.; and Liu, F. 2021. Dynamic Modeling Cross- and Self-Lattice Attention Network for Chinese NER. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 14515-14523.

Zhu, Y.; Li, D.; and Wang, G. 2019. CAN-NER: Convolutional Attention Network for Chinese Named Entity Recognition. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 3384-3393.