TREE STRUCTURE LSTM FOR CHINESE NAMED ENTITY RECOGNITION

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we remodel the bi-directional LSTM (Bi-LSTM) network for Chinese Named Entity Recognition (NER). We convert LSTM from chain-like structure into tree structure which is fixed to the dependency parsing tree of the sentence. The new structure model can fully leverage the syntax information of the sentence and has the capability of capturing long-range dependencies. In addition, we use dependency parsing label embedding to improve the performance of our approach. Experimental studies on four benchmarking Chinese NER datasets have verified the effectiveness of our approach.

1 INTRODUCTION

Named Entity Recognition (NER) is mainly treated as a sequence labeling task. Bi-LSTM-CRF model (Huang et al., 2015) is a widely used network model for NER task. However, the structure of long short-term memory (LSTM) is chain-like structure which is confronted with the challenge of long-term dependencies. To alleviate this problem, we consider to use recursive neural networks with a computational graph which is structured as a deep tree. Socher et al. (2011; 2013) verify that the tree structure can be fixed to the structure of the parse tree of the sentence. By using tree structure to transfer information, the length of information path is reduced, and the structure contains syntax analysis as additional information. However, we use dependency parsing tree rather than parse tree in this paper. Dependence grammar is the framework that demonstrate the language structure with the dependent relationship between the words. The use of the dependence grammar for parsing is one of the important means of natural language understanding. Dependency parsing breaks the rule that the subject is the sentence center in the traditional syntax analysis, and takes the verb in the predicate as the center of a sentence. Besides, the dependent relationship between two words denotes that one word dominate another, which means the relationship is directional. So the dependency parsing tree is sufficed for replacing the chain structure as the carrier of the information flow for LSTM, hardly do traditional syntax analysis suffice.

NER models can be divided into character-based model and word-based model (Zhang & Yang, 2018), while tree structure comes with the word-based model. The advantage of word-based model compared to character-based model is that semantic information such as words information and syntax information can be used more precisely. The mainstream character-based algorithms that leverage words information are equipped with lattice model (Zhang & Yang, 2018), the usage of word information is indistinctly. Although the attention mechanism of transformer model (Vaswani et al., 2017) can automatically learn a number of syntax information, the search space is huge and the target function is not clear, the usage of information is not complete.

The disadvantage of word-based model is that propagating errors of upstream steps such as word segmentation will reduce validity of the model. Upstream errors will affect the detection of entity boundary and the prediction of entity category in NER. Propagating errors in our approach are segmentation errors, Part-of-Speech (POS) tagging errors and dependency parsing errors. However, with the upstream algorithms updating in the future, the propagation errors should be gradually reduced.

In this paper, we take the advantages of the word-based Chinese NER method in Zhang & Yang (2018), and remodel the Bi-LSTM (Graves & Schmidhuber, 2005) from chain-like structure into tree structure which is fixed to the dependency parsing tree of the sentence. We enrich the word

representation with arc label embedding. Besides, we equip the proposed method with pre-trained model.

2 BACKGROUND AND RELATED WORK

Character-based methods mainly use character and bi-character information. However, The lattice model has been widely used in character-based methods (Zhang & Yang, 2018; Gui et al., 2019b; Sui et al., 2019; Ma et al., 2020; Li et al., 2020) by encoding and matching words in the lexicon to capture word information. Moreover, radical-level information has also been used in Character-based methods (Dong et al., 2016; Han et al., 2021; Wu et al., 2021) to fuse the structural information of Chinese characters.

Word-based methods mainly use word and POS information. However, it's more effective when combining character information into word information (Zhang & Yang, 2018). Recently, Dependent information has been proved to be useful in several natural language processing (NLP) tasks, and it comes with word-based method. Zhang et al. (2018) propose using the dependency parse trees to construct a graph for relation extraction. Zhang et al. (2021) propose to use SHV which is regarded as dependency forests for event extraction. Chen & Kong (2021) add a GAT layer to capture the internal dependency of phrases for NER task.

Generic architectures for NER task include the Bi-LSTM (Zhang & Yang, 2018; Ma et al., 2020), the Convolutional Neural Network (Gui et al., 2019a) and the transformer (Xue et al., 2019; Li et al., 2020; Wu et al., 2021). For transformer variants, Chen & Kong (2021) use Star-transformer (Guo et al., 2019) to construct a lightweight baseline system, and Wu et al. (2021) propose a Multi-metadata Embedding based Cross-Transformer (MECT). Although transformer has resently exceeded the traditional recurrent neural network (RNN) in various tasks, Bi-LSTM still has its vitality on NER. Character-based method SoftLexicon in Ma et al. (2020) still use Bi-LSTM as its sequence modeling layer, and become one of state-of-the-art methods.

3 MODEL

3.1 CONVERTING LSTM INTO TREE STRUCTURE

Figure 1 illustrates the dependency parsing result of a Chinese sentence. The result of the dependency parsing can be transformed into the tree structure. Figure 2 illustrates the dependency parsing tree of this sentence.

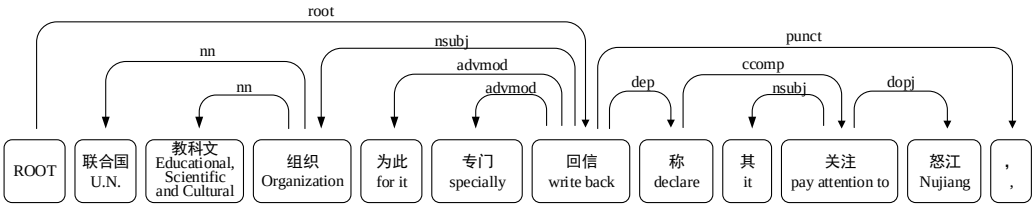

Figure 1: Dependency parsing of the sentence.

After getting dependency parsing tree from a sentence, LSTM can be converted from chain-like structure into tree structure. Since bidirectional LSTM is greater than mono directional LSTM, the tree structure LSTM is also consist of two directions. The route of forward direction is from root node to the leaf nodes along with the dependency parsing tree. The route of back-forward direction is from leaves to the root. Besides, before the root of forward direction, a back-forward direction process is carried out to merge the information of the sentence. Finally, the outputs of bidirectional LSTM are concatenated as the inputs of CRF nets. The whole structure of the model is shown in Figure 3.

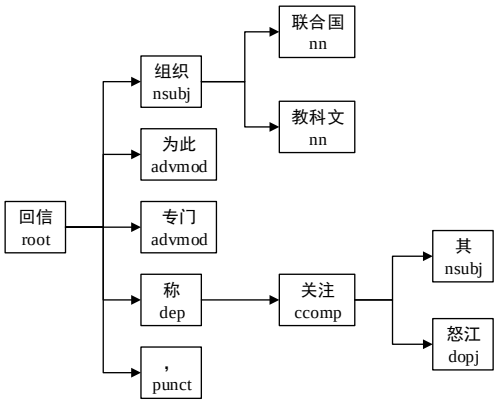

Figure 2: Dependency parsing tree of the sentence.

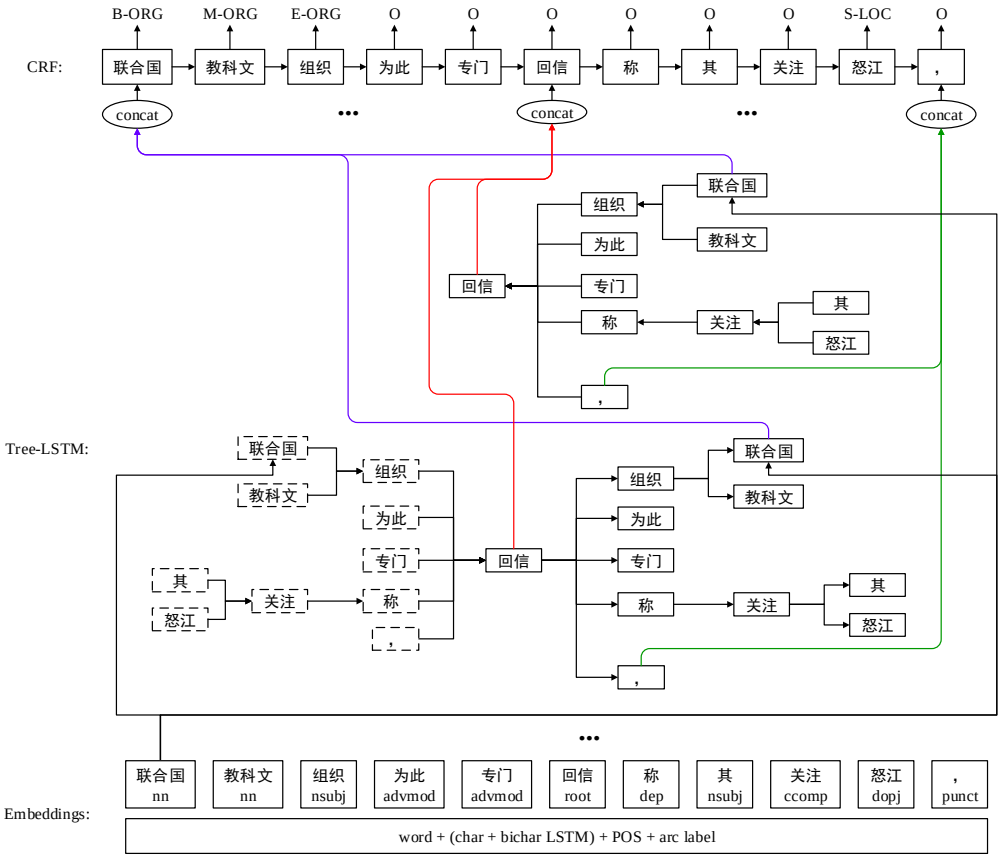

Figure 3: Overall structure of the model.

Embeddings: Both character CNN (Ma & Hovy, 2016) and LSTM (Lample et al., 2016) have been used for getting the information of the character sequence within a word. In accordance with the best result in Zhang & Yang (2018), we use word embedding + (char + bichar LSTM) as our baseline word representation. In char + bichar LSTM, a Bi-LSTM is used to learn hidden states for the character sequence in the word. For instance, $c_1, \ldots, c_{\text{len}(w)}$ denote the character sequence in the word w. The character represent is obtained by concatenation of character embedding and bi-character embedding. $\boldsymbol{h}^f_1, \ldots, \boldsymbol{h}^f_{\text{len}(w)}$ and $\boldsymbol{h}^b_1, \ldots, \boldsymbol{h}^b_{\text{len}(w)}$ respectively denote the forward direction and the backward direction hidden states for $c_1, \ldots, c_{\text{len}(w)}$. The final output for the representation of w is $[\boldsymbol{h}^f_{\text{len}(w)}; \boldsymbol{h}^b_{\text{len}(w)}]$.

At the meantime, we introduce POS tags and dependency parsing labels (arc labels) to enrich discrete representations. Although POS tags $\mathcal{P} = \{NN, VV, VA, AD, PU, \ldots\}$ and arc labels $\mathcal{L} = \{amod, tmod, nsubj, csubj, dobj, \ldots\}$ are relatively small discrete sets, they still exhibit many senmantical similarities like words (Chen & Manning, 2014). The final word representation is obtained by concatenation of word embedding, output of char + bichar LSTM, POS embedding and arc label embedding.

3.2 Tree Structure LSTM Model

Figure 3 illustrates the overall structure of the tree structure LSTM model (tree LSTM). The forward direction of tree LSTM consists of all routes which are from root node to leaf nodes. In contrast, the back forward direction routes are from leaves to root. Since two directions are not the same pattern, there are two kinds of LSTM cells in this model. The cell in the forward direction is the same as the original sequential LSTM cell. The information flows from one to one. The cell functions are :

$$\begin{bmatrix} \boldsymbol{i}_j \\ \boldsymbol{o}_j \\ \boldsymbol{f}_j \\ \boldsymbol{g}_j \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(\boldsymbol{W}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{x}_j \\ \boldsymbol{h}_{j-1} \end{bmatrix} + \boldsymbol{b} \right), \tag{1}$$

$$\boldsymbol{c}_j = \boldsymbol{f}_j \odot \boldsymbol{c}_{j-1} + \boldsymbol{i}_j \odot \boldsymbol{g}_j, \tag{2}$$

$$\boldsymbol{h}_j = \boldsymbol{o}_j \odot \tanh(\boldsymbol{c}_j). \tag{3}$$

where j^{th} denotes time steps. $\boldsymbol{i}_j$, $\boldsymbol{f}_j$ and $\boldsymbol{o}_j$ are a set of input, forget and output gates, respectively. $\boldsymbol{x}_j$ is the current input vector. $\boldsymbol{c}_j$ is the current state unit. $\boldsymbol{h}_j$ is the current hidden layer vector. $\boldsymbol{W}^{\mathrm{T}}$ and $\boldsymbol{b}$ denote respectively weight and bias parameters. $\sigma()$ represents the sigmoid function.

On the back forward direction, the information flows from several ($>= 1$) child cells to the parent cell. The information from these child cells should be merged. The cell functions are :

$$\begin{bmatrix} \boldsymbol{i}_{k,j} \\ \boldsymbol{o}_{k,j} \\ \boldsymbol{f}_{k,j} \\ \boldsymbol{g}_{k,j} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} \left(\boldsymbol{W}^{\mathrm{T}} \begin{bmatrix} \boldsymbol{x}_j \\ \boldsymbol{h}_{k,j-1} \end{bmatrix} + \boldsymbol{b} \right), k \in \mathbb{D}_j, \tag{4}$$

$$\boldsymbol{\alpha}_{k,j} = \frac{\exp(\boldsymbol{f}_{k,j})}{\sum_{k' \in \mathbb{D}_j} (\exp(\boldsymbol{f}_{k',j}) + \exp(\boldsymbol{i}_{k',j}))}, k \in \mathbb{D}_j, \tag{5}$$

$$\boldsymbol{\beta}_{k,j} = \frac{\exp(\boldsymbol{i}_{k,j})}{\sum_{k' \in \mathbb{D}_j} (\exp(\boldsymbol{f}_{k',j}) + \exp(\boldsymbol{i}_{k',j}))}, k \in \mathbb{D}_j, \tag{6}$$

$$\boldsymbol{c}_j = \sum_{k \in \mathbb{D}_j} (\boldsymbol{\alpha}_{k,j} \odot \boldsymbol{c}_{k,j-1}) + \sum_{k \in \mathbb{D}_j} (\boldsymbol{\beta}_{k,j} \odot \boldsymbol{g}_{k,j}), \tag{7}$$

$$\boldsymbol{c}_{k,j} = \boldsymbol{f}_{k,j} \odot \boldsymbol{c}_{k,j-1} + \boldsymbol{i}_{k,j} \odot \boldsymbol{g}_{k,j}, k \in \mathbb{D}_j, \tag{8}$$

$$\boldsymbol{o}_{k,j} = \frac{\exp(\boldsymbol{o}_{k,j})}{\sum_{k' \in \mathbb{D}_j} \exp(\boldsymbol{o}_{k',j})}, k \in \mathbb{D}_j, \tag{9}$$

$$\boldsymbol{h}_j = \sum_{k \in \mathbb{D}_j} (\boldsymbol{o}_{k,j} \odot \tanh(\boldsymbol{c}_{k,j})). \tag{10}$$

where $\mathbb{D}_j$ includes all the front (j^{th}-1) child cells of the current (j^{th}) parent cell. $\boldsymbol{x}_j$ is the current input vector. $\boldsymbol{h}_{k,j-1}$ is the hidden layer vector of the (k^{th}-1) child cell. $\boldsymbol{c}_{k,j-1}$ is the state unit of

the k^{th} child cell. $\boldsymbol{i}_{k,j}$, $\boldsymbol{f}_{k,j}$ and $\boldsymbol{o}_{k,j}$ are respectively input, forget and output gates corresponding to the k^{th} child cell. Parameters of $\boldsymbol{\alpha}_{k,j}$, $\boldsymbol{\beta}_{k,j}$, and $\boldsymbol{o}_{k,j}$ are calculated with softmax function. $\boldsymbol{c}_j$ is the current state unit. $\boldsymbol{h}_j$ is the current hidden layer vector.

3.3 CRF MODEL

After tree structure LSTM, we take the representation of $\{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_j, \ldots\}$ into a Conditional Random Field (CRF) layer (Lafferty et al., 2001). This layer is verified to be typically effective for sequence labeling, and widely used in the end of most NER models. Given a sequence $\boldsymbol{s} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \ldots, \boldsymbol{s}_n\}$, the probability of a label sequence $\boldsymbol{y} = \{y_1, y_2, \ldots, y_n\}$ can be defined:

$$\mathrm{P}(\boldsymbol{y}|\boldsymbol{s}) = \frac{\sum_{j=1}^{n} \exp(\mathrm{f}(y_{j-1}, y_j, \boldsymbol{s}_j))}{\sum_{y' \in \boldsymbol{Y}(\boldsymbol{s})} \sum_{j=1}^{n} \mathrm{f}(y'_{j-1}, y'_j, \boldsymbol{s}_j)}. \tag{11}$$

where $\boldsymbol{Y}(\boldsymbol{s})$ denotes all arbitrary label sequences, $\mathrm{f}(y_{j-1}, y_j, \boldsymbol{s})$ denotes the transition score from y_{j-1} to y_j and the score for y_j. The Viterbi Algorithm is used to calculate the maximum likelihood of $\mathrm{P}(\boldsymbol{y}|\boldsymbol{s})$.

3.4 HYPERPARAMETER CONFIGURATION

The hyperparameter values of the networks are detailed in Table 1. We keep the same settings of the hyperparameter values in Zhang & Yang (2018).

Table 1: Hyperparameter values used in the model.

Param	**Value**	**Param**	**Value**
word emb size	50	char hidden dim	50
char emb size	50	hidden dim	200
bichar emb size	50	dropout	0.5
POS emb size	50	learning rate *lr*	0.015
arc-label emb size	50	*lr* decay	0.05
batch size	1	momentum	0
LSTM layer	1		

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Data: We use four Chinese NER datasets, including: (1) **OntoNotes 4.0**[1] (Ralph et al., 2011). We adopt the same pre-process of OntoNotes as described in Che et al. (2013). (2) **MSRA** (Levow, 2006). (3) **Weibo NER**[2] (Peng & Dredze, 2015; He & Sun, 2016). (4) **Resume NER**[3] (Zhang & Yang, 2018). OntoNotes and MSRA are datesets of news domain, while Weibo NER and Resume NER are come from the website Sina Weibo and Sina Finance, respectively.

Segmentation: Gold-standard word segmentation is available in OntoNotes and the training set of MSRA. On the other hand, no segmentation is available in the testing set of MSRA, nor the Weibo or Resume dataset. We use the word segmentor of Yang et al. (2017), and segment the datasets in the same way as Zhang & Yang (2018) except MSRA. However, the entities in the training set of MSRA are not segmented. There are only 'S' and 'O' tags in the training set. Hence, after segmenting the testing set with the model trained on the training set, we manually correct the entities as gold segmentation result. On the other hand, the training set and the testing set are automatically segmented as auto segmentation result.

[1] https://catalog.ldc.upenn.edu/LDC2011T13

[2] https://github.com/cchen-nlp/weiboNER

[3] https://github.com/jiesutd/LatticeLSTM

POS: In four datasets, gold-standard POS tagging is only available in OntoNotes. The POS tags are automatically assigned with Standford CoreNLP tool (4.2.2 release)[4].

Dependency parse: In four datasets, gold-standard parsing is only available in OntoNotes. We convert gold parsing trees into dependency parsing trees with Standford CoreNLP tool. Dependency representation is Stanford Basic Dependencies (De Marneffe et al., 2006) and dependency architecture is single tree. Standford CoreNLP tool is also used to automatically parse syntactic dependency for four datasets.

Embedding: Word, character and character bigram embeddings are pretrained on Chinese Giga-Word using word2vec (Mikolov et al., 2013), which are released by Zhang & Yang (2018). We use the sentences with POS tags in Ontonotes 5.0[5] (Pradhan et al., 2013) dataset as the training corpus for word2vec, and the Gensim[6] tool is used to generate embeddings of POS tags. The training parameters are shown in Table 2. We extract one million sentences from sogou news corpus[7] (SogouCS), and use the Stanford CoreNLP tool to parse the sentences into dependency syntax. After that, we extract all the paths from root to leaf node of each sentence, e.g., $amod \rightarrow tmod \rightarrow subj \rightarrow csubj \rightarrow dobj$. These paths are taken as the training corpus for word2vec, and the Gensim tool is also used to generate embeddings of dependency labels (arc labels). The training parameters are also shown in Table 2. All of embeddings above are fine-tuned at model training.

Table 2: Settings for word2vec in Gensim.

Param	**POS embeddings**	**Arc label embeddings**
Vector size	50	50
Algorithm	skip-gram	skip-gram
Window	5	2
Min count	0	0
Others	default	default

The NER label is marked by BMESO. Precision (P), Recall (R) and F1 score (F1) are used as evaluation metrics.

4.2 Overall Performance

We applied the word-based methods as baseline for comparison on OntoNotes, as shown in Table 3. 'BiLSTM (word based)' denotes the best Bi-LSTM word-based method in Zhang & Yang (2018). Our method is superior to this method by 3.04% (F1) for gold process and 2.62% (F1) for auto process.

Table 3: Main results on OntoNotes.

Input	**Models**	**Lexicon**	**P**	**R**	**F1**
Gold seg +Gold POS +Gold parse	Yang et al. (2016)	–	72.98	80.15	76.40
	BiLSTM (word based)	YJ	78.62	73.13	75.77
	Tree LSTM	YJ	77.15	80.54	**78.81**
Auto seg +Auto POS +Auto parse	BiLSTM (word based)	YJ	73.36	70.12	71.70
	Tree LSTM	YJ	72.91	75.78	**74.32**

We also compared our method with several state-of-the-art methods on the four datasets to verify its effectiveness, as shown in Table 4. The results of other methods are from their original papers. All

[4] https://stanfordnlp.github.io/CoreNLP

[5] https://catalog.ldc.upenn.edu/LDC2013T19

[6] https://github.com/RaRe-Technologies/gensim

[7] https://hyper.ai/datasets/5487

other methods are based on character except 'BiLSTM (word based)' and 'Star+GAT+MultiTask'. It can be seen that character-based method is the mainstream of NER in recent years. The proposed method is superior to character-based methods for gold process (on Ontonotes and MSRA), and the effect of it is close to character-based methods for auto process. 'Star+GAT+MultiTask' is the state-of-the-art word-based method. The proposed method is inferior to it by 1.14% on the OntoNotes and by 10.33% on the Weibo. In other words, the effect of the proposed method is close to it on the normative dataset, but much worse to it on the dataset with language problem in norm. Because the syntactic parse tree is used as the main structure and no additional strategies are added, there is requirement for the normalization of sentences.

Table 4: Four datasets results (F1). 'YJ' denotes the pre-trained embeddings released by Zhang & Yang (2018), and 'LS' denotes the pre-trained embeddings released by Li et al. (2018). Results of 'BiLSTM' and 'Lattice LSTM' are from Zhang & Yang (2018). Results of 'LR-CNN' are from Gui et al. (2019a). Results of 'LGN' are from Gui et al. (2019b). Results of 'PLT' are from Xue et al. (2019). Results of 'FLAT' are from Li et al. (2020). Results of 'SoftLexicon (LSTM)' are from Ma et al. (2020). Results of 'Star+GAT+MultiTask' are from Chen & Kong (2021). Results of 'MECT' are from Wu et al. (2021).

Model	**Pre-emb**	**Ontonotes**	**MSRA**	**Weibo**	**Resume**
BiLSTM (word based)	YJ	gold: 75.77 auto: 71.70	90.28	52.33	93.58
BiLSTM (char based)	YJ	71.81	91.87	56.75	94.41
TENER	–	72.82	93.01	58.39	95.25
Lattice LSTM	YJ	73.88	93.18	58.79	94.46
LR-CNN	YJ	74.45	93.71	59.92	95.11
LGM	YJ	74.85	93.63	60.15	95.41
PLT	YJ	74.60	93.26	59.92	95.40
FLAT	YJ	76.45	94.12	60.32	95.45
FLAT	LS	75.70	94.35	63.42	94.93
SoftLexicon (LSTM)	YJ	75.64	93.66	61.42	95.53
Star + GAT + MultiTask	LS	79.95	–	70.14	–
MECT	YJ	76.92	94.32	63.30	95.89
Tree LSTM	YJ	gold: **78.81** auto: **74.32**	gold: **95.94** auto: **91.76**	**59.81**	**94.73**

4.3 Ablation Experiments

As shown in Table 5, we remove a component of the proposed method each time to validate its effect. The ablation study is conducted with gold-standard process on the Ontonotes. POS embeddings and arc label embeddings are directly concatenated into the word presentations. 'Merged root' denotes the back-forward direction process before the root of forward direction in the tree LSTM. In the forward direction, the flow of the information is only along a branch of the tree. Each node can only obtain the flow of the parent node. They can't obtain the information of the bypass node. Therefore, the initialization of the root node is of great significance. We use 'merged root' to merge the information of the whole sentence for initialization instead of the random initialization. From table we can find that each component contributes to the proposed model.

4.4 Compatibility with BERT

Recently, the pre-trainded model has an excellent effect on all kinds of NLP tasks. We also equip our model with BERT (Devlin et al., 2018) on four datasets, and results are shown in Table 6. We use fastNLP[8] tool to generate BERT embeddings and concatenate them into the character and the word

[8] https://github.com/fastnlp/fastNLP

Table 5: An ablation study of the proposed model.

Model	P	R	F1
Tree LSTM	77.15	80.54	78.81
– POS embedding	79.58	73.27	76.29
– arc label embedding	77.23	79.29	78.25
– merged root	77.74	79.01	78.37

representations, respectively. The pre-trained BERT model is 'BERT-wwm' which is released by Cui et al. (2020). The BERT embedding size is 768. The character hidden dimension in char LSTM is set from 50 to 500, which can fully leverage BERT embedding.

Table 6: Results (F1) of models equipped with BERT. 'BERT' denotes the BERT+MLP+CRF architecture, and the results of it are from Li et al. (2020). 'BERT+model' represents the model in Table 4 which is equipped with BERT.

Model	Pre-emb	Ontonotes	MSRA	Weibo	Resume
BERT	–	80.14	94.95	68.20	95.53
BERT+ FLAT	YJ	81.82	96.09	68.55	95.86
BERT+ SoftLexicon (LSTM)	YJ	82.81	95.42	70.50	96.11
BERT+ MECT	YJ	82.57	96.24	70.43	95.98
BERT+ Tree LSTM	YJ	gold: **80.78** auto: **77.92**	gold: **97.13** auto: **94.07**	64.46	95.52

We find that the proposed model equipped BERT has improved significantly. The results of Weibo require special instructions. As shown on Figure 4, the final results (F1=64.46%) come from the best development results (F1=73.08%). However, the F1-value is the minimum after 20th epoch, while the best F1-value is up to 68.80%. All the other models in the Table 6 are character-based model. However, Chinese BERT model is originally designed to generate character embeddings rather than word embedings, the proposed model doesn't improve as much as these character-based models.

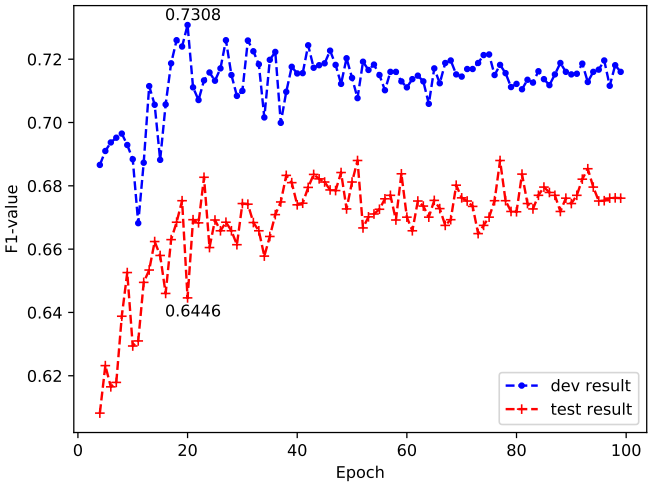

Figure 4: Results (F1) of the proposed model equipped with BERT on the Weibo dataset.

5 CONCLUSION

This paper presents a tree structure LSTM for Chinese NER. The tree structure model shows the capability of capturing long-range dependencies. With the use of arc label embedding, the performance of tree structure model is still further improved. Experimental studies on four Chinese NER datasets have verified the effectiveness of this model. Because propagating errors always exist in the inherent structure of the word-based model, further studies that consider joint model of segmentation, POS, dependency parsing and NER are necessary.

REFERENCES

Wanxiang Che, Mengqiu Wang, Christopher D. Manning, and Ting Liu. Named entity recognition with bilingual constraints. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 52–62, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL https://aclanthology.org/N13-1006.

Chun Chen and Fang Kong. 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)*, pp. 20–25, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-short.4. URL https://aclanthology.org/2021.acl-short.4.

Danqi Chen and Christopher Manning. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 740–750, Doha, Qatar, October 2014. Association for Computational Linguistics. doi: 10.3115/v1/D14-1082. URL https://aclanthology.org/D14-1082.

Yiming Cui, Wanxiang Che, Ting Liu, Bing Qin, Shijin Wang, and Guoping Hu. Revisiting pre-trained models for Chinese natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 657–668, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.findings-emnlp.58. URL https://aclanthology.org/2020.findings-emnlp.58.

Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Lrec*, volume 6, pp. 449–454, 2006.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Chuanhai Dong, Jiajun Zhang, Chengqing Zong, Masanori Hattori, and Hui Di. Character-based lstm-crf with radical-level features for chinese named entity recognition. In Chin-Yew Lin, Nianwen Xue, Dongyan Zhao, Xuanjing Huang, and Yansong Feng (eds.), *Natural Language Understanding and Intelligent Applications*, pp. 239–250, Cham, 2016. Springer International Publishing. ISBN 978-3-319-50496-4.

Alex Graves and Jürgen Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, 18(5):602–610, 2005. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2005.06.042. URL https://www.sciencedirect.com/science/article/pii/S0893608005001206. IJCNN 2005.

Tao Gui, Ruotian Ma, Qi Zhang, Lujun Zhao, Yu-Gang Jiang, and Xuanjing Huang. Cnn-based chinese ner with lexicon rethinking. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 4982–4988. International Joint Conferences on Artificial Intelligence Organization, 7 2019a. doi: 10.24963/ijcai.2019/692. URL https://doi.org/10.24963/ijcai.2019/692.

Tao Gui, Yicheng Zou, Qi Zhang, Minlong Peng, Jinlan Fu, Zhongyu Wei, and Xuanjing Huang. 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), pp. 1040–1050, Hong Kong, China, November 2019b. Association for Computational Linguistics. doi: 10.18653/v1/D19-1096. URL https://aclanthology.org/D19-1096.

Qipeng Guo, Xipeng Qiu, Pengfei Liu, Yunfan Shao, Xiangyang Xue, and Zheng Zhang. Star-transformer. 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)*, pp. 1315–1325, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1133. URL https://aclanthology.org/N19-1133.

Xuming Han, Feng Zhou, Zhiyuan Hao, Qiaoming Liu, Yong Li, and Qi Qin. Maf-cner: A chinese named entity recognition model based on multifeature adaptive fusion. *Complexity*, 2021, 2021.

Hangfeng He and Xu Sun. F-score driven max margin neural network for named entity recognition in chinese social media. *arXiv preprint arXiv:1611.04234*, 2016.

Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, abs/1508.01991, 2015. URL http://arxiv.org/abs/1508.01991.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pp. 282–289, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc. ISBN 1558607781.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 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*, pp. 260–270, San Diego, California, June 2016. Association for Computational Linguistics. doi: 10.18653/v1/N16-1030. URL https://aclanthology.org/N16-1030.

Gina-Anne Levow. The third international chinese language processing bakeoff: Word segmentation and named entity recognition. In *In Proceedings of the Fifth SIGHAN Workshop on Chinese Language Processing*, pp. 108–117, 2006.

Shen Li, Zhe Zhao, Renfen Hu, Wensi Li, Tao Liu, and Xiaoyong Du. Analogical reasoning on Chinese morphological and semantic relations. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 138–143, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-2023. URL https://aclanthology.org/P18-2023.

Xiaonan Li, Hang Yan, Xipeng Qiu, and Xuanjing Huang. FLAT: Chinese NER using flat-lattice transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6836–6842, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.611. URL https://aclanthology.org/2020.acl-main.611.

Ruotian Ma, Minlong Peng, Qi Zhang, Zhongyu Wei, and Xuanjing Huang. Simplify the usage of lexicon in Chinese NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 5951–5960, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.528. URL https://aclanthology.org/2020.acl-main.528.

Xuezhe Ma and Eduard Hovy. 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)*, pp. 1064–1074, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/P16-1101. URL https://aclanthology.org/P16-1101.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*, 26, 2013.

Nanyun Peng and Mark Dredze. Named entity recognition for chinese social media with jointly trained embeddings. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 548–554, 2015.

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

Weischedel Ralph, Palmer Martha, Marcus Mitchell, Hovy Eduard, Pradhan Sameer, Ramshaw Lance, Xue Nianwen, Taylor Ann, Kaufman Jeff, Franchini Michelle, El-Bachouti Mohammed, Belvin Robert, and Houston Ann. Ontonotes release 4.0 ldc2011t03. *Philadelphia: Linguistic Data Consortium*, 2011. doi: 10.35111/gfjf-7r50.

Richard Socher, Eric Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Ng. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger (eds.), *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/3335881e06d4d23091389226225e17c7-Paper.pdf.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

Dianbo Sui, Yubo Chen, Kang Liu, Jun Zhao, and Shengping Liu. 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)*, pp. 3830–3840, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1396. URL https://aclanthology.org/D19-1396.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Shuang Wu, Xiaoning Song, and Zhenhua Feng. MECT: Multi-metadata embedding based cross-transformer for 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 1: Long Papers)*, pp. 1529–1539, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.121. URL https://aclanthology.org/2021.acl-long.121.

Mengge Xue, Bowen Yu, Tingwen Liu, Bin Wang, Erli Meng, and Quangang Li. Porous lattice-based transformer encoder for chinese NER. *CoRR*, abs/1911.02733, 2019. URL http://arxiv.org/abs/1911.02733.

Jie Yang, Zhiyang Teng, Meishan Zhang, and Yue Zhang. Combining discrete and neural features for sequence labeling. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pp. 140–154. Springer, 2016.

Jie Yang, Yue Zhang, and Fei Dong. Neural word segmentation with rich pretraining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 839–849, Vancouver, Canada, July 2017. Association for Computational Linguistics. doi: 10.18653/v1/P17-1078. URL https://aclanthology.org/P17-1078.

Junchi Zhang, Qi He, and Yue Zhang. Syntax grounded graph convolutional network for joint entity and event extraction. *Neurocomputing*, 422:118–128, 2021. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2020.09.044. URL https://www.sciencedirect.com/science/article/pii/S0925231220314521.

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

Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1144. URL https://aclanthology.org/P18-1144.

Yuhao Zhang, Peng Qi, and Christopher D Manning. Graph convolution over pruned dependency trees improves relation extraction. *arXiv preprint arXiv:1809.10185*, 2018.