A Light Transfer Model for Chinese Named Entity Recognition for Specialty Domain

Jiaqi Wu[1], Tianyuan Liu[1,3], Yuqing Sun[1,2](✉), and Bin Gong[1,2]

[1] School of Software, Shandong University, Jinan, China
oofelvis@163.com, zodiacg@foxmail.com
[2] Engineering Research Center of Digital Media Technology, Shandong University, Jinan, China
{sun_yuqing,gb}@sdu.edu.cn
[3] School of Computer Science and Technology, Shandong University, Jinan, China

Abstract. Named entity recognition (NER) for specialty domain is a challenging task since the labels are specific and there are not sufficient labelled data for training. In this paper, we propose a simple but effective method, named Light Transfer NER model (LTN), to tackle this problem. Different with most traditional methods that fine tune the network or reconstruct its probing layer, we design an additional part over a general NER network for new labels in the specific task. By this way, on the one hand, we can reuse the knowledge learned in the general NER task as much as possible, from the granular elements for combining inputs, to higher level embedding of outputs. On the other hand, the model can be easily adapted to the domain specific NER task without reconstruction. We also adopt the linear combination on each dimension of input feature vectors instead of using vector concatenation, which reduces about half parameters in the forward levels of network and makes the transfer light. We compare our model with other state-of-the-art NER models on real datasets against different quantity of labelled data. The experimental results show that our model is consistently superior than baseline methods on both effectiveness and efficiency, especially in case of low-resource data for specialty domain.

Keywords: Named entity recognition · Light transfer model · Specialty domain

1 Introduction

Name entity recognition (NER) refers to the recognition of entities with specific meanings in the text, such as person, location, organization etc. It is a fundamental task in natural language processing that provides useful information on constructing knowledge graph, syntactic parsing or information retrieval.

There are lots of research works on the general NER task, One of the representative kind of methods focus on handcrafted rules, which need a lot of linguistic knowledge, and the rules of different languages or domains are not the same. Kim [10] proposed to use Brill rule and Hanisch et al. [5] proposed a system named ProMiner to recognize entities in biomedical text. Build handcrafted rules is laborious, and not portable. Traditional machine learning models like maximum entropy Markov models (MEMMs) [16] and conditional random fields (CRF) [11] have a good performance in NER task. But these methods often require feature engineering, which is troublesome. Compared to these models, the neural models handle the NER task more effectively in an end-to-end mode. The use of neural network structure to solve the problem of NER can be traced back to 2003, when Hammerton [4] first proposed a NER model based on LSTM. In 2011, Collobert et al. [2] proposed a multilayer neural network to tackle the NER task, where they use the language model to compute word embeddings and apply multiple tasks to train the model. Then, Huang et al. [7] proposed the neural architecture BiLSTM-CRF for NER, where BiLSTM is used to extract sequence representations and CRF is for decoding tags. Similarly, Ma and Hovy [15] adopt both word level and character level features to feed the proposed LSTM-CNN-CRF model. In recent years, language model pretraining methods such as BERT [3] and ELMO [18] achieved state-of-the-art performance in many NLP tasks including NER. Li et al. [12] proposed a BERT-MRC architecture for NER which recognitions entities through machine reading comprehension task and then Li et al. [13] introduce dice loss to improve the performance of BERT-MRC.

Although most of these methods achieve good performances, they usually require a lot of labelled data for training the neural network. Considering the general NER task in their works, i.e. the entities are recognized by common sense, the data can be shared across different methods.

There is an increasing need to recognize specific *entities* in specialty domain. For example, in the field of Biology, the entities that are considered for amino acids, such as Glycine, Glutamic acid, Lysine, rather than person names as the general NER task. In another specialty example of procuratorate, the extracted elements are domain specific. The considered entities are the crime suspect, procurator, court, and etc. Although the purpose of NER task for specialty domain is similar with the general NER task, the label set are completely different. Thus the pretrained neural network for the general NER task can not be directly applied to this specific task. Furthermore, there are usually not sufficient labelled data in specialty domain to train a specific neural network.

To address this challenging problem for low-resource NER in specialty domain, some works adopt the transfer learning. Wang et al. [19] propose a label-aware double transfer learning framework, where both the Bi-LSTM feature representations and the CRF parameters are transferred to the specific NER task. Lin and Lu [14] adapt the top layers of existing NER neural architecture to solve the specific task. Recently, Jia et al. [8] consider the language model as the companied task with NER model and adopt multi-task learning to tackle the NER task for a new specialty domain. As expected, these methods benefit a lot from the general NER task by transfer learning. But they are not appropriate for the Chinese NER task since they do not take advantage of the characteristics of Chinese components. The most related work is the lattice-structured LSTM model for Chinese NER, proposed by Zhang et al. [20]. It encodes a sequence of input characters as well as all potential words that match a lexicon and achieves the best results on their provided specialty domain datasets. However, in order to encode lexicon effectively, the lattice-structured LSTM model is designed with a very complex cell structure. Motivated by this work, we explore the effective use of character level and word level information of Chinese. But different with their method, we design the linear combination on each dimension of input feature vectors rather than the concatenation operation, which reduces about half parameters in the forward levels of network and makes the transfer light. Besides, our transfer part is delicately constructed. Our contribution are summarized as follows.

Firstly, to tackle the problem of NER in specialty domain, we propose a simple but effective Light Transfer NER model (LTN) that consists of the general part (LTN-G) and transfer part (LTN-T), as illustrated in Fig. 1. LTN-G is based on the combination of BiLSTM and CRF networks, which is pre-trained by a general NER dataset. For each token of the input sequence, we make full use of Chinese characteristics, including the pretrained embeddings of characters and words, as well as the information on part of speech (POS) tags. The POS embeddings are randomly initialized and updated with the network. For the light weight purpose, we combine the character and word embeddings in a linear way on each dimension instead of the concatenation operation on the whole vector as traditional methods, which can integrate the semantics and at the same time reduce the number of parameters in the forward network.

Secondly, in the transfer part, we try to reuse the knowledge learned in the general part as much as possible, from the granular elements for combining inputs, to a higher level embeddings of sentence. Thus, we take into account the hidden states of the final level of LTN-G as the inputs to the transfer part since these high-level sequences contain rich semantics and latent syntax knowledge. Similarly, the outputs of general tag sequence are embedded and are fed to the transfer part also. For the light purpose, we adopt the gated recurrent unit (GRU) in this part since it has fewer parameters compared to LSTM.

Finally, we perform experiments on real datasets and compare with other state-of-the-art methods. The results on two Chinese specialty domain datasets show that our model outperforms other baseline methods in the Chinese NER task on both effectiveness and efficiency. Even with a small quantity of labelled data, our model still has an acceptable result. Besides, it is worth mentioning that there is no need to change either the data tags or the CRF layer of LTN-G in the transfer process.

2 The Light Transfer Model

2.1 General Part

As illustrated in Fig. 1, the general part of our model LTN is based on the BiLSTM-CRF [6,7] structure where the BiLSTM encodes context information better than LSTM and the CRF has proved to be effective in NER task. It is pretrained on the general NER datasets and is fine-tuned for the specific NER task.

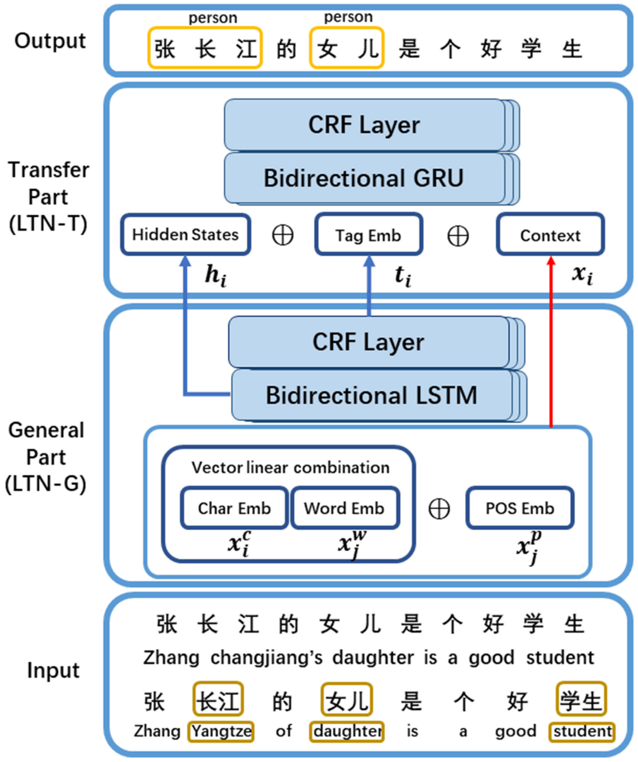
Output
person
person
张 长 江 的 女 儿 是 个 好 学 生
Transfer Part (LTN-T)
CRF Layer
Bidirectional GRU
Hidden States
Tag Emb
Context
h_i
t_i
x_i
General Part (LTN-G)
CRF Layer
Bidirectional LSTM
Vector linear combination
Char Emb
Word Emb
POS Emb
x_i^c
x_j^w
x_j^p
Input
张 长 江 的 女 儿 是 个 好 学 生
Zhang changjiang's daughter is a good student
张 长江 的 女儿 是 个 好 学生
Zhang Yangtze of daughter is a good student

Fig. 1. The light transfer NER model (LTN)

Let $\mathbf{x} = c_1, ..., c_n$ be a Chinese character sequence of input sentence, and $\mathbf{y} = y_1, ..., y_n$ be the ground truth NER tag sequence for $\mathbf{x}$. Each character c_i is mapped to vector $\mathbf{x}_i^c$ using a pretrained character embedding lookup table:

$$\mathbf{x}_i^c = \mathbf{e}^c(c_i). \tag{1}$$

After word segmentation, the input sentence can be seen as a word sequence $\mathbf{x}' = w_1, ..., w_m$. Each word w_j is mapped to vector $\mathbf{x}_j^w$ with pretrained word embeddings:

$$\mathbf{x}_j^w = \mathbf{e}^w(w_j). \tag{2}$$

If a word consists of only a single character after word segmentation, it is also regarded as a word that is pretrained with other words, denoted by $\mathbf{x}_i^w = \mathbf{e}^w(c_i)$. Each word embedding has the same dimensions with a character embedding. For clarity, we adopt $\mathbf{e}^w$ to represent the pretrained word embedding lookup table and $\mathbf{e}^c$ to represent the one for characters. The footmark i represents the i^{th} character in $\mathbf{x}$ and j represents the j^{th} word in $\mathbf{x}'$.

Different from other character and word based NER models, we use linear combination on each dimension of their embeddings which reduce the number of parameters in the forward network. In Chinese NER task, the input sequences usually involves gold word segmentation. Hence, the error brought by word segmentation will affect the network performance. To alleviate this problem, in our model, each character belongs to a word in an input sequence has two different representations after word segmentation. If a character c_i is the end of a word w_j, we use the character embedding $\mathbf{x}_i^c$ and the word embedding $\mathbf{x}_j^w$ to represent it. For the characters that are not in the end of w_j, we represent the characters with a single character word embedding $\mathbf{x}_i^w$ instead of the word embedding $\mathbf{x}_j^w$. Compared with the method that uses same word embedding for all characters in a word, our method could better encode Chinese word information and improve the performance of the NER model. Therefore, for each position c_i, the combined vector $\mathbf{x}_i^l$ is computed by:

$$\mathbf{x}_i^l = \begin{cases} \boldsymbol{\alpha}\mathbf{x}_i^c + (\mathbf{1}-\boldsymbol{\alpha})\mathbf{x}_i^w & \text{if } c_i \text{ is not the end of } w_j \\ \boldsymbol{\alpha}\mathbf{x}_i^c + (\mathbf{1}-\boldsymbol{\alpha})\mathbf{x}_j^w & \text{if } c_i \text{ is the end of } w_j \end{cases} \tag{3}$$

where $\boldsymbol{\alpha}$ is a parameter vector to balance the weights between character and word.

Furthemore, we add POS information to the representation of each character. We embed POS information with a shared POS embedding lookup table. Each word w_j has a unique POS tag p_j belonging to it. Unlike word embedding, all characters in a word have the same POS embedding. The POS embedding of each word is represented by:

$$\mathbf{x}_j^p = \mathbf{e}^p(p_j). \tag{4}$$

Here, $\mathbf{e}^p$ is the POS embedding lookup table which is randomly initialized at the beginning of the training and updates during the training.

For the input sequence $\mathbf{x} = c_1, ..., c_n$, the final input representation at cell i is:

$$\mathbf{x}_i = \mathbf{x}_i^l \oplus \mathbf{x}_i^p \tag{5}$$

where $\oplus$ represents concatenation. We apply a standard bi-directional LSTM for $\mathbf{x}$ to learn the context, where each cell accepts $\mathbf{x}_i$ and computes it with the previous hidden state for the current state $\mathbf{h}_i$:

$$\overrightarrow{\mathbf{h}}_i = LSTM(\overrightarrow{\mathbf{h}}_{i-1}, \mathbf{x}_i) \tag{6}$$

$$\overleftarrow{\mathbf{h}}_i = LSTM(\overleftarrow{\mathbf{h}}_{i+1}, \mathbf{x}_i) \tag{7}$$

$$\mathbf{h}_i = \overrightarrow{\mathbf{h}}_i \oplus \overleftarrow{\mathbf{h}}_i \tag{8}$$

The standard CRF model is used as the output layer for LTN-G. The output probability $p(\mathbf{y}|\mathbf{x})$ is computed over label sequence $\mathbf{y} = y_1, ..., y_n$:

$$p(\mathbf{y}|\mathbf{x}) = \frac{exp\{\sum_i(\mathbf{w}_{CRF}^{y_i}\mathbf{h}_i + b_{CRF}^{(y_{i-1},y_i)})\}}{\sum_{y'} exp\{\sum_i(\mathbf{w}_{CRF}^{y'_i}\mathbf{h}_i + b_{CRF}^{(y'_{i-1},y'_i)})\}} \tag{9}$$

where y' represents a possible label sequence, and $\mathbf{w}_{CRF}^{y_i}$ is a model parameter specific to y_i, and $b_{CRF}^{(y_{i-1},y_i)}$ is a bias specific to y_{i-1} and y_i. We use the first-order *Viterbi* algorithm to find the label sequence with a highest score. Given a set of labelled training data G from general domain, the sentence-level negative log-likelihood loss with L2 regularization is used to train the general part model:

$$Loss = -\sum_{(\mathbf{x},\mathbf{y})\in G} log(p(\mathbf{y}|\mathbf{x})) + \lambda\|\Theta\|^2 \tag{10}$$

where λ is the L2 regularization parameter and Θ represents the parameter set.

2.2 Transfer Part

The transfer part LTN-T is based on BiGRU-CRF structure since GRU [1] has fewer parameters and is more efficient than LSTM. We takes the output tag sequence and hidden states of LTN-G as the inputs to the transfer part. For an input sequence $\mathbf{x} = c_1, ..., c_n$, let $\mathbf{l} = l_1, ..., l_n$ denote its output tag sequence of by LTN-G. Each tag l_i is mapped to an embedding $\mathbf{x}_i^t$ represented as:

$$\mathbf{x}_i^t = \mathbf{e}^t(l_i). \tag{11}$$

Here $\mathbf{e}^t$ represents a shared tag embedding lookup table. So the input sequence $\mathbf{t} = t_1, ..., t_n$ of LTN-T is computed by:

$$\mathbf{t}_i = \mathbf{x}_i \oplus \mathbf{x}_i^t \oplus \mathbf{h}_i \tag{12}$$

where $\mathbf{h}_i$ represents output of general part BiLSTM at step i and $\mathbf{x}_i$ is the input representation from the general part. We apply a standard bi-directional GRU layer for the input T, where the computation at each cell i is as follows.

$$\overrightarrow{\mathbf{h}}_i = GRU(\overrightarrow{\mathbf{h}}_{i-1}, \mathbf{t}_i) \tag{13}$$

$$\overleftarrow{\mathbf{h}}_i = GRU(\overleftarrow{\mathbf{h}}_{i+1}, \mathbf{t}_i) \tag{14}$$

$$\mathbf{h}_i = \overrightarrow{\mathbf{h}}_i \oplus \overleftarrow{\mathbf{h}}_i \tag{15}$$

Similar with the general part, we apply the standard CRF as the output of transfer part and use the first-order *Viterbi* algorithm to compute the highest scored label sequence. The loss function is the same with Eq. 10.

2.3 Training Process

We first train the general part LTN-G with the labelled data for the general NER task. After it performs well in the general task, it can be reused for NER tasks in different domains. When we train the network for a specific NER task, there is no need to change the tag set of the CRF layer in LTN-G. The new labels for the specific task are only applied to the CRF layer of the transfer part. The whole process is presented in Algorithm 1. When applying it to different specialty domains, we can reuse the parameters of the general part, which makes transfer model light.

Algorithm 1. Transfer learning

```
Require: General domain data $G$ and tag set $T^G$; Specialty domain data $S$ and tag
    set $T^S$
Ensure: Transfered model for target domain
 1: General part training:
 2: while training steps not end do
 3:     split training data into minibatches $G_m \subset G$:
 4:     $\Delta h^{LSTM}, \Delta w^{CRF}, \Delta\alpha ... \leftarrow train(G_m)$
 5:     update general part parameters $\Theta$
 6: end while
 7: load the parameters $\Theta$ of LTN-G
 8: Whole model training:
 9: while training steps not end do
10:     split training data into minibatches $S_m \subset S$:
11:     $\Delta h^{GRU}, \Delta h^{LSTM}, \Delta w^{CRF}, \Delta\alpha ... \leftarrow train(S_m)$
12:     update LTN parameters
13: end while
```

3 Experiments

3.1 Datasets and Experiment Settings

We adopt three chinese datasets in different domains, **MSRA** NER [9], **Weibo** NER [17] and **Resume** NER [20], the statistics of which are shown in Table 1.

MSRA [9]**:** It is a news domain dataset which contains three types of entities: PER (person), LOC (location), and ORG (organization). **MSRA** is used as shared task on SIGNAN backoff 2006. Since this dataset contains sufficient data and is widely adopted in the Chinese NER research works, we select it to pretrain LTN and other comparison models for the general NER task.

Weibo [17]**:** It is a low-source social media domain dataset from Sina Weibo which contains four named entities: PER.NAM (person), LOC.NAM (location), ORG.NAM (organization) and GPE.NAM(geo-political). It also contains four nominal entities:PER.N0M (person), LOC.N0M (location), ORG.N0M (organization) and GPE.N0M(geo-political). There are five types of entities different from **MSRA**.

Resume [20]**:** It is a resume domain dataset annotated by Zhang et al. [20]. It contains eight types of entities: NAME (name), LOC (location), ORG (organization), CONT (Nationality), RACE (race), TITLE (title), EDU (education) and PRO (profession). There are five types of entities different from **MSRA**.

Comparatively, both **Weibo** and **Resume** contain different entities. Thus, they are used for the specific NER task. It is worth mentioning that the labelled entities in **Weibo** are more sparse than **Resume**.

We implement the algorithm in Pytorch. The sizes of embeddings of character, word and part-of-speech are set 50. Dropout rate is set 0.5. We use stochastic gradient descent(SGD) for optimization and the learning rate is set to 0.01.To ensure the fairness of the experiment, we use the same pre-trained embedding to initialize the embedding layer of all models.

Table 1. Datasets statistics

Dataset	Type	Train	Dev	Test	Entity/Sentence
MSRA	Sentence	46.4k	–	4.4k	1.58
	Char	2169.9k	–	172.6k	
	Entity	73.3k	–	4.3k	
Resume	Sentence	3.8k	0.46k	0.48k	3.54
	Char	124.1k	13.9k	15.1k	
	Entity	13.438k	1.63k	1.497k	
Weibo	Sentence	1.4k	0.27k	0.27k	1.35
	Char	73.8k	14.5k	14.8k	
	Entity	1.885k	0.414k	0.389k	

3.2 Comparison Methods and Metrics

The comparison models in this paper are as follows, where the precision (P), recall (R) and F1-score (F1) are adopted as the evaluation metrics. CharLSTM was selected as the baseline model to test the improvement brought by semantic information embedding, such as POS and word embedding.

1. **CharLSTM:** the BiLSTM-CRF structure network with the character embeddings as input. To compare the effects of transfer learning, we introduce a variant **CharLSTM-T** that is pretrained for the general NER and transferred for the specific NER by replacing the CRF layer of **CharLSTM**.

Table 2. Comparison results on **Weibo**

Models	Precision	Recall	F1
CharLSTM	58.77	48.55	53.17
Lattice	67.61	51.93	58.74
LTN-GC	**69.16**	51.45	59.00
LTN-G	68.47	**51.93**	**59.07**
La-DTL*	–	–	57.74
CharLSTM-T	62.35	48.79	54.74
Lattice-T	67.69	53.14	59.54
LTN	**69.45**	**58.21**	**63.34**

*indicates the result reported in the corresponding reference.

Table 3. Comparison results on **Resume**

Models	Precision	Recall	F1
CharLSTM	92.69	92.64	92.67
Lattice	**94.81**	94.11	94.46
LTN-GC	94.45	94.05	94.25
LTN-G	94.37	**94.60**	**94.49**
CharLSTM-T	93.89	94.36	94.12
Lattice-T	94.67	94.85	94.76
LTN	**94.87**	**95.40**	**95.14**

2. Lattice [20]**:** the state-of-the-art Chinese NER model that achieves the best results on **Weibo** and **Resume**. To compare the effects of transfer learning, we also introduce a variant **Lattice-T** that is transferred for the specific NER on its CRF layer.
3. La-DTL [19]**:** a transfer learning model for cross-specialty NER which conducts both feature representation transfer and parameter transfer with label-aware constraints.
4. LTN and Variants: LTN is our transfer model. The general part of LTN named LTN-G and the transfer part of LTN named LTN-T. In addition, to verify the effect of linear combination between character embedding and word embedding in the general part, we design the model **LTN-GC** to replace the linear combination with concatenation operation.

3.3 Experiment Results

Effectiveness. We first verify the performance of our model comparing with other models and present the results in Table 2 and 3, respectively. The models in the upper part of each table are trained directly on the dataset for specialty domain without any transfer learning, while the lower part contains the models with transfer learning. The models in the lower part of each table are first pretrained on **MSRA**, and then transfered to **Weibo** and **Resume**.

On both datasets, our model **LTN** achieves the best results on F1-score. It is worth mentioning that the result of our model general part (LTN-G) on two datasets is similar with Lattice without transfer learning. Comparing the variants of our model, LTN-G and LTN-GC achieve similar results on both two datasets. This illustrates that the linear combination of the input vectors in LTN-G reduces the parameters of network without loss of accuracy.

The results in the lower part of Table 2 and 3 show that our transfer model outperforms other methods. This illustrates that the adoption of contextual information learned in the general domain benefits the transfer process, which is better than simply adapting pre-training weights to a specialty domain. On **Weibo**, our model LTN also performs better than La-DTL. Although La-DTL uses more sophisticated techniques for the transfer learning approach, it does not leverage the semantic information of the general domain more effectively than LTN.

Comparing the results on two datasets, we can see that the performance on **Resume** is much better that **Weibo**. This is because **Resume** contain more labelled data than **Weibo** such that models can be trained well without transfer learning. Our model **LTN** improves a lot comparing with baseline methods, which convinces the performance of transfer learning and illustrates the effectiveness of reusing knowledge from the general task.

Table 4. F1-score against different data size on **Weibo**

Model	10% data	25% data	50% data	100% data
LTN	45.02	51.41	62.57	63.34
Lattice	28.39	39.45	53.96	58.74
CharLSTM	22.71	32.48	48.16	53.17

Table 5. Model processing speed on **Weibo**

Model	Train (sent/sec)	Test (sent/sec)
LTN	19.37	67.87
Lattice	8.10	24.32
CharLSTM	23.98	135.45

Robustness. Then we quantify the robustness of models under insufficient data, we conducted the experiments against different proportion of data on **Weibo**. **Lattice** and **CharLSTM** are trained directly on **Weibo** without transfer learning. Table 4 shows the F1-score of models trained with **Weibo** of different dataset size, namely 10%, 25%, 50%, and 100%, respectively.

The results in Table 4 show our model performs much better than **Lattice** and **CharLSTM** on all cases. Specially, even in the case of 10%, our model still achieves an acceptable result. This convinces that the performance benefits from transfer learning, while **Lattice** and **CharLSTM** model are affected much by the dataset size. It is worth mentioning that our model needs only half of the data to achieve the equivalent result of Lattice.

The results on the robustness of LTN illustrate that the proposed transfer learning based method can be adapted to a new area without much labelled data. This attributes to that the knowledge that are learned in the general NER are reused in the domain specific NER task.

Efficiency. Since a light model should process data fast in either training or testing, we compare the proposed model with others on this metric. Table 5 shows the processing speed in terms of sentences per minute at the same environment. The results on both training and test processes show that our model **LTN** is much faster than **Lattice**. **Lattice** is designed in order to better use lexicon information and the complicated network structure leads to the slow speed of training and testing. Although the **CharLSTM** model with the simplest network structure is faster than our model, together considering its performance, our model is light and with the best performance.

4 Conclusion

We propose a light and effective transfer model for the NER task for specialty domain. By designing an additional part over a general NER network, we reuse the knowledge learned in the general NER task as much as possible, from the granular elements for combining inputs, to a higher level embeddings of outputs. At the same time the model can be easily adapted to the domain specific NER task without reconstruction. We compare our model with related works on real datasets. The experimental results show that our model is consistently superior than baseline methods on both effectiveness and efficiency, especially in case of low-resource data for specialty domain.

Acknowledgement. This work was supported by the National Key R&D Program of China (2018YFC0831401), the Key R&D Program of Shandong Province (2019JZZY010107), the National Natural Science Foundation of China (91646119), the Major Project of NSF Shandong Province (ZR2018ZB0420), and the Key R&D Program of Shandong province (2017GGX10114). The scientific calculations in this paper have been done on the HPC Cloud Platform of Shandong University.

References

1. Chung, J., Gülçehre, Ç., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. CoRR abs/1412.3555 (2014)
2. Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., Kuksa, P.: Natural language processing (almost) from scratch. JMLR **12**, 2493–2537 (2011)
3. Devlin, J., Chang, M., Lee, K., Toutanova, K.: BERT: pre-training of deep bidirectional transformers for language understanding. CoRR abs/1810.04805 (2018)
4. Hammerton, J.: Named entity recognition with long short-term memory. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, CONLL 2003, vol. 4, pp. 172–175. Association for Computational Linguistics, USA (2003)
5. Hanisch, D., Fundel, K., Mevissen, H.T., Zimmer, R., Fluck, J.: Prominer: rule-based protein and gene entity recognition. BMC Bioinform. **6**, S14 (2005)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
7. Huang, Z., Xu, W., Yu, K.: Bidirectional LSTM-CRF models for sequence tagging. CoRR abs/1508.01991 (2015)
8. Jia, C., Liang, X., Zhang, Y.: Cross-domain NER using cross-domain language modeling. In: ACL, pp. 2464–2474 (2019)
9. Jin, G., Chen, X.: The fourth international Chinese language processing bakeoff: Chinese word segmentation, named entity recognition and Chinese POS tagging. In: Proceedings of the Sixth SIGHAN Workshop on Chinese Language Processing (2008)
10. Kim, J.H., Woodl, P.: A rule-based named entity recognition system for speech input. In: ICSLP (2000)
11. Lafferty, J.D., McCallum, A., Pereira, F.C.N.: Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: ICML, pp. 282–289 (2001)
12. Li, X., Feng, J., Meng, Y., Han, Q., Wu, F., Li, J.: A unified MRC framework for named entity recognition. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 5849–5859. Association for Computational Linguistics (2020)
13. Li, X., Sun, X., Meng, Y., Liang, J., Wu, F., Li, J.: Dice loss for data-imbalanced NLP tasks. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 465–476. Association for Computational Linguistics (2020)
14. Lin, B.Y., Lu, W.: Neural adaptation layers for cross-domain named entity recognition. In: EMNLP, pp. 2012–2022 (2018)
15. Ma, X., Hovy, E.: End-to-end sequence labeling via bi-directional LSTM-CNNS-CRF (2016). arXiv preprint arXiv:1603.01354
16. McCallum, A., Freitag, D., Pereira, F.C.N.: Maximum entropy Markov models for information extraction and segmentation. In: ICML, pp. 591–598 (2000)
17. Peng, N., Dredze, M.: Named entity recognition for Chinese social media with jointly trained embeddings. In: EMNLP, pp. 548–554 (2015)
18. Peters, M.E., et al.: Deep contextualized word representations. CoRR abs/1802.05365 (2018)
19. Wang, Z., et al.: Label-aware double transfer learning for cross-specialty medical named entity recognition. In: NAACL, pp. 1–15 (2018)
20. Zhang, Y., Yang, J.: Chinese NER using lattice LSTM. In: ACL, pp. 1554–1564 (2018)