Modularized Interaction Network for Named Entity Recognition

Fei Li[1], **Zheng Wang**[2*], **Siu Cheung Hui**[2], **Lejian Liao**[1], **Dandan Song**[1*], **Jing Xu**[1], **Guoxiu He**[3], **Meihuizi Jia**[1]

[1]Beijing Institute of Technology, China
[2]Nanyang Technological University, Singapore
[3]Wuhan University, China

{lifei926,liaolj,sdd,xujing,jmhuizi24}@bit.edu.cn
{wang_zheng,asschui}@ntu.edu.sg, guoxiu.he@whu.edu.cn

Abstract

Although the existing Named Entity Recognition (NER) models have achieved promising performance, they suffer from certain drawbacks. The sequence labeling-based NER models do not perform well in recognizing long entities as they focus only on word-level information, while the segment-based NER models which focus on processing segment instead of single word are unable to capture the word-level dependencies within the segment. Moreover, as boundary detection and type prediction may cooperate with each other for the NER task, it is also important for the two sub-tasks to mutually reinforce each other by sharing their information. In this paper, we propose a novel Modularized Interaction Network (MIN) model which utilizes both segment-level information and word-level dependencies, and incorporates an interaction mechanism to support information sharing between boundary detection and type prediction to enhance the performance for the NER task. We have conducted extensive experiments based on three NER benchmark datasets. The performance results have shown that the proposed MIN model has outperformed the current state-of-the-art models.

1 Introduction

Named Entity Recognition (NER) is one of the fundamental tasks in natural language processing (NLP) that intends to find and classify the type of a named entity in text such as person (PER), location (LOC) or organization (ORG). It has been widely used for many downstream applications such as relation extraction (Xiong et al., 2018), entity linking (Gupta et al., 2017), question generation (Zhou et al., 2017) and coreference resolution (Barhom et al., 2019).

Currently, there are two types of methods for the NER task. The first one is sequence labeling-based methods (Lample et al., 2016; Chiu and Nichols, 2016; Luo et al., 2020), in which each word in a sentence is assigned a special label (e.g., B-PER or I-PER). Such methods can capture the dependencies between adjacent word-level labels and maximize the probability of predicted labels over the whole sentence. It has achieved the state-of-the-art performance in various datasets over the years. However, NER is a segment-level recognition task. As such, the sequence labeling-based models which focus only on word-level information do not perform well especially in recognizing long entities (Ye and Ling, 2018). Recently, segment-based methods (Kong et al., 2016; Li et al., 2020b; Yu et al., 2020b; Li et al., 2021) have gained popularity for the NER task. They process segment (i.e., a span of words) instead of single word as the basic unit and assign a special label (e.g., PER, ORG or LOC) to each segment. As these methods adopt segment-level processing, they are capable of recognizing long entities. However, the word-level dependencies within a segment are usually ignored.

NER aims at detecting the entity boundaries and the type of a named entity in text. As such, the NER task generally contains two separate and independent sub-tasks on boundary detection and type prediction. However, from our experiments, we observe that the boundary detection and type prediction sub-tasks are actually correlated. In other words, the two sub-tasks can interact and mutually reinforce each other by sharing their information. Consider the following example sentence: “Emmy Rossum was from New York University”. If we know “University” is an entity boundary, it will be more accurate to predict the corresponding entity type to be “ORG”. Similarly, if we know an entity has an “ORG” type, it will be more accurate to predict that “University” is the end boundary of

[*] Corresponding authors.

the entity “New York University” instead of “York” (which is the end boundary for the entity “New York”). However, sequence labeling-based models consider the boundary and type as labels, and thus such information cannot be shared between the sub-tasks to improve the accuracy. On the other hand, segment-based models first detect the segments and then classify them into the corresponding types. These methods generally cannot use entity type information in the process of segment detection and may have errors when passing such information from segment detection to segment classification.

In this paper, we propose a Modularized Interaction Network (MIN) model which consists of the NER Module, Boundary Module, Type Module and Interaction Mechanism for the NER task. To tackle the issue on recognizing long entities in sequence labeling-based models and the issue of utilizing word-level dependencies within a segment in segment-based models, we incorporate a pointer network (Vinyals et al., 2015) into the Boundary Module as the decoder to capture segment-level information on each word. Then, these segment-level information and the corresponding word-level information on each word are concatenated as the input to the sequence labeling-based models.

To enable interaction information, we propose to separate the NER task into the boundary detection and type prediction sub-tasks to enhance the performance of the two sub-tasks by sharing the information from each sub-task. Specifically, we use two different encoders to extract their distinct contextual representations from the two sub-tasks and propose an Interaction Mechanism to mutually reinforce each other. Finally, these information are fused into the NER Module to enhance the performance. In addition, the NER Module, Boundary Module and Type Module share the same word representations and we apply multitask training when training the proposed MIN model.

In summary, the main contributions of this paper include:

- We propose a novel Modularized Interaction Network (MIN) model which utilizes both the segment-level information from segment-based models and word-level dependencies from sequence labeling-based models in order to enhance the performance of the NER task.

- The proposed MIN model consists of the NER Module, Boundary Module, Type Module and Interaction Mechanism. We propose to separate boundary detection and type prediction into two sub-tasks and the Interaction Mechanism is incorporated to enable information sharing between the two sub-tasks to achieve the state-of-the-art performance.

- We conduct extensive experiments on three NER benchmark datasets, namely CoNLL2003, WNUT2017 and JNLPBA, to evaluate the performance of the proposed MIN model. The experimental results have shown that our MIN model has achieved the state-of-the-art performance and outperforms the existing neural-based NER models.

2 Related Work

In this section, we review the related work on the current approaches for Named Entity Recognition (NER). These approaches can be categorized into sequence labeling-based NER and segment-based NER.

2.1 Sequence Labeling-based NER

Sequence labeling-based NER is regarded as a sequence labeling task, where each word in a sentence is assigned a special label (e.g., B-PER, I-PER). Huang et al. (Huang et al., 2015) utilized the BiLSTM as an encoder to learn the contextual representation of words, and then Conditional Random Fields (CRFs) was used as a decoder to label the words. It has achieved the state-of-the-art results on various datasets for the past many years. Inspired by the success of the BiLSTM-CRF architecture, many other state-of-the-art models have adopted such architecture. Chiu and Nichols (Chiu and Nichols, 2016) used Convolutional Neural Network (CNN) to capture spelling features, and the character-level and word-level embeddings are concatenated as the input of BiLSTM with CRF network. Further, Lample et al. (Lample et al., 2016) proposed RNN-BiLSTM-CRF as an alternative. More recently, pretrained language models such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019) have been adopted to further enhance the performance of NER.

2.2 Segment-based NER

Segment-based NER identifies segments in a sentence and classifies each segment with a special label (e.g., PER, ORG or LOC). Kong et al. (Kong et al., 2016) used BiLSTM to map arbitrary-length segment into a fixed-length vector, and then these vectors were passed to Semi-Markov Conditional Random Fields (Semi-CRFs) for labeling the segments. Zhuo et al. (Zhuo et al., 2016) adopted a gated recursive Convolutional Neural Network instead of BiLSTM to build a pyramid-like structure for extracting segment-level features in a hierarchical way. In recent years, Ye et al. (Ye and Ling, 2018) exploited the weighted sum of word-level within segment to learn segment-level features with Semi-CRFs which was then trained jointly on word-level with the BiLSTM-CRF network. Li et al. (Li et al., 2020a) used a recurrent neural network encoder-decoder framework with a pointer network to detect entity segments. Li et al. (Li et al., 2020b) treated NER as a machine reading comprehension (MRC) task, where entities were extracted as retrieved answer spans. Yu et al. (Yu et al., 2020b) ranked all the spans in terms of the pairs of start and end tokens in a sentence using a biaffine model.

3 Proposed Model

This section presents our proposed Modularized Interaction Network (MIN) for NER. The overall model architecture is shown in Figure 1(a), which consists of the NER Module, Boundary Module, Type Module and Interaction Mechanism.

3.1 NER Module

In the NER Module, we adopt the RNN-BiLSTM-CRF model (Lample et al., 2016) as our backbone, which consists of three components: word representation, BiLSTM encoder and CRF decoder.

Word Representation Given an input sentence $S =< w_1, w_2, \cdots, w_n >$, each word $w_i (1 \leq i \leq n)$ is represented by concatenating a word-level embedding x_i^w and a character-level word embedding x_i^c as follows:

$$x_i = [x_i^w; x_i^c] \tag{1}$$

where x_i^w is the pre-trained word embedding, and the character-level word embedding x_i^c is obtained with a BiLSTM to capture the orthographic and morphological information. It considers each character in the word as a vector, and then inputs them to a BiLSTM to learn the hidden states. The final hidden states from the forward and backward outputs are concatenated as the character-level word information.

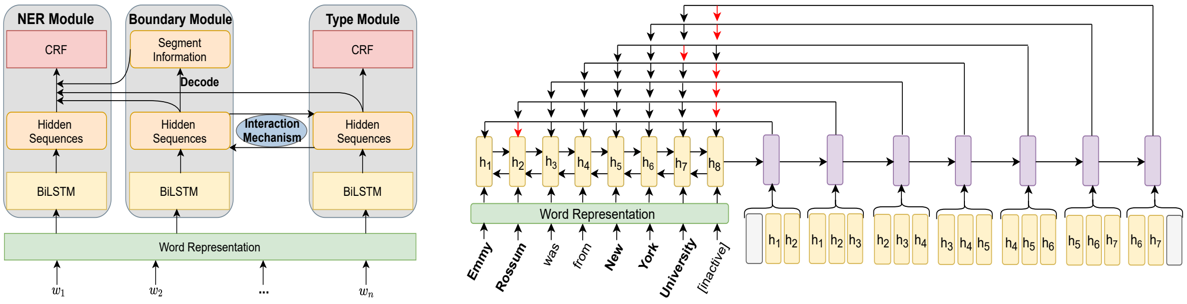
BiLSTM Encoder The distributed word embeddings $X =< x_1, x_2, \cdots, x_n >$ are then fed into the BiLSTM encoder to extract the hidden sequences $H =< h_1, h_2, \cdots, h_n >$ of all words as follows:

$$\begin{aligned} h_i &= \left[\overrightarrow{h_i}; \overleftarrow{h_i}\right] \\ \overrightarrow{h_i} &= LSTM\left(x_i, \overrightarrow{h_{i-1}}\right) \\ \overleftarrow{h_i} &= LSTM\left(x_i, \overleftarrow{h_{i-1}}\right) \end{aligned} \tag{2}$$

In the NER Module, we fuse the distinct contextual boundary representation and type representation for the NER task. In addition, we also fuse the segment information from the Boundary Module to support the recognition of long entities. Note that the boundary information and type information can mutually reinforce each other. Thus, we use an interaction mechanism to reinforce them before fusing these information in the NER Module. Instead of directly concatenating these information with hidden representations in the NER module, we follow the previous studies (Zhang et al., 2018; Yu et al., 2020a) to use a gate function to dynamically control the amount of information flowing by infusing the expedient part while excluding the irrelevant part. The gate function uses the information from the NER Module to guide the process, which is described formally as follows:

$$\begin{aligned} \overline{H}^{Bdy}, \overline{H}^{Type} &= interact(H^{Bdy}, H^{Type}) \\ H^B &= \sigma\left(W_1^\top H + W_B^\top \overline{H}^{Bdy}\right) \otimes \overline{H}^{Bdy} \\ H^T &= \sigma\left(W_2^\top H + W_T^\top \overline{H}^{Type}\right) \otimes \overline{H}^{Type} \\ H^S &= \sigma\left(W_3^\top H + W_S^\top H^{Seg}\right) \otimes H^{Seg} \end{aligned} \tag{3}$$

where H^{Bdy} and H^{Type} represent the distinct representations of hidden sequences from the Boundary Module and Type Module respectively, and H^{Seg} represents the segment information from the Boundary Module. We will discuss them in Section 3.2 and Section 3.3. $\overline{H}^{Bdy}$ and $\overline{H}^{Type}$ represent the distinct representations of hidden sequences from the Boundary Module and Type Module respectively after the interaction using an interaction mechanism $interact(\cdot, \cdot)$, and we will discuss them in Section 3.4. H^B, H^T and H^S represent the boundary, type and segment information respectively to be injected into the NER Module from the gate function. σ denotes the logistic



(a) Overall Architecture

(b) Boundary Module

Figure 1: The architecture of our proposed Modularized Interaction Network.

sigmoid function and $\otimes$ denotes the element-wise multiplication.

The final hidden representations in the NER Module are as follows:

$$H^{NER} = W^{\top}[H; H^{B}; H^{T}; H^{S}] + b \quad (4)$$

CRF Decoder CRF has been widely used in the state-of-the-art NER models (Chiu and Nichols, 2016; Lample et al., 2016) to model tagging decisions when considering strong connections between output tags. For an input sentence $S =< w_1, w_2, \cdots, w_n >$, the score of a predicted sequence of labels $y =< y_1, y_2, \cdots, y_n >$ is defined as follows:

$$sc\left(S, y\right) = \sum_{i=0}^{n} T_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i} \quad (5)$$

where $T_{y_i, y_{i+1}}$ represents the score of a transition from y_i to y_{i+1}, and P_{i,y_i} is the score of the y_i tag of the i^{th} word in a sentence.

The CRF model describes the probability of predicted labels y over all possible tag sequences in the set Y, that is:

$$p\left(y|S\right) = \frac{e^{sc(S,y)}}{\sum_{\widetilde{y}\in Y} e^{sc(S,\widetilde{y})}} \quad (6)$$

We maximize the log-probability of the correct sequence of labels during the training. During decoding, we predict the label sequence with the maximum score:

$$y^{*} = \underset{\widetilde{y}\in Y}{\arg\max}\, sc\left(S, \widetilde{y}\right) \quad (7)$$

3.2 Boundary Module

The Boundary Module needs to provide not only distinct contextual boundary information but also segment information for the NER Module. Here, we use another BiLSTM as encoder to extract distinct contextual boundary information. And inspired by BDRYBOT (Li et al., 2020a), a recurrent neural network encoder-decoder framework with a pointer network is used to detect entity segments for segment information. The BDRYBOT model processes the starting boundary word in an entity to point to the corresponding ending boundary word. The other entity words in the entity are skipped. The non-entity words are pointed to a specific position. This method has achieved promising results in the boundary detection task. However, due to the variable length of entities, this model is deprived of the power of batch training. In addition, as the segment information of each word in an entity is the same as the starting boundary word, the segment information for all the words within a segment will be incorrect if the starting boundary word is detected wrongly. To avoid this problem, we improve the training process and propose a novel method to capture the segment information of each word.

We train the starting boundary word to point to the corresponding ending boundary word, and the other words in the sentence to a sentinel word *inactive*. The process is shown in Figure 1(b). Specifically, we use another BiLSTM as encoder to obtain the distinct boundary hidden sequences $H^{Bdy} =< h_1^{Bdy}, h_2^{Bdy}, \cdots, h_n^{Bdy} >$, and a sentinel vector is padded into the last positions of hidden sequences H^{Bdy} for the sentinel word *inactive*. Then, a unidirectional LSTM is used as a decoder to generate the decoded state d_j at each time step j. To add extra information to the input of the LSTM, we follow (Fernández-González and Gómez-Rodríguez, 2020) and use the sum of the hidden states of current (h_i^{Bdy}), previous (h_{i-1}^{Bdy}) and next (h_{i+1}^{Bdy}) words instead of word embedding as the input to the decoder as follows:

$$\begin{aligned} s_j &= h_{j-1}^{Bdy} + h_j^{Bdy} + h_{j+1}^{Bdy} \\ d_j &= LSTM\left(s_j, d_{j-1}\right) \end{aligned} \tag{8}$$

Note that the first word and last word do not have hidden states of previous and next, we use zero vectors to represent it which are shown as grey blocks in Figure 1(b).

After that, we use the biaffine attention mechanism (Dozat and Manning, 2017) to generate a feature representation for each possible boundary position i at time step j, and the $Softmax$ function is used to obtain the probability of word w_i for determining an entity segment that starts with word w_j and ends with word w_i.

$$\begin{aligned} u_i^j &= {d_j}^T W h_i^{Bdy} + U^T d_j + V^T h_i^{Bdy} + b \\ p\left(w_i | w_j\right) &= Softmax\left(u_i^j\right), i \in [j, n+1] \end{aligned} \tag{9}$$

where W is the weight matrix of bi-linear term, U and V are the weight matrices of linear terms, b is the bias vector and $i \in [j, n+1]$ indicates a possible position in decoding.

Different from the existing methods (Zhuo et al., 2016; Sohrab and Miwa, 2018) that enumerate all segments starting with word w_j with equal importance, we use the probability $p\left(w_i|w_j\right)$ as the confidence of the segment that starts with word w_j and ends with word w_i, and then all these segments under the probability $p\left(w_i|w_j\right)$ are summed as the segment information of word w_j.

$$\begin{gathered} H_j^{Seg} = \sum_{i=j}^{n} p\left(w_i|w_j\right) h_{j,i}^p \\ h_{j,i}^p = [h_j^{Bdy}; h_i^{Bdy}; h_i^{Bdy} - h_j^{Bdy}; h_i^{Bdy} \odot h_j^{Bdy}] \end{gathered} \tag{10}$$

where $h_{j,i}^p$ is the representation of the segment that starts with word w_j and ends with word w_i, and $\odot$ is element-wise product.

3.3 Type Module

For the Type Module, we use the same network structure as in the NER Module. Given the shared input $X =< x_1, x_2, \cdots, x_n >$, BiLSTM is used to extract distinct contextual type information $H^{Type} =< h_1^{Type}, h_2^{Type}, \cdots, h_n^{Type} >$, and then CRF is used to tag type labels.

3.4 Interaction Mechanism

As discussed in Section 1, the boundary information and type information can mutually reinforce each other. We first follow (Cui and Zhang, 2019; Qin et al., 2021) and use a self-attention mechanism over each sub-task labels to obtain the explicit label representations. Then, we concatenate these representations and contextual information of corresponding sub-tasks to get label-enhanced contextual information. For the i^{th} label-enhanced boundary contextual representation h_i^{B-E}, we first use the biaffine attention mechanism (Dozat and Manning, 2017) to grasp the attention scores between h_i^{B-E} and the label-enhanced type contextual information $< h_1^{T-E}, h_2^{T-E}, \cdots, h_n^{T-E} >$. The attention scores $< \alpha_{i,1}^{B-E}, \alpha_{i,2}^{B-E}, \cdots, \alpha_{i,n}^{B-E} >$ are computed in the same way as in Equation (9). Then, we concatenate the i^{th} label-enhanced boundary representation h_i^{B-E} and the interaction representation r_i^{B-E} by considering the type information as its updated boundary representation:

$$\begin{aligned} r_i^{B-E} &= \sum_{j=1}^{n} \alpha_{i,j}^{B-E} h_j^{T-E} \\ \overline{h}_i^{Bdy} &= [h_i^{B-E}, r_i^{B-E}] \end{aligned} \tag{11}$$

Similarity, we can obtain the updated type representation $\overline{h}_i^{Type}$ by considering the boundary information.

3.5 Joint Training

There are three modules in our proposed MIN model: NER Module, Boundary Module and Type Module. They share the same word representations. Thus, the whole model can be trained with multitask training. During training, we minimize the negative log-probability of the correct sequence of labels in Equation (6) for the NER Module and Type Module, while the cross-entropy loss is used for the Boundary Module:

$$\begin{aligned} \mathcal{L}^{NER} &= -\log\left(p\left(\hat{y}^{NER}|X\right)\right) \\ \mathcal{L}^{Type} &= -\log\left(p\left(\hat{y}^{Type}|X\right)\right) \\ \mathcal{L}^{Bdy} &= -\frac{1}{n}\sum_{i=1}^{n} \hat{y}_i^{Bdy} \log p_i^{Bdy} \end{aligned} \tag{12}$$

where X represents input sequence, and $\hat{y}^{NER}$ and $\hat{y}^{Type}$ represent the correct sequence of labels for the NER Module and Type Module respectively. p_i^{Bdy} is the probability distribution of the gold label and $\hat{y}_i^{Bdy}$ is the gold one-hot vector for the

Boundary Module. Then, the final multitask loss is a weighted sum of the three losses:

$$\mathcal{L} = \mathcal{L}^{NER} + \mathcal{L}^{Type} + \mathcal{L}^{Bdy} \tag{13}$$

4 Experiments

In this section, we first introduce the datasets, baseline models and implementation details. Then, we present the experimental results on three benchmark datasets. Moreover, an ablation study is also conducted. Finally, we give some insights on further analysis.

4.1 Datasets

We evaluate the proposed model on three benchmark NER datasets: CoNLL2003 (Sang and De Meulder, 2003), WNUT2017 (Derczynski et al., 2017) and JNLPBA (Kim et al., 2004).

- CoNLL2003 - It is collected from Reuters news articles. Four different types of named entities including *PER*, *LOC*, *ORG* and *MISC* are defined by the CoNLL 2003 NER shared task.

- WNUT2017 - It is a set of noisy user-generated text including YouTube comments, StackExchange posts, Twitter text, and Reddit comments. Six types of entities including *PER*, *LOC*, *Group*, *Creative_work*, *Corporation* and *Product* are annotated.

- JNLPBA - It is collected from MEDLINE abstracts. Five types of entities including *DNA*, *RNA*, *protein*, *cell_line* and *cell_type* are annotated.

Table 1 presents the statistics of these datasets.

Dataset		train	dev	test
CoNLL2003	#sentences	14,987	3,466	3,684
	#entities	23,499	5,942	5,648
WNUT2017	#sentences	3,394	1,009	1,287
	#entities	3,160	1,250	1,589
JNLPBA	#sentences	16,691	1,853	3,855
	#entities	46,388	4,902	8,657

Table 1: Statistics of CoNLL2003, WNUT2017, and JNLPBA datasets.

4.2 Baseline Models

We compare the proposed MIN model with several baseline models including sequence labeling-based models and segment-based models.

The compared sequence labeling-based models include:

- CNN-BiLSTM-CRF (Chiu and Nichols, 2016) - This model utilizes CNN to capture character-level word features, and then the character-level and word-level embeddings are concatenated as the input to the BiLSTM-CRF network. It is a classical baseline for NER.

- RNN-BiLSTM-CRF (Lample et al., 2016) - This model uses RNN instead of CNN in CNN-BiLSTM-CRF.

- ELMo (Peters et al., 2018) - This model uses a deep bidirectional language model to learn contextualized word representation on a large text corpus, which is then fed into BiLSTM-CRF for NER.

- Flair (Akbik et al., 2018) - This model uses BiLSTM-CRF with character-level contextualized representations for NER.

- BERT (Devlin et al., 2019) - This model learns contextualized word representation based on a bidirectional Transformer, which is then fed into BiLSTM-CRF for NER.

- HCRA (Luo et al., 2020) - This model uses sentence-level and document-level representations to augment the contextualized representation based on a funnel-shaped CNN with BiLSTM-CRF for NER.

The compared segment-based models include:

- BiLSTM-Pointer[1] (Li et al., 2020a) - This model uses BiLSTM as the encoder and another unidirectional LSTM with pointer networks as the decoder for entity boundary detection. Then, the entity segments generated by the decoder are classified with the Softmax classifier for NER.

- HSCRF (Ye and Ling, 2018) - This model exploits the weighted sum of word-level within segment to learn segment-level features with Semi-CRFs which is then trained jointly on word-level with the BiLSTM-CRF network.

[1] In (Li et al., 2020a), the pointer networks is used for detecting entity boundaries only. We reproduce this work and add a Softmax layer for the NER task.

- MRC+BERT (Li et al., 2020b) - This model formulates the NER task as a machine reading comprehension task.

- Biaffine+BERT (Yu et al., 2020b) - This model ranks all the spans in terms of the pairs of start and end tokens in a sentence using a biaffine model.

4.3 Implementation Details

Our proposed MIN model is implemented with the PyTorch framework. We use 100-dimensional pre-trained Glove word embeddings [2] (Pennington et al., 2014). The char embeddings is initialized randomly as 25-dimensional vectors. When training the model, both of the embeddings are updated along with other parameters. We use Adam optimizer (Kingma and Ba, 2014) for training with a mini-batch. The initial learning rate is set to 0.01 and will shrunk by 5% after each epoch, dropout rate to 0.5, the hidden layer size to 100, and the gradient clipping to 5. We report the results based on the best performance on the development set. All of our experiments are conducted on the same machine with 8-cores of Intel(R) Xeon(R) E5-1630 CPU@3.70GHz and two Nvidia GeForce-GTX GPU. Following the work in (Ye and Ling, 2018), the maximum segment length for segment information discussed in Section 3.2 is set to 6 for better computational efficiency.

4.4 Experimental Results

Table 2 shows the experimental results of our proposed MIN model and the baseline models. In Table 2, when compared with models without using any language models or external knowledge, we observe that our MIN model outperforms all the compared baseline models in terms of precision, recall and F1 scores, and achieves 0.57%, 4.77% and 3.26% improvements on F1 scores for the CoNLL2003, WNUT2017 and JNLPBA datasets respectively.

Among the compared models, the F1 scores of the BiLSTM-Pointer model are generally lower than other models. This is because it does not utilize the word-level dependencies within a segment and also suffers from the problem on boundary error propagation during boundary detection and type prediction. The CNN-BiLSTM-CRF and RNN-BiLSTM-CRF models have achieved similar performance results on the three datasets, which perform worse than that of HCRA and HSCRF. The HCRA model uses sentence-level and document-level representations to augment the contextualized word representation, while the HSCRF model considers the segment-level and word-level information with multitask training. However, the HCRA model does not consider the segment-level information, and the HSCRF model does not model directly the word-level dependencies within a segment. In addition, all the above models do not share information between the boundary detection and type prediction sub-tasks. Our MIN model has achieved the best performance as it is capable of considering all these information.

When pre-trained language models such as ELMo and BERT are incorporated, all the models have achieved better performance results. In particular, we observe that our MIN model has achieved 0.95%, 3.83% and 2.73% improvements on the F1 scores for the CoNLL2003, WNUT2017 and JNLPBA datasets respectively when compared with the other models. The results are consistent with what have been discussed in models without using any pre-trained language models.

4.5 Ablation Study

To show the importance of each component in our proposed MIN model, we conduct an ablation experiment on the Boundary Module, Type Module and Interaction Mechanism. As shown in Table 3, we can see that all these components contribute significantly to the effectiveness of our MIN model.

The discussion on the effectiveness of each component is given with respect to the three datasets. The Boundary Module improves the F1 scores by 1.13%, 3.58% and 2.1% for CoNLL2003, WNUT2017 and JNLPBA respectively. This is because it not only provides segment-level information for the NER Module but also provides the boundary information for the Type Module. As such, it helps recognize long entities and predict the entity types more accurately.

The Type Module improves the F1 scores by 1.02%, 2.81% and 1.42% for CoNLL2003, WNUT2017 and JNLPBA respectively. This is because it provides the type information for the Boundary Module which can help detect entity boundaries more accurately. In addition, it can also help obtain more effective segment information.

[2] http://nlp.stanford.edu/projects/glove/

Model	CoNLL2003			WNUT2017			JNLPBA		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
CNN-BiLSTM-CRF	91.35	91.06	91.21	57.54	32.90	41.86	73.96	70.52	72.20
RNN-BiLSTM-CRF	91.12	90.76	90.94	50.86	35.50	41.81	73.08	71.56	72.31
HCRA	92.20	91.71	91.96	-	-	-	-	-	-
BiLSTM-Pointer	90.34	90.31	90.32	54.23	30.43	38.98	67.72	74.90	71.13
HSCRF	-	-	91.53	-	-	-	69.67	75.33	72.39
MIN (ours)	92.91	92.15	**92.53**	59.17	38.48	**46.63**	74.91	76.24	**75.57**
+ Language Models/External Knowledge									
ELMo	-	-	92.22	-	-	45.33	71.18	77.68	74.29
Flair	92.37	93.12	92.74	-	-	45.96	71.18	77.68	74.29
BERT	-	-	92.80	-	-	46.10	70.73	80.36	75.24
HCRA+BERT	-	-	93.37	-	-	-	-	-	-
BiLSTM-Pointer+BERT	92.02	92.45	92.23	56.82	36.87	44.72	68.56	77.32	72.68
MRC+BERT	92.33	94.61	93.04	-	-	-	-	-	-
Biaffine+BERT	93.70	93.30	93.50	-	-	-	-	-	-
MIN+BERT (ours)	94.75	94.15	**94.45**	60.54	42.48	**49.93**	75.00	81.19	**77.97**

Table 2: Experimental results on three benchmark datasets.

Model	CoNLL2003			WNUT2017			JNLPBA		
	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)	P(%)	R(%)	F1(%)
MIN	92.91	92.15	**92.53**	59.17	38.48	**46.63**	74.91	76.24	**75.57**
NER Module only	91.12	90.76	90.94	50.86	35.50	41.81	73.08	71.56	72.31
w/o Boundary Module	91.62	91.18	91.40	53.35	36.08	43.05	73.39	73.55	73.47
w/o Type Module	91.79	91.23	91.51	54.47	36.65	43.82	74.04	74.26	74.15
w/o Interaction Mechanism	92.15	91.83	91.99	56.45	37.09	44.77	74.68	75.02	74.85

Table 3: Experimental results of the ablation study of the MIN model.

The Interaction Mechanism has achieved 0.54%, 1.86% and 0.72% improvements on F1 scores for CoNLL2003, WNUT2017 and JNLPBA respectively. As it bridges the gap between the Boundary Module and Type Module for information interaction and sharing, it can help improve the performance of boundary detection and type prediction simultaneously.

Overall, the different components of the proposed model can work effectively with each other with multitask training and enable the model achieve the state-of-the-art performance for the NER task.

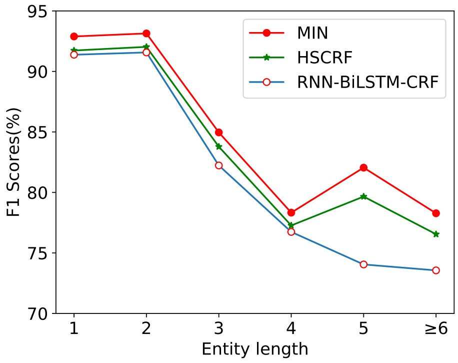


Figure 2: Performance against entity length.

4.6 Performance Against Entity Length

As our proposed MIN model is capable of recognizing long entities, we compare the performance of our MIN model with RNN-BiLSTM-CRF and HSCRF. Note that the RNN-BiLSTM-CRF model is the base model used in our MIN model. And the HSCRF model also considers the segment-level and word-level information with multitask training. The results are shown in Figure 2. The experiment is conducted on the CoNLL2003 test dataset. We follow the setting in (Ye and Ling, 2018) and group the data according to the number of entities from 1 to ≥ 6 in a sentence. We observe that our MIN model and the HSCRF model consistently outperform RNN-BiLSTM-CRF in each group. In particular, the improvement is obvious when the entity length is longer than 4 because both our MIN model and the HSCRF model consider the segment-level information. However, our MIN model performs better than the HSCRF model in each group. More specifically, when the entity length is longer than 4, our MIN model has great improvement over HSCRF. This is because the HSCRF model directly uses segment-level features with Semi-CRFs to tag the segments, which ignore word-level dependencies within the segment. In contrast, our MIN model combines segment-level information with word-level dependencies within a segment for the NER task.

5 Conclusion

In this paper, we have proposed a novel Modularized Interaction Network (MIN) model for the NER task. The proposed MIN model utilizes both segment-level information and word-level dependencies, and incorporates an interaction mechanism to support information sharing between boundary detection and type prediction to enhance the performance for the NER task. We have conducted extensive experiments on three NER benchmark datasets. The experimental results have shown that our proposed MIN model has achieved the state-of-the-art performance.

Acknowledgments

This research has been supported by the National Key R&D Program of China under Grant No. 2020AAA0106600, the National Natural Science Foundation of China under Grants No. 62062012 and 61976021, and the Ministry of Education (MoE) of Singapore under the Academic Research Fund (AcRF) Tier 1 Grant RG135/18.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Shany Barhom, Vered Shwartz, Alon Eirew, Michael Bugert, Nils Reimers, and Ido Dagan. 2019. Revisiting joint modeling of cross-document entity and event coreference resolution. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 4179–4189.

Jason PC Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Leyang Cui and Yue Zhang. 2019. Hierarchically-refined label attention network for sequence labeling. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4106–4119.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the wnut2017 shared task on novel and emerging entity recognition. In *Proceedings of the 3rd Workshop on Noisy User-generated Text*, pages 140–147.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT (1)*.

Timothy Dozat and Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Daniel Fernández-González and Carlos Gómez-Rodríguez. 2020. Discontinuous constituent parsing with pointer networks. In *Proceedings of the AAAI conference on artificial intelligence*, pages 7724–7731.

Nitish Gupta, Sameer Singh, and Dan Roth. 2017. Entity linking via joint encoding of types, descriptions, and context. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2681–2690.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Jin-Dong Kim, Tomoko Ohta, Yoshimasa Tsuruoka, Yuka Tateisi, and Nigel Collier. 2004. Introduction to the bio-entity recognition task at jnlpba. In *Proceedings of the international joint workshop on natural language processing in biomedicine and its applications*, pages 70–75. Citeseer.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Lingpeng Kong, Chris Dyer, and Noah A Smith. 2016. Segmental recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*.

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

Fei Li, Zheng Wang, Siu Cheung Hui, Lejian Liao, Dandan Song, and Jing Xu. 2021. Effective named entity recognition with boundary-aware bidirectional neural networks. In *Proceedings of The Web Conference 2021*.

Jing Li, Aixin Sun, and Yukun Ma. 2020a. Neural named entity boundary detection. *IEEE Transactions on Knowledge and Data Engineering*.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020b. A unified mrc framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859.

Ying Luo, Fengshun Xiao, and Hai Zhao. 2020. Hierarchical contextualized representation for named entity recognition. In *AAAI*, pages 8441–8448.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237.

Libo Qin, Tailu Liu, Wanxiang Che, Bingbing Kang, Sendong Zhao, and Ting Liu. 2021. A co-interactive transformer for joint slot filling and intent detection. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8193–8197.

Erik F Sang and Fien De Meulder. 2003. Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 142–147.

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

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pages 2692–2700.

Chenyan Xiong, Zhengzhong Liu, Jamie Callan, and Tie-Yan Liu. 2018. Towards better text understanding and retrieval through kernel entity salience modeling. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 575–584. ACM.

Zhixiu Ye and Zhen-Hua Ling. 2018. Hybrid semi-markov crf for neural sequence labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 235–240.

Jianfei Yu, Jing Jiang, Li Yang, and Rui Xia. 2020a. Improving multimodal named entity recognition via entity span detection with unified multimodal transformer. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3342–3352.

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

Qi Zhang, Jinlan Fu, Xiaoyu Liu, and Xuanjing Huang. 2018. Adaptive co-attention network for named entity recognition in tweets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

Qingyu Zhou, Nan Yang, Furu Wei, Chuanqi Tan, Hangbo Bao, and Ming Zhou. 2017. Neural question generation from text: A preliminary study. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 662–671. Springer.

Jingwei Zhuo, Yong Cao, Jun Zhu, Bo Zhang, and Zaiqing Nie. 2016. Segment-level sequence modeling using gated recursive semi-markov conditional random fields. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1413–1423.