A Boundary-aware Neural Model for Nested Named Entity Recognition

Changmeng Zheng[1], **Yi Cai**[1]*, **Jingyun Xu**[1], **Ho-fung Leung**[2] and **Guandong Xu**[3]
[1]School of Software Engineering, South China University of Technology, Guangzhou, China
[2]The Chinese University of Hong Kong, Hong Kong SAR, China
[3]Advanced Analytics Institute, University of Technology Sydney, Australia
sethecharm@mail.scut.edu.cn, ycai@scut.edu.cn

Abstract

In natural language processing, it is common that many entities contain other entities inside them. Most existing works on named entity recognition (NER) only deal with flat entities but ignore nested ones. We propose a boundary-aware neural model for nested NER which leverages entity boundaries to predict entity categorical labels. Our model can locate entities precisely by detecting boundaries using sequence labeling models. Based on the detected boundaries, our model utilizes the boundary-relevant regions to predict entity categorical labels, which can decrease computation cost and relieve error propagation problem in layered sequence labeling model. We introduce multitask learning to capture the dependencies of entity boundaries and their categorical labels, which helps to improve the performance of identifying entities. We conduct our experiments on nested NER datasets and the experimental results demonstrate that our model outperforms other state-of-the-art methods.

1 Introduction

Named entity recognition (NER) is a task that seeks to locate and classify named entities in unstructured texts into pre-defined categories such as person names, locations or medical codes. NER is generally treated as single-layer sequence labeling problem (Lafferty et al., 2001; Lample et al., 2016) where each token is tagged with one label. The label is composed by an entity boundary label and a categorical label. For example, a token can be tagged with $B\text{-}PER$, where B indicates the boundary of an entity and PER indicates the corresponding entity categorical label. However, when entities are nested within one another, single-layer sequence labeling models can not extract both entities simultaneously. A token contained inside many entities has more than one categorical label. Consider an example in Figure 1 from GENIA corpus (Kim et al., 2003), "Human TR Beta 1" is an $protein$ and it is also a part of a DNA "Human TR Beta 1 mRNA". Both entities contain the same token "Human". Thus the token should have two different categorical labels. In that case, assigning a single categorical label for "Human" is improper.

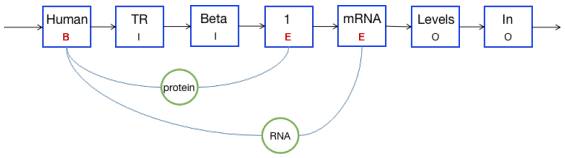

Figure 1: An example of nested entities and their boundary labels. "B" and "E" indicate the beginning and end of an entity. They are the boundary labels. "I" and "O" denote tokens inside and outside entities, respectively. $protein$ and RNA are categories of entities.

Traditional methods coping with nested entities rely on hand-craft features (Shen et al., 2003; Alex et al., 2007) and suffer from heavy feature engineering. Recent studies tackle the nested NER using neural models without relying on linguistics features or external knowledge resources. Ju et al. (2018) propose a layered sequence labeling model and Sohrab and Miwa (2018) propose a exhaustive region classification model.

- Layered sequence labeling model will first extract the inner entities (contained by other entities) and feed them into the next layer to extract outer entities. Thus, this model suffers from error propagation. When the previous layer extracts wrong entities, the performance of next layer will be affected. Moreover, when an outer entity is extracted first, the inner one will not be detected.

*Corresponding author

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 357–366, Hong Kong, China, November 3–7, 2019.

- Exhaustive region classification model enumerates all possible regions or spans in sentences to predict entities in a single layer. One issue of their method is the explicit boundary information is ignored, leading to extraction of some non-entities. **We consider an example**. In a sequence of tokens in GENIA dataset, "novel TH protein" is an entity and "a novel TH protein" is not an entity. However, since they share many tokens, the merged region representations of them are similar to each other. "novel" and "protein" are the boundary of the entity. Without the boundary information, both candidate regions are extracted as the entities.

Despite their shortcomings, layered sequence labeling model and exhaustive region classification model are complementary to each other. Therefore, we can combine them to improve the performance of nested NER. We leverage the sequence labeling model to consider the boundary information into locating entities. In the example mentioned above, "novel" is the boundary of the entity "novel TH protein", while "a" is a general token whose representation is different from "novel". With the guidance of boundary information, the model can detect "novel" as a boundary of the entity rather than token "a". We also utilize the region classification model to predict entities without considering the dependencies of inner and outer entities. In such case, Our model will not suffer error propagation problem.

In this paper, we propose a boundary-aware neural model that makes the fusion of sequence labeling model and region classification model. We apply a single-layer sequence labeling model to identify entity boundaries because the tokens in nested entities can share the same boundary labels. For example, as shown in Figure 1, "Human" can be tagged with the label B although it is the beginning of two different entities. Based on the detected entity boundaries, we predict entity categorical labels by classifying boundary-relevant regions. As shown in Figure 1, we match each token with label B to tokens with label E. The regions between them are considered as candidate entities. The representation of candidate entities will be utilized to classify categorical labels.

Our model is advanced than exhaustive region classification model in two ways: (1) we leverage the explicit boundary information to guide the model to locate and classify entities precisely. Exhaustive region classification model classifies entity regions individually, however, our model can consider the context information of boundary tokens with a sequence labeling model. That facilitates the detection of boundaries. (2) Our model only classifies the boundary-relevance regions which are much fewer than all possible regions. That decreases the time cost. Our model is advanced than layered sequence labeling model because we extract entities without distinguishing inner and outer entities.

Multitask learning is considered good at optimising the overall goal via alternatively tuning 2+ objectives, which are reinforced each other (Ruder, 2017). Considering our boundary detection module and entity categorical label prediction module share the same entity boundaries, we apply a multitask loss for training the two tasks simultaneously. The shared features of two modules are extracted by a bidirectional long short-term memory (LSTM) layer. Extensive experiments show the framework of multitask learning improves final performance in a large margin.

In summary, we make the following major contributions in this paper:

- We propose a boundary-aware neural model which leverages entity boundaries to predict categorical labels. Our model can locate entities precisely by detecting boundaries using sequence labeling models. Based on the detected boundaries, our model utilizes boundary-relevant regions to predict entity categorical labels, which can decrease computation cost and relieve error propagation problem.

- We introduce the multitask learning to capture the dependencies of entity boundaries and their categorical labels, which helps to improve the performance of identifying entities.

- We conduct our experiments on public nested NER datasets. The experimental results demonstrate our model outperforms previous state-of-the-art methods and our model is much faster in inference speed.

2 Related Work

NER has drawn the attention of NLP researchers because several downstream tasks such as entity

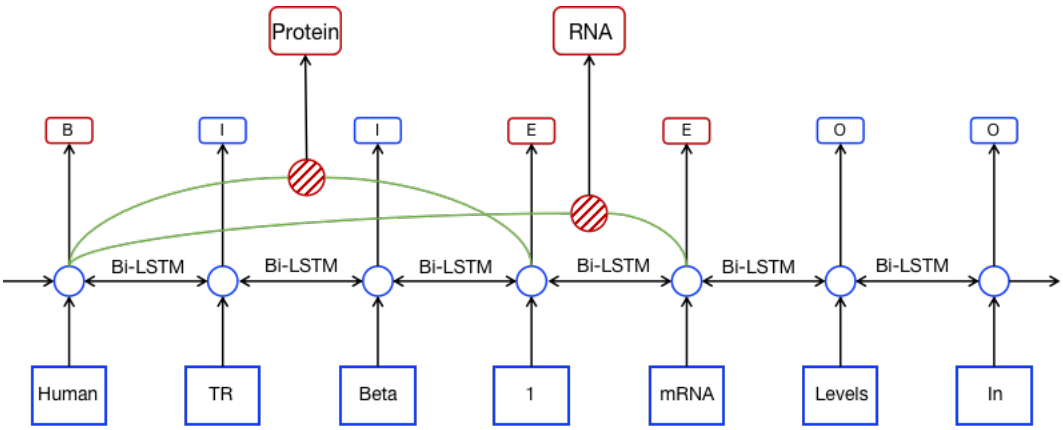

Figure 2: The Architecture of our boundary-aware model. The representation of each token in sentence "Human TR Beta 1 mRNA Levels in.." is feed into a shared bidirectional LSTM layer. We leverage the outputs of Bi-LSTM to detect entity boundaries and their categorical labels. The red circle indicates entity region representations between entity boundaries.

linking (Gupta et al., 2017), relation extraction (Mintz et al., 2009; Liu et al., 2017), co-reference resolution (Chang et al., 2013) and conversation system (Ren et al., 2019) rely on it. Several methods have been proposed on flat named entity recognition (Lample et al., 2016; Ma and Hovy, 2016; Strubell et al., 2017) while few of them address nested entities. Early work on nested entities rely on hand-craft features or rule-based post-processing (Zhang et al., 2004; Zhou et al., 2004; Zhou, 2006). They detect the innermost flat entities with a Hidden Markov Model and then use rule-based post-processing to extract the outer entities.

While most work concerns about named entities, Lu and Roth (2015) present a novel hypergraph-based method to tackle the problem of entity mention detection. One issue of their method is the spurious structure of hyper-graphs. Muis and Lu (2017) improve the method of Lu and Roth (2015) by incorporating mention separators along with features.

Recent studies reveal that stacking sequence model like conditional random filed(CRF) layer can extract entities from inner to outer. Alex et al. (2007) propose several CRF-based methods for the GENIA dataset. However, their approach can not recognize nested entities of the same type. Finkel and Manning (2009) present a chart-based parsing method where each named entity is a constituent in the parsing tree. However, their method is not scalable to larger corpus with a cubic time complexity. Ju et al. (2018) dynamically stack flat NER layers to extract nested entities, each flat layer is based on a Bi-LSTM layer and then a cascaded CRF layer. Their model suffers error propagation from layer to layer, an inner entity can not be detected when a outer entity is identified first.

It is difficult for sequence model, like CRF, to extract nested entities where a token can be included in several entities. Wang et al. (2018) present a transition-based model for nested mention detection using a forest representation. One drawback of their model is the greedy training and decoding. Sohrab and Miwa (2018) consider all possible regions in a sentence and classify them into their entity type or non-entity. However, their exhaustive method considers too many irrelevant regions(non-entity regions) into detecting entity types and the regions are classified individually, without considering the contextual information. Our model focuses on the boundary-relevant regions which is much fewer and the explicit leveraging of boundary information helps to locate entities more precisely.

3 Method

In this paper, we propose a boundary-aware neural model which considers the boundary information into locating and classifying entities. The architecture is illustrated in Figure 2.

Our model is built upon a shared bidirectional LSTM layer. It uses the outputs of LSTM layer to detect entity boundaries and predict categorical labels. We extract entity boundaries as paired tokens with label B and label E, "B" indicates the beginning of an entity and "E" means the end of an entity. We match every detected token with label

B and its corresponding token with label E, the regions between them are identified as candidate entities. We represent entities using the corresponding region outputs of shared LSTM and classify them into categorical labels.

The boundary detection module and entity categorical label prediction module are training simultaneously with a multitask loss function, which can capture the underlying dependencies of entity boundaries and their categorical labels. We will describe each part of our model in detail.

3.1 Token Representation

We represent each token in the sentence following the success of Ma and Hovy (2016) and Lample et al. (2016) that leverages character embedding for the flat NER task.

For a given sentence consisting of n tokens (t_1,t_2,....t_n), we represent the word embedding of i-th token t_i as equation(1):

$$\mathbf{x}_i^w = \mathbf{e}^w(t_i) \tag{1}$$

where $\mathbf{e}^w$ denotes a word embedding lookup table. We use pre-trained word embedding (Chiu et al., 2016) to initialize it.

We capture the orthographic and morphological features of the word by integrating character representations. Denoting the representation of characters within t_i as $\mathbf{x}_i^c$, The embedding of each character within token t_i is denoted as $\mathbf{e}^c(c_j)$. $\mathbf{e}^c$ is the character embedding lookup which is initialized randomly. Then we feed them into a bidirectional LSTM layer to learn hidden states. The forward and backward outputs are concatenated to construct character representations:

$$\mathbf{x}_i^c = [\overrightarrow{\mathbf{h}_i^c}; \overleftarrow{\mathbf{h}_i^c}] \tag{2}$$

where $\overleftarrow{\mathbf{h}_i^c}$ and $\overrightarrow{\mathbf{h}_i^c}$ denote the forward and backward outputs of bidirectional LSTM.

The final token representation is obtained as equation (3), where [;] denotes concatenation.

$$\mathbf{x}_i^t = [\mathbf{x}_i^w; \mathbf{x}_i^c] \tag{3}$$

3.2 Shared Feature Extractor

As shown in Figure 2, we apply the hard parameter sharing mechanism (Ruder, 2017) for multitask training using bidirectional LSTM as shared feature extractor. Hard parameter sharing greatly reduces the risk of overfitting (Baxter, 1997) and increases the correlation of our boundary detection module and categorical label prediction module. Specifically, the hidden state of bidirectional LSTM can be expressed as following:

$$\overrightarrow{\mathbf{h}}_i^t = \overrightarrow{\mathrm{LSTM}}(\mathbf{x}_i^t, \overrightarrow{\mathbf{h}}_{i-1}^t) \tag{4}$$

$$\overleftarrow{\mathbf{h}}_i^t = \overleftarrow{\mathrm{LSTM}}(\mathbf{x}_i^t, \overleftarrow{\mathbf{h}}_{i-1}^t) \tag{5}$$

$$\mathbf{h}_i^t = [\overrightarrow{\mathbf{h}}_i^t; \overleftarrow{\mathbf{h}}_i^t] \tag{6}$$

where $\mathbf{x}_i^t$ is the token representation which is mentioned in section 3.1. We feed $\mathbf{x}_i^t$ into a Dropout layer to prevent overfitting. $\overrightarrow{\mathbf{h}}_i^t$ and $\overleftarrow{\mathbf{h}}_i^t$ denote the i-th forward and backward hidden state of Bi-LSTM layer. Formally, we extract the shared features of each token in a sentence as $\mathbf{h}_i^t$.

3.3 Entity Boundary Detection

Previous works (Lample et al., 2016; Ma and Hovy, 2016) on flat NER (non-nested named entities recognition) predict entity boundaries and categorical labels jointly. However, when entities are nested in other entities, one individual token can be included in many different entities. This means assigning one single categorical label for each token is inappropriate.

We divide nested NER into two subtasks: entity boundary detection and categorical label prediction tasks. Unlike assigning an entity categorical label for each token, we predict the boundary labels first. Formally, given a sentence (t_1,t_2,...t_n), and one entity in the sentence. we represent the entity as $R(i, j)$, which denotes the entity is composed by a continuous token sequence (t_i,t_{i+1},...t_j). Specially, we tag the boundary token t_i as "B" and t_j as "E". The tokens inside entities are assigned with label "I" and non-entity tokens are assigned with "O" labels.

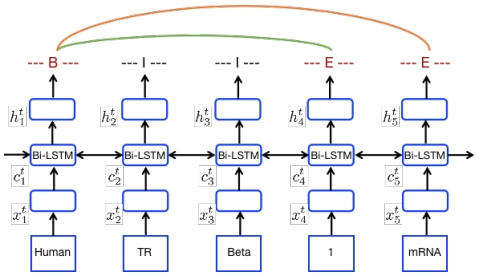

Figure 3: The architecture of entity boundary detection module. We feed the representation of each token in the sentence into a bidirectional LSTM layer, the outputs of LSTM layer are utilized to predict boundary labels.

We detect entity boundaries as shown in Fig-

ure 3. For each token t_i in a sentence, we predict a boundary label by feeding its corresponding shared feature representation $\mathbf{h}_i^t$ (described in section 3.2) into a ReLU activation function and a softmax classifier:

$$\mathbf{o}_i^t = \mathbf{U}\mathbf{h}_i^t + \mathbf{b} \quad (7)$$

$$\mathbf{d}_i^t = \mathrm{softmax}(\mathbf{o}_i^t) \quad (8)$$

where $\mathbf{U}$ and $\mathbf{b}$ are trainable parameters. We compute the KL-divergence multi-label loss between the true distribution $\hat{\mathbf{d}}_i^t$ and the predicted distribution $\mathbf{d}_i^t$ as equation (9):

$$L_{bcls} = -\sum(\hat{\mathbf{d}}_i^t)\log(\mathbf{d}_i^t) \quad (9)$$

Conditional random field (CRF) (Lafferty et al., 2001) is considered good at modeling sequence label dependencies (e.g., label I must be after B). We make a comparison of choosing softmax or CRF as output layer because our sequence labels are different from flat NER models.

3.4 Entity Categorical Label Prediction

Given an input sentence sequence X = (x_1,x_2, ... x_n), and a corresponding boundary label sequence L = (l_1,l_2, ... l_n), we match each token with label B to the token with label E to construct candidate entity regions. Especially, considering there are entities containing one single token, we match tokens with label B to themselves firstly. The representation of entity $R(i,j)$ is obtained as following:

$$\mathrm{R}_{i,j} = \left[\frac{1}{j-i+1}\sum_{k=i}^{j}\mathbf{h}_k^t\right] \quad (10)$$

where $\mathbf{h}_k^t$ denotes the outputs of the shared bidirectional LSTM layer for k-th token in sentence. We simply average the representations for each token within boundary regions. The final representation of entities will be sent into a ReLU activation function and the softmax layer to predict entity categorical labels. We compute the loss of categorical label prediction in equation (11-12):

$$\mathbf{d}_{i,j}^e = \mathrm{softmax}(\mathbf{U}_{i,j}^e\mathrm{R}_{i,j} + \mathbf{b}_{i,j}^e) \quad (11)$$

$$L_{ecls} = -\sum(\hat{\mathbf{d}}_{i,j}^e)\log(\mathbf{d}_{i,j}^e) \quad (12)$$

where $\mathbf{U}_{i,j}^e$ and $\mathbf{b}_{i,j}^e$ are trainable parameters. $\hat{\mathbf{d}}_{i,j}^e$ and $\mathbf{d}_{i,j}^e$ denote the true distribution and predicted distribution of entity categorical labels, respectively.

3.5 Multitask Training

In our model, it is inconvenient and inefficient for the reason that we predict entity categorical labels after all boundary-relative regions have been detected. Considering our boundary detection module and entity categorical label prediction module share the same entity boundaries, we apply a multitask loss for training the two tasks simultaneously.

During training phase, we feed the ground-truth boundary labels into entity categorical label prediction module so that the classifier will be trained without affection from error boundary detection. As for testing phase, the outputs of boundary detection will be collected. The detected boundaries will indicate which entity regions should be considered into predicting categorical labels. The multitask loss function is defined as follows:

$$L_{multi} = \alpha\sum L_{bcls} + (1-\alpha)\sum L_{ecls} \quad (13)$$

where L_{bcls} and L_{ecls} denote the categorical cross-entropy loss for boundary detection module and entity categorical label prediction module, respectively. α is a hyper-parameter which is assigned to control the degree of importance for each task.

4 Experimental Settings

4.1 Dataset

To provide empirical evidence for effectiveness of the proposed model, we employ our experiments on three nested NER datasets: GENIA (Kim et al., 2003), JNLPBA (Kim et al., 2004) and GermEval 2014 (Benikova et al., 2014).

GENIA dataset is constructed based on the GENIA v3.0.2 corpus. We preprocess the dataset following the same settings of (Finkel and Manning, 2009) and (Lu and Roth, 2015). The dataset is split into 8.1:0.9:1 for training, development and testing. The statistics of GENIA dataset is shown as Table 1.

Item	Train	Dev	Test	Overall	Nested
Document	1599	189	212	2000	-
Sentences	15023	1669	1854	18546	-
Percentage	81%	9%	10%	100%	-
DNA	7650	1026	1257	9933	1744
RNA	692	132	109	933	407
Protein	28728	2303	3066	34097	1902
Cell Line	3027	325	438	3790	347
Cell Type	5832	551	604	6987	389
Overall	45929	4337	5474	55740	4789

Table 1: Statistics of GENIA dataset

JNLPBA dataset is originally from GENIA corpus. It contains a training and testing datasets.

However, only the flat and top-most entities are preserved. We collapse the sub-categories into 5 categories following the same settings as GENIA dataset.

GermEval 2014 dataset contains German nested named entities. The dataset covers over 31,000 sentences corresponding to over 590,000 tokens.

4.2 Baseline Methods

We compare our model with several state-of-the-art models on GENIA dataset. These methods can be divided into three groups:

Finkel and Manning (2009) and Ju et al. (2018) propose CRF-based sequence labeling approaches for nested named entity recognition. Finkel and Manning (2009) leverage entity-level features while Ju et al. (2018) propose neural-based method. We rerun the codes of Ju et al. (2018) because they have not shared their pre-processed dataset.

Sohrab and Miwa (2018) propose an exhaustive region classification model for nested NER. We re-implement their method according to their paper because they have not shared the codes.

Lu and Roth (2015) and Muis and Lu (2017) build hyper-graphs to represent both the nested entities and their mentions. Muis and Lu (2017) improve the method of Lu and Roth (2015).

4.3 Parameter Settings

Our model is implemented by PyTorch framework[1] [2]. We use Adam optimizer for training our model. We initialize word vectors with a 200-dimension pre-trained word embedding the same as Ju et al. (2018) and Sohrab and Miwa (2018) while the char embedding is set to 50-dimension and initialized randomly. The learning rate is set to 0.005. We set a 0.5 dropout rate for the Dropout layer employed after token-level LSTM during training phase. The output dimension of our shared bidirectional LSTM is 200. The coefficient α of multitask loss is tuned during development process. All of our experiments are performed on the same machine (NVIDIA 1080ti GPU and Intel i7-8700 CPU).

[1] https://pytorch.org/

[2] Code is available at https://github.com/thecharm/boundary-aware-nested-ner

4.4 Evaluation Metrics

We use a strict evaluation metrics that an entity is confirmed correct when the entity boundary and the entity categorical label are correct simultaneously. We employ precision, recall and F-score to evaluate the performance.

5 Results and Discussion

5.1 Overall Evaluation

We conduct our experiments on GENIA test dataset for nested named entity recognition. Table 2 shows our method outperforms the compared methods both in recall and F-score metrics. The CRF-based model is considered as more efficient in sequence labeling task, we compare the utilization of softmax and CRF as output layer of boundary detection module. The results show they gain comparable scores in precision, recall and F-score. However, the CRF-based model is time-consuming, about 3-5 times slower than the softmax-based model in inference speed.

Model	P(%)	R(%)	F(%)
Finkel and Manning (2009)[3]	75.4	65.9	70.3
Lu and Roth (2015)[3]	72.5	65.2	68.7
Muis and Lu (2017)[3]	75.4	66.8	70.8
Sohrab and Miwa (2018)	73.3	68.3	70.7
Ju et al. (2018)	**76.1**	66.8	71.1
Our model(softmax)	75.9	**73.6**	**74.7**
Our model(CRF)	74.6	73.2	73.9

Table 2: Performance on GENIA test set. Our models with softmax and CRF outperform other state-of-the-art methods.

Our model achieves a recall value of 73.6% and outperforms compared methods in Recall value with a large margin. We think that our model extract entities with a more accurate boundaries comparing to other methods. We evaluate it in experiments on boundary detection module.

Model	P(%)	R(%)	F(%)
Sohrab and Miwa (2018)	**75.0**	60.8	67.2
Ju et al. (2018)	72.9	61.5	66.7
Our model	74.5	**69.1**	**71.7**

Table 3: Performance on GermEval 2014 test set. Our model outperforms two state-of-the-art methods in nested NER.

The GermEval 2014 dataset from KONVENS 2014 shared task is a German NER dataset. It

[3] The results are taken from their papers.

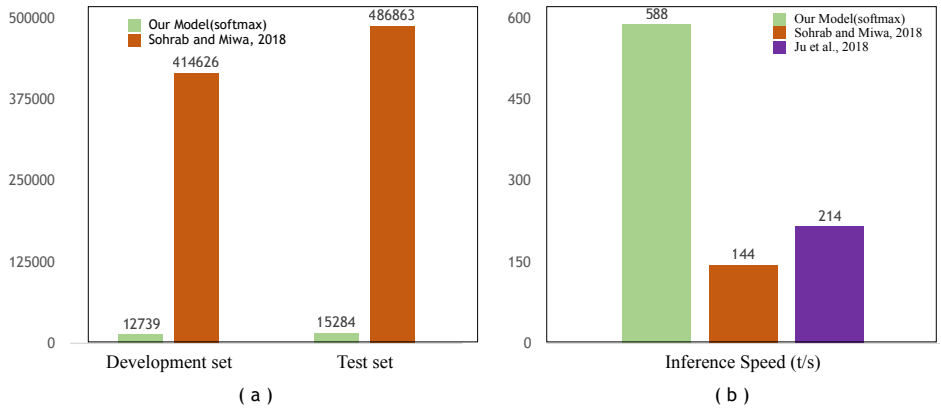

Figure 4: (a): The number of candidate entity regions in our model with softmax and the approach of Sohrab and Miwa (2018) when evaluating on GENIA test and development set; (b): The inference speed of our model and compared models on GENIA test set. t/s indicates token per second.

contains few nested entities. Previous works in this dataset ignore nested entities or extract inner and outer entities in two independent models. Our method can extract nested entities in an end-to-end way. We compare our method with two state-of-the-art approaches in Table 3. Our method outperforms their approaches both in Recall and F-score metrics.

Table 4 describes the performances of our model on the five categories on the test dataset. Our model outperforms the model described in Ju et al. (2018) and Sohrab and Miwa (2018) with F-score value on all categories.

Category	P(%)	R(%)	F(%)	Ju. F(%)	Soh. F(%)
DNA	73.6	67.8	**70.6**	70.1	67.8
RNA	82.2	80.7	**81.5**	80.8	75.9
protein	76.7	76.0	**76.4**	72.7	72.9
cell line	77.8	65.8	**71.3**	66.9	63.6
cell type	73.9	71.2	**72.5**	71.3	69.8
overall	75.8	73.6	**74.7**	71.1	70.7

Table 4: Our results on five categories compared to Ju et al. (2018) and Sohrab and Miwa (2018) on GENIA test set.

5.2 Performance of Boundary Detection

We conduct experiments on boundary detection to illustrate that our model extract entity boundaries more precisely comparing to Sohrab and Miwa (2018) and Ju et al. (2018). Table 5 shows the results of boundary detection on GENIA test dataset. Our model locates entities more accurately with a higher recall value (76.9%) than the comparing methods. It gives a reason why our model outperforms other state-of-the-art methods in recall value. We exploit boundary information explicitly and consider the dependencies of boundaries and entity categorical labels with a multitask loss. While in the method of Sohrab and Miwa (2018), candidate entity regions are classified individually.

Model	Boundary Detection		
	P(%)	R(%)	F(%)
Sohrab and Miwa (2018)	76.6	69.2	72.7
Ju et al. (2018)	79.9	67.08	73.4
Our model(softmax)	79.7	**76.9**	**78.3**

Table 5: Performance of Boundary Detection on GENIA test set.

Table 6 describes the performance of our model in detecting boundary labels for each token in sentences. The results are based on the shared bidirectional LSTM and a softmax classifier. Our model extracts entity boundaries with a relatively high performance. This facilitates the prediction of entity categorical labels because the candidate entity regions are more likely to be true entities.

Boundary Label	P(%)	R(%)	F(%)
O (non-entity)	99.3	99.0	99.2
B (beginning)	84.4	84.3	84.3
E (end)	86.0	87.2	86.6
I (inner-entity)	82.8	88.6	85.6

Table 6: Performance of Boundary Label Prediction with softmax classifier on GENIA test set.

5.3 Inference Time

Figure 4(a) shows the number of candidate entity regions in our model with softmax and the approach of Sohrab and Miwa (2018). Comparing to classifying all possible regions in sentences, our

model only concerns about boundary-relevant regions which is much fewer. We compare the inference speed of our model and the approaches of Sohrab and Miwa (2018) and Ju et al. (2018) in Figure 4(b). Our model is about 4 times faster than Sohrab and Miwa (2018) and about 3 times faster than Ju et al. (2018). The cascaded CRF layers of Ju et al. (2018) are the limitation in inference speed.

5.4 Performance of Multitask Learning

Table 7 shows the performance of our pipeline model and multitask model on GENIA development set and test set. For pipeline model, we train the boundary detection module and entity categorical label prediction module separately. Our multitask model has a higher F value both in development set and test set.

Model	Development Set			Test Set		
	P(%)	R(%)	F(%)	P(%)	R(%)	F(%)
Pipeline	74.5	74.8	74.6	75.4	72.2	73.8
Multitask	74.5	75.6	**75.0**	75.9	73.4	**74.7**

Table 7: Performance Comparison of our pipeline model and multitask model on GENIA development set and test set.

Multitask learning can capture the underlying dependencies of boundaries and entity categorical labels. It helps the model focus its attention on those features that actually matter (Ruder, 2017). In pipeline model, entity categorical prediction module will not share information with boundary detection module because they are trained separately. However, entity categorical prediction module and boundary detection module share the same entity boundaries. We assign a shared feature extractor (the bidirectional LSTM layer) to extract the features utilized in both entity categorical prediction and boundary detection. The results have demonstrated that the framework of multitask learning improves final performance.

5.5 Ablation Study and Flat NER

We conduct ablation experiments on GENIA development set to evaluate the contributions of neural components including dropout layer, pretrained word embedding and the character-level LSTM. The results are listed in Table 8. All these components contribute to the effectiveness of our model. Dropout layer contributes significantly for both precision and recall values.

Setting	P(%)	R(%)	F(%)
Our Model(softmax)	74.5	75.6	**75.0**
without Dropout	72.6	73.1	72.9
without Pre-trained	73.8	**75.7**	74.7
without Char repr.	**75.3**	73.9	74.6

Table 8: Results of Ablation Tests on GENIA development set.

To prove our model can work on nested NER and also flat NER task, we perform experiments on the JNLPBA dataset. We achieve 73.6 in term of F-score which is comparable with the state-of-the-art result of Gridach (2017).

6 Case Study

Table 9 shows a case study comparing our model with exhaustive model (Sohrab and Miwa, 2018) and Layered model (Ju et al., 2018). In the example, "human TATA binding factor" is an entity nested in entity "transcriptionally active human TATA binding factor". Our model with multitask learning extracts both entities with exact boundaries and entity categorical labels. Exhaustive model gets the error boundaries and misses the token "human" in entities. Comparing to layered model only detecting an outer entity, our model extract both inner and outer entities. It demonstrates that our combination of sequence labeling models and region classification models can locate entities precisely and extract both inner and outer entities.

Sentence	Cloning of a transcriptionally active human TATA binding factor.
Gold Label	protein: {human TATA binding factor; transcriptionally active human TATA binding factor}
Exhaustive model	protein: {TATA binding factor; transcriptionally active human TATA binding factor}
Layered model	protein: {transcriptionally active human TATA binding factor}
Our model(pipeline)	protein: {human TATA binding factor;}
Our model(multitask)	protein: {human TATA binding factor; transcriptionally active human TATA binding factor}

Table 9: An example of predicted results in GENIA test dataset.

For our pipeline model, without the dependencies information from entity categorical labels, it misses the outer entity boundaries and only extracts the inner one. It verifies that the multitask learning can share boundary information between boundary detection module and entity categorical

label prediction module, which is very effective for nested NER.

7 Conclusion

This paper presents a boundary-aware model which leverages boundaries to predict entity categorical labels. Our model combines sequence labeling model and region classification model to locate and classify nested entities with high performance. To capture the underlying dependencies of boundary detection module and entity categorical prediction module, we apply a multitask loss for training the two tasks simultaneously. Our model outperforms existing nested models in terms of F-score.

For future work, we consider to model the dependencies among entity regions explicitly and improve the performance of boundary detection module which is important for entity categorical label prediction.

Acknowledgement

This work was supported by the Fundamental Research Funds for the Central Universities, SCUT (No. 2017ZD048, D2182480), the Science and Technology Planning Project of Guangdong Province (No.2017B050506004), the Science and Technology Programs of Guangzhou (No. 201704030076,201802010027,201902010046) and a CUHK Research Committee Funding (Direct Grants) (Project Code: EE16963).

References

Beatrice Alex, Barry Haddow, and Claire Grover. 2007. Recognising nested named entities in biomedical text. In *Proceedings of the Workshop on BioNLP 2007: Biological, Translational, and Clinical Language Processing*, pages 65–72. Association for Computational Linguistics.

Jonathan Baxter. 1997. A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine learning*, 28(1):7–39.

Darina Benikova, Chris Biemann, and Marc Reznicek. 2014. Nosta-d named entity annotation for german: Guidelines and dataset. In *LREC*, pages 2524–2531.

Kai-Wei Chang, Rajhans Samdani, and Dan Roth. 2013. A constrained latent variable model for coreference resolution. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 601–612.

Billy Chiu, Gamal Crichton, Anna Korhonen, and Sampo Pyysalo. 2016. How to train good word embeddings for biomedical nlp. In *Proceedings of the 15th workshop on biomedical natural language processing*, pages 166–174.

Jenny Rose Finkel and Christopher D Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 141–150. Association for Computational Linguistics.

Mourad Gridach. 2017. Character-level neural network for biomedical named entity recognition. *Journal of biomedical informatics*, 70:85–91.

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.

Meizhi Ju, Makoto Miwa, and Sophia Ananiadou. 2018. A neural layered model for nested named entity recognition. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1446–1459.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun‘‘ichi Tsujii. 2003. Genia corpus‘‘a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

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.

John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Liyuan Liu, Xiang Ren, Qi Zhu, Shi Zhi, Huan Gui, Heng Ji, and Jiawei Han. 2017. Heterogeneous supervision for relation extraction: A representation learning approach. *arXiv preprint arXiv:1707.00166*.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 857–867.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics.

Aldrian Obaja Muis and Wei Lu. 2017. Labeling gaps between words: Recognizing overlapping mentions with mention separators. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2608–2618, Copenhagen, Denmark. Association for Computational Linguistics.

Da Ren, Yi Cai, Xue Lei, Jingyun Xu, Qing Li, and Ho-fung Leung. 2019. A multi-encoder neural conversation model. *Neurocomputing*, 358:344–354.

Sebastian Ruder. 2017. An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Dan Shen, Jie Zhang, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2003. Effective adaptation of a hidden markov model-based named entity recognizer for biomedical domain. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine-Volume 13*, pages 49–56. Association for Computational Linguistics.

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

Emma Strubell, Patrick Verga, David Belanger, and Andrew McCallum. 2017. Fast and accurate entity recognition with iterated dilated convolutions. *arXiv preprint arXiv:1702.02098*.

Bailin Wang, Wei Lu, Yu Wang, and Hongxia Jin. 2018. A neural transition-based model for nested mention recognition. *arXiv preprint arXiv:1810.01808*.

Jie Zhang, Dan Shen, Guodong Zhou, Jian Su, and Chew-Lim Tan. 2004. Enhancing hmm-based biomedical named entity recognition by studying special phenomena. *Journal of biomedical informatics*, 37(6):411–422.

GD Zhou. 2006. Recognizing names in biomedical texts using mutual information independence model and svm plus sigmoid. *International Journal of Medical Informatics*, 75(6):456–467.

Guodong Zhou, Jie Zhang, Jian Su, Dan Shen, and Chewlim Tan. 2004. Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, 20(7):1178–1190.