A Span-Based Model for Joint Overlapped and Discontinuous Named Entity Recognition

Fei Li[1] and **Zhichao Lin**[2] and **Meishan Zhang**[2] and **Donghong Ji**[1*]
1. Department of Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, China
2. School of New Media and Communication, Tianjin University, China
{lifei_csnlp,dhji}@whu.edu.cn and mason.zms@gmail.com

Abstract

Research on overlapped and discontinuous named entity recognition (NER) has received increasing attention. The majority of previous work focuses on either overlapped or discontinuous entities. In this paper, we propose a novel span-based model that can recognize both overlapped and discontinuous entities jointly. The model includes two major steps. First, entity fragments are recognized by traversing over all possible text spans, thus, overlapped entities can be recognized. Second, we perform relation classification to judge whether a given pair of entity fragments to be overlapping or succession. In this way, we can recognize not only discontinuous entities, and meanwhile doubly check the overlapped entities. As a whole, our model can be regarded as a relation extraction paradigm essentially. Experimental results on multiple benchmark datasets (i.e., CLEF, GENIA and ACE05) show that our model is highly competitive for overlapped and discontinuous NER.

1 Introduction

Named entity recognition (NER) (Sang and De Meulder, 2003) is one fundamental task for natural language processing (NLP), due to its wide application in information extraction and data mining (Lin et al., 2019b; Cao et al., 2019). Traditionally, NER is presented as a sequence labeling problem and widely solved by conditional random field (CRF) based models (Lafferty et al., 2001). However, this framework is difficult to handle overlapped and discontinuous entities (Lu and Roth, 2015; Muis and Lu, 2016), which we illustrate using two examples as shown in Figure 1. The two entities "Pennsylvania" and "Pennsylvania radio station" are nested with each other,[1] and the second example shows a discontinuous entity "mitral leaflets thickened" involving three fragments.

There have been several studies to investigate overlapped or discontinuous entities (Finkel and Manning, 2009; Lu and Roth, 2015; Muis and Lu, 2017; Katiyar and Cardie, 2018; Wang and Lu, 2018; Ju et al., 2018; Wang et al., 2018; Fisher and Vlachos, 2019; Luan et al., 2019; Wang and Lu, 2019). The majority of them focus on overlapped NER, with only several exceptions to the best of our knowledge. Muis and Lu (2016) present a hypergraph model that is capable of handling both overlapped and discontinuous entities. Wang and Lu (2019) extend the hypergraph model with long short-term memories (LSTMs) (Hochreiter and Schmidhuber, 1997). Dai et al. (2020) proposed a transition-based neural model for discontinuous NER. By using these models, NER could be conducted universally without any assumption to exclude overlapped or discontinuous entities, which could be more practical in real applications.

The hypergraph (Muis and Lu, 2016; Wang and Lu, 2019) and transition-based models (Dai et al., 2020) are flexible to be adapted for different tasks, achieving great successes for overlapped or discontinuous NER. However, these models need to manually define graph nodes, edges and transition actions. Moreover, these models build graphs or generate transitions along the words in the sentences gradually, which may lead to error propagation (Zhang et al., 2016). In contrast, the span-based scheme might be a good alternative, which is much simpler including only span-level classification. Thus, it needs less manual intervention and meanwhile span-level classification can be fully parallelized without error propagation. Recently, Luan et al. (2019) utilized the span-based model for information extraction effectively.

In this work, we propose a novel span-based joint model to recognize overlapped and discon-

*Corresponding author.

[1] We consider "nested" as a special case of "overlapped".

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 4814–4828
August 1–6, 2021.

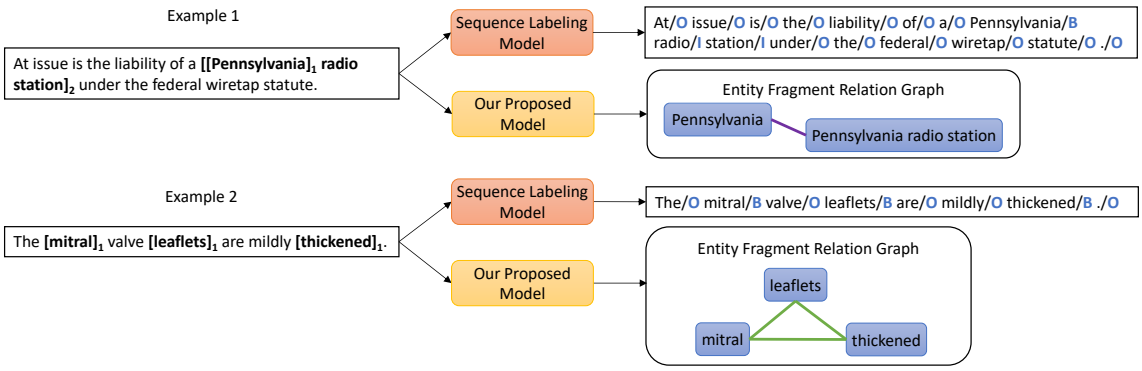


Figure 1: Examples to illustrate the differences between the sequence labeling model and our span-based model. On the left, word fragments marked with the same number belong the same entity. On the right, blue rectangles denote the recognized entity fragments, and solid lines indicate the `Succession` or `Overlapping` relations between them (the two relations are mutually exclusive).

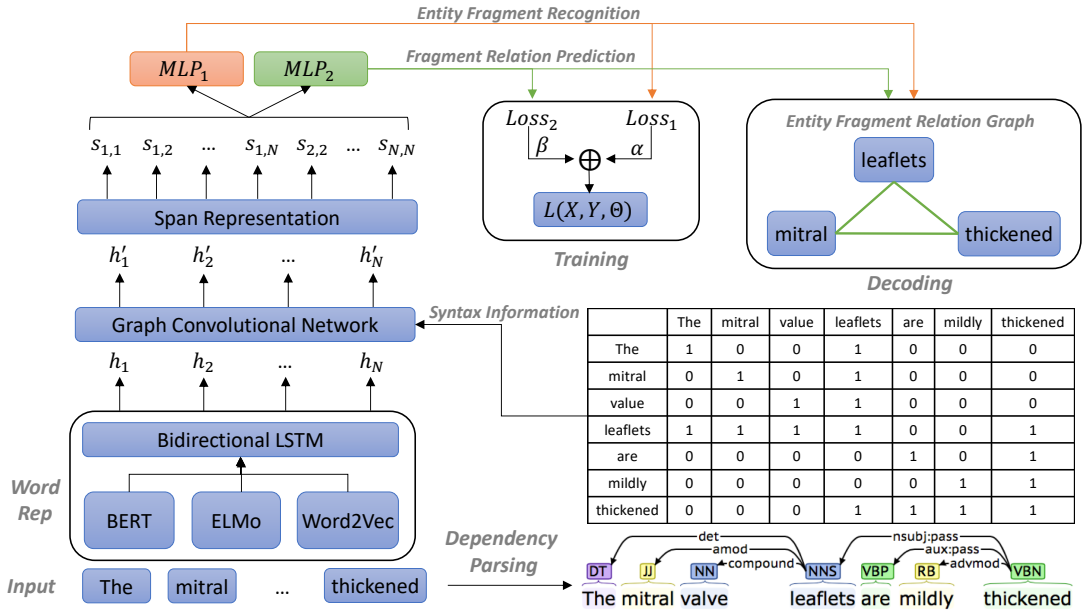
tinuous entities simultaneously in an end-to-end way. The model utilizes BERT (Devlin et al., 2019) to produce deep contextualized word representations, and then enumerates all candidate text spans (Luan et al., 2019), classifying whether they are entity fragments. Following, fragment relations are predicted by another classifier to determine whether two specific fragments involve a certain relation. We define two relations for our goal: `Overlapping` or `Succession`, which are used for overlapped and discontinuous entities, respectively. In essence, the joint model can be regarded as one kind of relation extraction models, which is adapted for our goal. To enhance our model, we utilize the syntax information as well by using a dependency-guided graph convolutional network (Kipf and Welling, 2017; Zhang et al., 2018; Jie and Lu, 2019; Guo et al., 2019).

We evaluate our proposed model on several benchmark datasets which includes both overlapped and discontinuous entities (e.g., CLEF (Suominen et al., 2013)). The results show that our model outperforms the hypergraph (Muis and Lu, 2016; Wang and Lu, 2019) and transition-based models (Dai et al., 2020). Besides, we conduct experiments on two benchmark datasets including only overlapped entities (i.e., GENIA (Kim et al., 2003) and ACE05). Experimental results show that our model can also obtain comparable performances with the state-of-the-art models (Luan et al., 2019; Wadden et al., 2019; Straková et al., 2019). In addition, we observe that our approaches for model enhancement are effective in the benchmark datasets. Our code is available at `https://github.com/foxlf823/sodner`.

2 Related Work

In the NLP domain, NER is usually considered as a sequence labeling problem (Liu et al., 2018; Lin et al., 2019b; Cao et al., 2019). With well-designed features, CRF-based models have achieved the leading performance (Lafferty et al., 2001; Finkel et al., 2005; Liu et al., 2011). Recently, neural network models have been exploited for feature representations (Chen and Manning, 2014; Zhou et al., 2015). Moreover, contextualized word representations such as ELMo (Peters et al., 2018), Flair (Akbik et al., 2018) and BERT (Devlin et al., 2019) have also achieved great success. As for NER, the end-to-end bi-directional LSTM CRF models (Lample et al., 2016; Ma and Hovy, 2016; Yang et al., 2018) is one representative architecture. These models are only capable of recognizing regular named entities.

For overlapped NER, the earliest model to our knowledge is proposed by Finkel and Manning (2009), where they convert overlapped NER as a parsing task. Lu and Roth (2015) propose a hypergraph model to recognize overlapped entities and lead to a number of extensions (Muis and Lu, 2017; Katiyar and Cardie, 2018; Wang and Lu, 2018). Moreover, recurrent neural networks (RNNs) are also used for overlapped NER (Ju et al., 2018; Wang et al., 2018). Other approaches include multi-grained detection (Xia et al., 2019), boundary detection (Zheng et al., 2019), anchor-region network (Lin et al., 2019a) and machine reading comprehension (Li et al., 2020). The state-of-the-art models for overlapped NER include the sequence-to-sequence (seq2seq) model (Straková et al., 2019), where the decoder predicts multiple



	The	mitral	value	leaflets	are	mildly	thickened
The	1	0	0	1	0	0	0
mitral	0	1	0	1	0	0	0
value	0	0	1	1	0	0	0
leaflets	1	1	1	1	0	0	1
are	0	0	0	0	1	0	1
mildly	0	0	0	0	0	1	1
thickened	0	0	0	1	1	1	1

Figure 2: The architecture of our model. The input is "The $[\textbf{mitral}]_1$ valve $[\textbf{leaflets}]_1$ are mildly $[\textbf{thickened}]_1$". $\boldsymbol{h}_1$ denotes the original word representation and $\boldsymbol{h}'_1$ denotes the syntax-enhanced word representation. $\boldsymbol{s}_{1,2}$ denotes the span representation. α and β control the loss weights of two tasks, namely recognizing entity fragments from text spans and predicting the relation between each pair of fragments.

labels for a word and move to next word until it outputs the "end of word" label, and the span-based model (Luan et al., 2019; Wadden et al., 2019), where overlapped entities are recognized by classification for enumerated spans.

Compared with the number of related work for overlapped NER, there are no related studies for only discontinuous NER, but several related studies for both overlapped and discontinuous NER. Early studies addressed such problem by extending the BIO label scheme (Tang et al., 2013; Metke-Jimenez and Karimi, 2016). Muis and Lu (2016) first proposed a hypergraph-based model for recognizing overlapped and discontinuous entities, and then Wang and Lu (2019) utilized deep neural networks to enhance the model. Very recently, Dai et al. (2020) proposed a transition-based neural model with manually-designed actions for both overlapped and discontinuous NER. In this work, we also aim to design a competitive model for both overlapped and discontinuous NER. Our differences are that our model is span-based (Luan et al., 2019) and it is also enhanced by dependency-guided graph convolutional network (GCN) (Zhang et al., 2018; Guo et al., 2019).

To our knowledge, syntax information is commonly neglected in most previous work for overlapped or discontinuous NER, except Finkel and Manning (2009). The work employs a constituency parser to transform a sentence into a nested entity tree, and syntax information is used naturally to facilitate NER. By contrast, syntax information has been utilized in some studies for traditional regular NER. Under the traditional statistical setting, syntax information is used by manually-crafted features (Hacioglu et al., 2005; Ling and Weld, 2012) or auxiliary tasks (Florian et al., 2006) for NER. Recently, Jie et al. (2017) build a semi-CRF model based on dependency information to optimize the research space of NER recognition. Jie and Lu (2019) stack the dependency-guided graph convolutional network (Zhang et al., 2018; Guo et al., 2019) on top of the BiLSTM layer. These studies have demonstrated that syntax information could be an effective feature source for NER.

3 Method

The key idea of our model includes two mechanisms. First, our model enumerates all possible text spans in a sentence and then exploits a multi-classification strategy to determine whether one span is an entity fragment as well as the entity type. Based on this mechanism, overlapped entities could be recognized. Second, our model performs pairwise relation classifications over all entity fragments to recognize their relationships. We define three kinds of relation types:

- `Succession`, indicating that the two entity fragments belong to one single named entity.
- `Overlapping`, indicating that the two entity fragments have overlapped parts.
- `Other`, indicating that the two entity fragments have other relations or no relations.

With the `Succession` relation, we can recognize discontinuous entities. Through the `Overlapping` relation, we aim to improve the recognition of overlapped entities with double supervision. The proposed model is essentially a relation extraction model being adapted for our task. The architecture of our model is illustrated in Figure 2, where the main components include the following parts: (1) word representation, (2) graph convolutional network, (3) span representation, and (4) joint decoding, which are introduced by the following subsections, respectively.

3.1 Word Representation

We exploit BERT (Devlin et al., 2019) as inputs for our model, which has demonstrated effective for a range of NLP tasks.[2] Given an input sentence $\boldsymbol{x} = \{x_1, x_2, ..., x_N\}$, we convert each word x_i into word pieces and then feed them into a pre-trained BERT module. After the BERT calculation, each sentential word may involve vectorial representations of several pieces. Here we employ the representation of the beginning word piece as the final word representation following (Wadden et al., 2019). For instance, if "fevers" is split into "fever" and "##s", the representation of "fever" is used as the whole word representation. Therefore, all the words in the sentence $\boldsymbol{x}$ correspond to a matrix $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N\} \in \mathbb{R}^{N \times d_h}$, where d_h denotes the dimension of $\boldsymbol{h}_i$.

3.2 Graph Convolutional Network

Dependency syntax information has been demonstrated to be useful for NER previously (Jie and Lu, 2019). In this work, we also exploit it to enhance our proposed model.[3] Graph convolutional network (GCN) (Kipf and Welling, 2017) is one representative method to encode dependency-based graphs, which has been shown effective in information extraction (Zhang et al., 2018). Thus, we choose it as one standard strategy to enhance our word representations. Concretely, we utilize the attention-guided GCN (AGGCN) (Guo et al., 2019) to reach our goal, as it can bring better performance compared with the standard GCN.

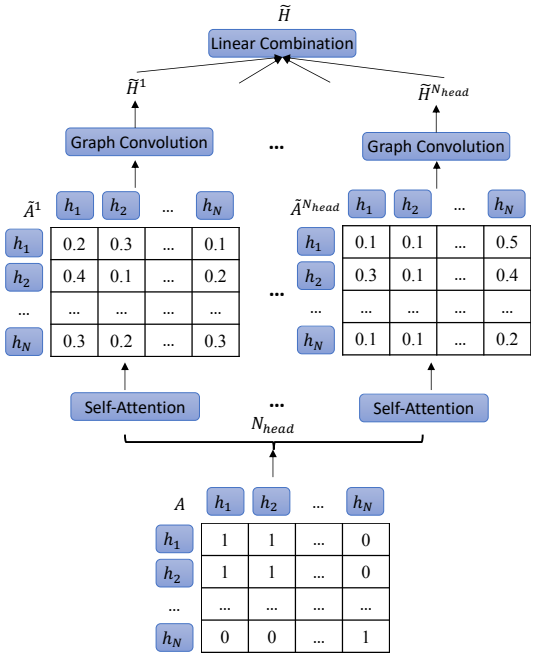


Figure 3: The architecture of our graph convolutional network. Graph Convolution: Equation 1. Self-Attention: Equation 2.

In order to illustrate the network of AGGCN (Figure 3), we start with the standard GCN module. Given the word representations $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N\}$, the standard GCN uses the following equation to update them:

$$\boldsymbol{h}_i^{(l)} = \sigma(\sum_{j=1}^{N} \boldsymbol{A}_{ij} \boldsymbol{W}^{(l)} \boldsymbol{h}_j^{(l-1)} + \boldsymbol{b}^{(l)}), \quad (1)$$

where $\boldsymbol{W}^{(l)}$ and $\boldsymbol{b}^{(l)}$ are the weight and bias of the l-th layer. $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ is an adjacency matrix obtained from the dependency graph, where $\boldsymbol{A}_{ij} = 1$ indicates there is an edge between the word i and j in the dependency graph. Figure 2 offers an example of the matrix which is produced by the corresponding dependency syntax tree.

In fact, $\boldsymbol{A}$ can be considered as a form of hard attention in GCN, while AGGCN (Guo et al., 2019) aims to improve the method by using $\boldsymbol{A}$ in the lower layers and updating $\boldsymbol{A}$ at the higher layers via multi-head self-attention (Vaswani et al., 2017) as below:

$$\tilde{\boldsymbol{A}}^t = \text{softmax}(\frac{\boldsymbol{H}^t \boldsymbol{W}_Q^t \times (\boldsymbol{H}^t \boldsymbol{W}_K^t)^T}{\sqrt{d_{head}}}), \quad (2)$$

[2] We also investigate the effects of different word encoders in the experiments. Please refer to Appendix A.

[3] Some cases are shown in Appendix B.

where $\boldsymbol{W}_Q^t$ and $\boldsymbol{W}_K^t$ are used to project the input $\boldsymbol{H}^t \in \mathbb{R}^{N \times d_{head}}$ ($d_{head} = \frac{d_h}{N_{head}}$) of the t-th head into a query and a key. $\tilde{\boldsymbol{A}}^t \in \mathbb{R}^{N \times N}$ is the updated adjacency matrix for the t-th head.

For each head t, AGGCN uses $\tilde{\boldsymbol{A}}^t$ and a densely connected layer to update the word representations, which is similar to the standard GCN as shown in Equation 1. The output of the densely connected layer is $\tilde{\boldsymbol{H}}^t \in \mathbb{R}^{N \times d_h}$. Then a linear combination layer is used to merge the output of each head, namely $\tilde{\boldsymbol{H}} = [\tilde{\boldsymbol{H}}^1, \cdots, \tilde{\boldsymbol{H}}^{N_{head}}]\boldsymbol{W}_1$, where $\boldsymbol{W}_1 \in \mathbb{R}^{(N_{head} \times d_h) \times d_h}$ is the weight and $\tilde{\boldsymbol{H}} \in \mathbb{R}^{N \times d_h}$ is the final output of AGGCN.

After that, $\tilde{\boldsymbol{H}}$ is concatenated with the original word representations $\boldsymbol{H}$ to form final word representations $\boldsymbol{H}' \in \mathbb{R}^{N \times (d_h + d_f)} = [\boldsymbol{H}, \tilde{\boldsymbol{H}}\boldsymbol{W}_2]$, where $\boldsymbol{W}_2 \in \mathbb{R}^{d_h \times d_f}$ indicates a linear transformation for dimensionality reduction.[4]

3.3 Span Representation

We employ span enumeration (Luan et al., 2019) to generate text spans. Take the sentence "The mitral valve leaflets are mildly thickened" in Figure 2 as an example, the generated text spans will be "The", "The mitral", "The mitral valve", ..., "mildly", "mildly thickened" and "thickened". To represent a text span, we use the concatenation of word representations of its startpoint and endpoint. For example, given word representations $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N\} \in \mathbb{R}^{N \times d_h}$ (or $\boldsymbol{H}' = \{\boldsymbol{h}'_1, \boldsymbol{h}'_2, ..., \boldsymbol{h}'_N\}$) and a span (i, j) that starts at the position i and ends at j, the span representation will be

$$\boldsymbol{s}_{i,j} = [\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{w}] \text{ or } [\boldsymbol{h}'_i, \boldsymbol{h}'_j, \boldsymbol{w}], \quad (3)$$

where $\boldsymbol{w}$ is a 20-dimensional embedding to represent the span width following previous work (Luan et al., 2019; Wadden et al., 2019). Thus, the dimension d_s of $\boldsymbol{s}_{i,j}$ is $2d_h + 20$ (or $2(d_h + d_f) + 20$).

3.4 Decoding

Our decoding consists of two parts. First, we recognize all valid entity fragments, and then perform pairwise classifications over the fragments to uncover their relationships.

Entity Fragment Recognition: Given a span (i, j) represented as $\boldsymbol{s}_{i,j}$, we utilize one MLP to classify whether the span is an entity fragment and what is the entity type, formalized as:

$$\boldsymbol{p}_1 = \text{softmax}(\text{MLP}_1(\boldsymbol{s}_{i,j})), \quad (4)$$

where $\boldsymbol{p}_1$ indicates the probabilities of entity types such as *Organization*, *Disease* and *None* (i.e., not an entity fragment).

Fragment Relation Prediction: Given two entity fragments (i, j) and $(\tilde{i}, \tilde{j})$ represented as $\boldsymbol{s}_{i,j}$ and $\boldsymbol{s}_{\tilde{i},\tilde{j}}$, we utilize another MLP to classify their relations:

$$\boldsymbol{p}_2 = \text{softmax}(\text{MLP}_2([\boldsymbol{s}_{i,j}, \boldsymbol{s}_{i,j} * \boldsymbol{s}_{\tilde{i},\tilde{j}}, \boldsymbol{s}_{\tilde{i},\tilde{j}}])), \quad (5)$$

where $\boldsymbol{p}_2$ indicates the probabilities of three classes, namely *Succession*, *Overlapping* and *Other*, and the feature representations are mostly referred from Luan et al. (2019) and Wadden et al. (2019). Noticeably, although the overlapped entities can be recognized at the first step, here we use the `Overlapping` as one auxiliary strategy to further enhance the model.

During decoding (Algorithm 1), our model recognizes entity fragments from text spans (lines 2-4) in the input sentence and selects each pair of these fragments to determine their relations (lines 5-7). Therefore, the prediction results can be considered as an *entity fragment relation graph* (line 8), where a node denotes an entity fragment and an edge denotes the relation between two entity fragments.[5] The decoding object is to find all the subgraphs in which each node connects with any other node (line 9). Thus, each of such subgraph composes an entity (line 10). In particular, the entity fragment that has no edge with others composes an entity by itself.

Algorithm 1 Decoding algorithm.

Input: An input sentence $\boldsymbol{x} = \{x_1, x_2, ..., x_N\}$
Output: The recognized results R
1: S = EnumerateSpan($\boldsymbol{x}$) where $S = \{s_{1,1}, s_{1,2}, ...\}$
2: **for** $s_{i,j}$ in S **do**
3: **if** IsEntityFragment($s_{i,j}$) **then**
4: $V \leftarrow s_{i,j}$
5: **for** each pair $s_{i,j}$, $s_{\tilde{i},\tilde{j}}$ in V **do**
6: **if** IsSuccession($s_{i,j}$, $s_{\tilde{i},\tilde{j}}$) **then**
7: $E \leftarrow < s_{i,j}, s_{\tilde{i},\tilde{j}} >$
8: Graph $G = \{V, E\}$
9: **for** g in FindCompleteSubgraphs(G) **do**
10: $R \leftarrow g$
11: **return** R

[4] We employ third-party tools to perform parsing for the corpora that do not contain gold syntax annotations. Since sometimes parsing may fail, dependency-guided GCN will be noneffective. Concatenation can remedy such problem since $\boldsymbol{H}$ still works even if $\tilde{\boldsymbol{H}}$ is invalid.

[5] We only use the `Succession` relations during decoding while ignore the `Overlapping` relations. The `Overlapping` relations are only used during training.

		CLEF	CLEF-Dis	CADEC	GENIA	ACE
# Documents or Sentences	Train	179	534	875	1,599	370
	Dev	20	303	187	200	43
	Test	99	430	188	200	51
% of Overlapped Entities	Train	6	29	15	18	40
	Dev	7	38	14	18	37
	Test	8	36	13	22	39
% of Discontinuous Entities	Train	11	54	11	0	0
	Dev	13	55	10	0	0
	Test	8	52	9	0	0

Table 1: Dataset statistics. For the CLEF, CLEF-Dis, CADEC, GENIA and ACE05 datasets, we follow the settings of Dai et al. (2020), Wang and Lu (2019), Luan et al. (2019) and Lu and Roth (2015) respectively. The statistics of CLEF-Dis are sentence numbers, others are document numbers.

3.5 Training

During training, we employ multi-task learning (Caruana, 1997; Liu et al., 2017) to jointly train different parts of our model.[6] The loss function is defined as the negative log-likelihood of the two classification tasks, namely *Entity Fragment Recognition* and *Fragment Relation Prediction*:

$$\mathcal{L} = -\sum \alpha \log \boldsymbol{p}_1(y_{\text{ent}}) + \beta \log \boldsymbol{p}_2(y_{\text{rel}}), \quad (6)$$

where y_{ent} and y_{rel} denote the corresponding gold-standard labels for text spans and span pairs, α and β are the weights to control the task importance. During training, we use the BertAdam algorithm (Devlin et al., 2019) with the learning rate 5×10^{-5} to finetune BERT and 1×10^{-3} to finetune other parts of our model. The training process would terminate if the performance does not increase by 15 epochs.

4 Experimental Setup

Datasets: To evaluate our model for simultaneously recognizing overlapped and discontinuous entities, we follow prior work (Muis and Lu, 2016; Wang and Lu, 2019; Dai et al., 2020) and employ the data, called **CLEF**, from the ShARe/CLEF eHealth Evaluation Lab 2013 (Suominen et al., 2013), which consists of 199 and 99 clinical notes for training and testing. Note that Dai et al. (2020) used the full CLEF dataset in their experiments (179 for training, 20 for development and 99 for testing), while Muis and Lu (2016) and Wang and Lu (2019) used a subset of the union of the CLEF dataset and SemEval 2014 Task 7 (Pradhan et al., 2014). Concretely, they used the training set and test set of the ShARe/CLEF eHealth Evaluation Lab 2013 as the training and development set, and they also used the development set of the SemEval 2014 Task 7 as the test set. In addition, they selected only the sentences that contain at least one discontinuous entity. Finally, the training, development and test sets contain 534, 303 and 430 sentences, respectively. We call this dataset as **CLEF-Dis** in this paper. Moreover, we also follow Dai et al. (2020) to evaluate models using the **CADEC** dataset proposed by Karimi et al. (2015). We follow the setting of Dai et al. (2020) to split the dataset and conduct experiments.

To show our model is comparable with the state-of-the-art models for overlapped NER, we conduct experiments on **GENIA** (Kim et al., 2003) and **ACE05**. For the GENIA and ACE05 datasets, we employ the same experimental setting in previous works (Lu and Roth, 2015; Muis and Lu, 2017; Wang and Lu, 2018; Luan et al., 2019), where 80%, 10% and 10% sentences in 1,999 GENIA documents, and the sentences in 370, 43 and 51 ACE05 documents are used for training, development and test, respectively. The statistics of all the datasets we use in this paper is shown in Table 1.

Evaluation Metrics: In terms of evaluation metrics, we follow prior work (Lu and Roth, 2015; Muis and Lu, 2016; Wang and Lu, 2018, 2019) and employ the precision (P), recall (R) and F1-score (F1). A predicted entity is counted as true-positive if its boundary and type match those of a gold entity. For a discontinuous entity, each span should match a span of the gold entity. All F1 scores reported in Section 5 are the mean values from five runs of the same setting.

[6] Please refer to Appendix C for the effect of multi-task learning.

Related Work	Method	F1
Tang et al. (2013)	CRF, BIOHD	75.0
Tang et al. (2015)	CRF, BIOHD1234	78.3
Dai et al. (2020)	Transition-based, ELMo[7]	77.7
Our Model	Span-based, BERT	**83.2**
	– Dep-guided GCN	82.5
	– Overlap Relation	82.2
	– BERT	78.6

Table 2: Results on the CLEF dataset.

Related Work	Method	F1
Muis and Lu (2016)	Hypergraph	52.8
Wang and Lu (2019)	Hypergraph, RNN	56.1
Dai et al. (2020)	Transition-based, ELMo	62.9
Our Model	Span-based, BERT	**63.3**
	– Dep-guided GCN	62.9
	– Overlap Relation	62.6
	– BERT	56.4

Table 3: Results on the CLEF-Dis dataset.

Implementation Details: For hyper-parameters and other details, please refer to Appendix D.

5 Results and Analyses

5.1 Results on CLEF

Table 2 shows the results on the CLEF dataset. As seen, Tang et al. (2013) and Tang et al. (2015) adapted the CRF model, which is usually used for flat NER, to overlapped and discontinuous NER. They modified the `BIO` label scheme to `BIOHD` and `BIOHD1234`, which use "H" to label overlapped entity segments and "D" to label discontinuous entity segments. Surprisingly, the recently-proposed transition-based model (Dai et al., 2020) does not perform better than the CRF model (Tang et al., 2015), which may be because Tang et al. (2015) have conducted elaborate feature engineering for their model. In contrast, our model outperforms all the strong baselines with at least about 5% margin in F1. Our model does not rely on feature engineering or manually-designed transitions, which is more suitable for modern end-to-end learning.

We further perform ablation studies to investigate the effect of dependency-guided GCN and the overlapping relation, which can be removed without influencing our major goal. As shown in Table 2, after removing either of them, the F1 scores go down by 0.7% and 1.0%. The observation suggests that both dependency-guided GCN and the overlapping relation are effective for our model. Moreover, after we replace BERT with the word embeddings pretrained on PubMed (Chiu et al., 2016), the F1 score goes down by 4.6%, which demonstrates that BERT plays an important role in our model.

[7] Dai et al. (2020) found that BERT did not perform better than ELMo in their experiments.

Related Work	Method	F1
Baseline (2016)	CRF, BIOHD	60.2
Tang et al. (2018)	LSTM-CRF, Multilabel	66.3
Dai et al. (2020)	Transition-based, ELMo	69.0
Our Model	Span-based, BERT	69.5
	– Dep-guided GCN	**69.9**
	– Overlap Relation	**69.9**
	– BERT	66.8

Table 4: Results on the CADEC dataset. "Baseline (2016)" indicates Metke-Jimenez and Karimi (2016).

5.2 Results on CLEF-Dis

Table 3 shows the results on the CLEF-Dis dataset. As seen, our model outperforms the previous best model (Dai et al., 2020) by 0.4% in F1, which indicates that our model is very competitive, leading to a new state-of-the-art result on the dataset. Similarly, we further perform ablation studies to investigate the effect of dependency-guided GCN, the overlapping relation and BERT on this dataset. As shown, after removing either of the GCN or overlapping relation, the F1 score decreases by 0.4% or 0.7%, which is consistent with the observations in Table 2. In addition, to fairly compare with Wang and Lu (2019), we also replace BERT with the word embeddings pretrained on PubMed (Chiu et al., 2016). As we can see, our model also outperforms their model by 0.3%.

5.3 Results on CADEC

As shown in Table 4, Metke-Jimenez and Karimi (2016) employed the similar method in (Tang et al., 2013) by expanding the `BIO` label scheme to `BIOHD`. Tang et al. (2018) also experimented the `BIOHD` label scheme, but they found that the result of the `BIOHD`-based method was slightly worse than that of the "Multilabel" method (65.5% vs. 66.3% in F1). Compared with the method in (Metke-Jimenez and Karimi, 2016), the performance improvement might be mainly because they used deep neural networks (e.g., LSTM) instead of shallow non-neural models.

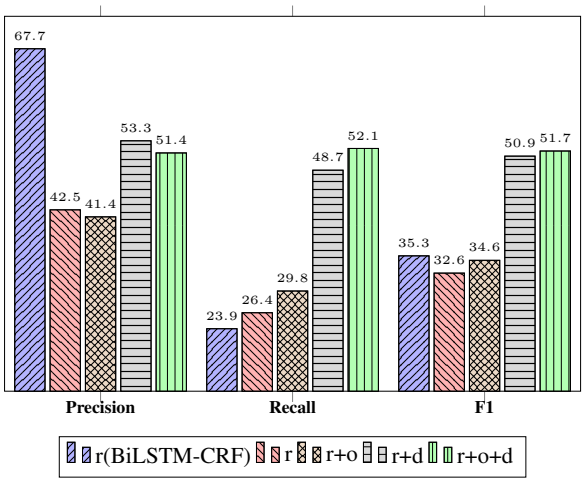


Figure 4: Result analysis based on entity types (i.e., (r)egular, (o)verlapped and (d)iscontinuous) on the CLEF-Dis dataset, comparing with BiLSTM-CRF. [8]

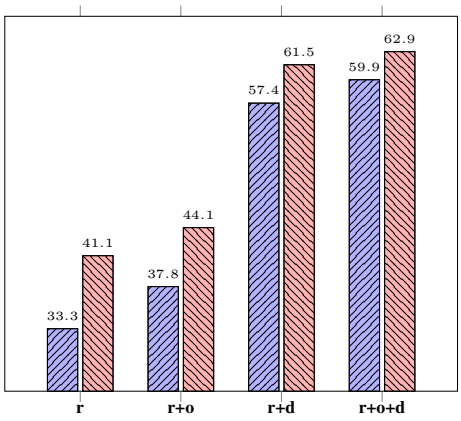


Figure 5: Result analysis based on entity types on the CLEF-Dis dataset, comparing with Dai et al. (2020) (blue).

Compared with the above baselines, the transition-based model Dai et al. (2020) is still the best. Our full model slightly outperforms the transition-based model by 0.5%. In this dataset, we do not observe mutual benefit between the dependency-guided GCN and overlapped relation prediction modules, since our model achieves better results when using them separately (69.9%) than using them jointly (69.5%). However, when using them separately, the F1 is still 0.6% higher than the one using neither of them. Without BERT, the performance of our model drops by about 3% but it is still comparable with the performances of the methods without contextualized representations.

5.4 Result Analysis based on Entity Types

Comparing with BiLSTM-CRF To show the necessity of building one model to recognize regular, overlapped and discontinuous entities simultaneously, we analyze the predicted entities in the CLEF-Dis dataset and classify them based on their types, as shown in Figure 4. In addition, we compare our model with BiLSTM-CRF (Lample et al., 2016; Ma and Hovy, 2016; Yang et al., 2018), to show our model does not influence the performance of regular NER significantly. For a fair comparison, we replace BERT with Glove (Pennington et al., 2014) and keep the setting of our model the same with the setting of the BiLSTM-CRF model used in previous work (Yang et al., 2018).

As seen, if only considering regular entities, the BiLSTM-CRF model can achieve a better performance compared with our model, especially the precision value is much higher. One likely reason might be that the BiLSTM-CRF model is capable of using the label dependence to detect entity boundaries accurately, ensuring the correctness of the recognized entities, which is closely related to the precision. Nevertheless, our model can lead to higher recall, which reduces the gap between the two models.

If considering both regular and overlapped entities, the recall of our model is greatly boosted, and thus the F1 increases concurrently. If both regular and discontinuous entities are included, the performance of our model rises significantly to 50.9% due to the large scale of discontinuous entities. When all types of entities are concerned, the F1 of our model further increases by 0.8%, indicating the effectiveness of our model in joint recognition of overlapped, discontinuous and regular entities.

Comparing with the Transition-Based Model As shown in Figure 5, we also compare our model with the transition-based model (Dai et al., 2020) based on entity types by analyzing the results from one run of experiments. Note that since we do not tune the hyper-parameters of the transition-based model elaborately, the performance is not as good as the one that they have reported. As seen, our model performs better in all of the four groups, namely regular, regular+overlapped, regular+discontinuous, regular+overlapped+discontinuous entity recognition. However, based on the observation on the bars in different groups, we find that the main superiority

[8] Many discontinuous entities are also overlapped, but we do not count them as overlapped entities in this figure.

Related Work	Method	GENIA	ACE05
Finkel and Manning (2009)	Constituency parsing	70.3	–
Lu and Roth (2015)	Hypergraph	70.3	58.7
Muis and Lu (2017)	Hypergraph	70.8	61.3
Katiyar and Cardie (2018)	Hypergraph, RNN	73.8	70.5
Wang et al. (2018)	Transition-based parsing, RNN	73.9	73.0
Ju et al. (2018)	Dynamically stacking, RNN	74.7	72.2
Zheng et al. (2019)	Boundary detection, RNN	74.7	–
Lin et al. (2019a)	Anchor-region detection, RNN, CNN	74.8	74.9
Wang and Lu (2018)	Hypergraph, RNN	75.1	74.5
Xia et al. (2019)	Multi-grained detection, RNN, ELMo	–	78.2
Fisher and Vlachos (2019)	Merge and label, BERT	–	82.4
Luan et al. (2019)	Span-based, ELMo, Coref	76.2	82.9
Wadden et al. (2019)	Span-based, BERT, Coref	77.9	–
Straková et al. (2019)	Seq2Seq, ELMo, BERT, Flair	**78.3**	**84.3**
Our Model	Span-based, BERT	77.8	83.0
	– Dep-guided GCN	77.4	82.6
	– Overlap Relation	77.4	82.7

Table 5: Comparisons with prior work on the GENIA and ACE05 datasets.

of our model comes from regular entity recognition. In recognizing overlapped entities, our model is comparable with the transition-based model, but in recognizing discontinuous entities, our model performs slightly worse than the transition-based model. This suggests that a combination of span-based and transition-based models may be a potential method for future research.

5.5 Results on GENIA and ACE05

Table 5 shows the results of the GENIA and ACE05 datasets, which include only regular and overlapped entities. Our final model achieves 77.8% and 83.0% F1s in the GENIA and ACE05 datasets, respectively. By removing the dependency-guided GCN, the model shows an averaged decrease of 0.4%, indicating the usefulness of dependency syntax information. The finding is consistent with that of the CLEF dataset. Interestingly, we note that the overlapping relation also brings a positive influence in this setting. Actually, the relation extraction architecture is not necessary for only regular and overlapped entities, because the decoding can be finished after the first entity fragment recognition step. The observation doubly demonstrates the advantage of our final model. We also compare our results with several state-of-the-art results of the previous work on the two datasets in Table 5. Only the studies with the same training, development and test divisions are listed. We can see that our model can achieve very competitive performances on both datasets. Note that Luan et al. (2019) and Wadden et al. (2019) use extra coreference resolution information, and Straková et al. (2019) exploit much richer word representations by a combination of ELMo, BERT and Flair.

6 Conclusion

In this work, we proposed an efficient and effective model to recognize both overlapped and discontinuous entities simultaneously, which can be applied to any NER dataset theoretically, since no extra assumption is required to limit the type of named entities. First, we enumerate all spans in a given sentence to determine whether they are valid entity fragments, and then relation classifications are performed to check the relationships between all fragment pairs. The results show that our model is highly competitive to the state-of-the-art models for overlapped or discontinuous NER. We have conducted detailed studies to help comprehensive understanding of our model.

Acknowledgments

We thank the reviewers for their comments and recommendation. This work is supported by the National Natural Science Foundation of China (No. 61772378), the National Key Research and Development Program of China (No. 2017YFC1200500), the Research Foundation of Ministry of Education of China (No. 18JZD015).

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.

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jindi, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical bert embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. 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 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Yixin Cao, Zikun Hu, Tat-seng Chua, Zhiyuan Liu, and Heng Ji. 2019. Low-resource name tagging learned with weakly labeled data. In *Proceedings of the 2019 Conference on EMNLP*, pages 261–270.

Rich Caruana. 1997. Multitask learning. *Machine learning*, 28(1):41–75.

Danqi Chen and Christopher D Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 740–750.

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.

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2020. An effective transition-based model for discontinuous NER. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5860–5870, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics.

Jenny Rose Finkel and Christopher D Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on EMNLP*, pages 141–150.

Joseph Fisher and Andreas Vlachos. 2019. Merge and label: A novel neural network architecture for nested ner. In *Proceedings of the 57th Annual Meeting of the ACL*, pages 5840–5850.

Radu Florian, Hongyan Jing, Nanda Kambhatla, and Imed Zitouni. 2006. Factorizing complex models: A case study in mention detection. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 473–480. Association for Computational Linguistics.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2018. Allennlp: A deep semantic natural language processing platform. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 1–6, Melbourne, Australia. Association for Computational Linguistics.

Zhijiang Guo, Yan Zhang, and Wei Lu. 2019. Attention guided graph convolutional networks for relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 241–251.

Kadri Hacioglu, Benjamin Douglas, and Ying Chen. 2005. Detection of entity mentions occurring in english and chinese text. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 379–386. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Zhanming Jie and Wei Lu. 2019. Dependency-guided lstm-crf for named entity recognition. In *Proceedings of the 2019 Conference on EMNLP*, pages 3853–3863.

Zhanming Jie, Aldrian Obaja Muis, and Wei Lu. 2017. Efficient dependency-guided named entity recognition. In *Thirty-First AAAI Conference on Artificial Intelligence*.

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 ACL*, pages 1446–1459.

Sarvnaz Karimi, Alejandro Metke-Jimenez, Madonna Kemp, and Chen Wang. 2015. Cadec: A corpus of adverse drug event annotations. *Journal of biomedical informatics*, 55:73–81.

Arzoo Katiyar and Claire Cardie. 2018. Nested named entity recognition revisited. In *Proceedings of the 2018 Conference of the North American Chapter of the ACL*, pages 861–871.

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.

Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the eighteenth international conference on machine learning, ICML*, volume 1, pages 282–289.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the NAACL*, pages 260–270.

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

Hongyu Lin, Yaojie Lu, Xianpei Han, and Le Sun. 2019a. Sequence-to-nuggets: Nested entity mention detection via anchor-region networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5182–5192.

Ying Lin, Liyuan Liu, Heng Ji, Dong Yu, and Jiawei Han. 2019b. Reliability-aware dynamic feature composition for name tagging. In *Proceedings of the 57th Annual Meeting of the ACL*, pages 165–174.

Xiao Ling and Daniel S Weld. 2012. Fine-grained entity recognition. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Fangzheng Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Pengfei Liu, Xipeng Qiu, and Xuan-Jing Huang. 2017. Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the ACL*, pages 1–10.

Xiaohua Liu, Shaodian Zhang, Furu Wei, and Ming Zhou. 2011. Recognizing named entities in tweets. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 359–367.

Wei Lu and Dan Roth. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the 2015 Conference on EMNLP*, pages 857–867.

Yi Luan, Dave Wadden, Luheng He, Amy Shah, Mari Ostendorf, and Hannaneh Hajishirzi. 2019. A general framework for information extraction using dynamic span graphs. In *Proceedings of the 2019 Conference of the North American Chapter of the ACL*, pages 3036–3046.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *Proceedings of the 54th ACL*, pages 1064–1074.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the ACL: system demonstrations*, pages 55–60.

Alejandro Metke-Jimenez and Sarvnaz Karimi. 2016. Concept identification and normalisation for adverse drug event discovery in medical forums. In *BM-DID@ ISWC*. Citeseer.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 3111–3119. Curran Associates, Inc.

Aldrian Obaja Muis and Wei Lu. 2016. Learning to recognize discontiguous entities. In *Proceedings of the 2016 Conference on EMNLP*, pages 75–84.

Aldrian Obaja Muis and Wei Lu. 2017. Labeling gaps between words: Recognizing overlapping mentions with mention separators. In *Proceedings of the 2017 Conference on EMNLP*, pages 2608–2618.

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 Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the ACL*, pages 2227–2237.

Sameer Pradhan, Wendy Chapman, Suresh Man, and Guergana Savova. 2014. Semeval-2014 task 7: Analysis of clinical text. In *Proc. of the 8th International Workshop on Semantic Evaluation (SemEval 2014*. Citeseer.

Erik Tjong Kim 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*, pages 142–147.

Jana Straková, Milan Straka, and Jan Hajic. 2019. Neural architectures for nested ner through linearization. In *Proceedings of the 57th Annual Meeting of the ACL*, pages 5326–5331.

Hanna Suominen, Sanna Salanterä, Sumithra Velupillai, Wendy W Chapman, Guergana Savova, Noemie Elhadad, Sameer Pradhan, Brett R South, Danielle L Mowery, Gareth JF Jones, et al. 2013. Overview of the share/clef ehealth evaluation lab 2013. In *International Conference of the CLEF for European Languages*, pages 212–231.

Buzhou Tang, Qingcai Chen, Xiaolong Wang, Yonghui Wu, Yaoyun Zhang, Min Jiang, Jingqi Wang, and Hua Xu. 2015. Recognizing disjoint clinical concepts in clinical text using machine learning-based methods. In *AMIA annual symposium proceedings*, volume 2015, page 1184. American Medical Informatics Association.

Buzhou Tang, Jianglu Hu, Xiaolong Wang, and Qingcai Chen. 2018. Recognizing continuous and discontinuous adverse drug reaction mentions from social media using lstm-crf. *Wireless Communications and Mobile Computing*, 2018.

Buzhou Tang, Yonghui Wu, Min Jiang, Joshua C Denny, and Hua Xu. 2013. Recognizing and encoding discorder concepts in clinical text using machine learning and vector space model. *CLEF (Working Notes)*, 665.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*, pages 5998–6008.

David Wadden, Ulme Wennberg, Yi Luan, and Hannaneh Hajishirzi. 2019. Entity, relation, and event extraction with contextualized span representations. In *Proceedings of the 2019 Conference on EMNLP*, pages 5788–5793.

Bailin Wang and Wei Lu. 2018. Neural segmental hypergraphs for overlapping mention recognition. In *Proceedings of the 2018 Conference on EMNLP*, pages 204–214.

Bailin Wang and Wei Lu. 2019. Combining spans into entities: A neural two-stage approach for recognizing discontiguous entities. In *Proceedings of the 2019 Conference on EMNLP*, pages 6217–6225.

Bailin Wang, Wei Lu, Yu Wang, and Hongxia Jin. 2018. A neural transition-based model for nested mention recognition. In *Proceedings of the 2018 Conference on EMNLP*, pages 1011–1017.

Congying Xia, Chenwei Zhang, Tao Yang, Yaliang Li, Nan Du, Xian Wu, Wei Fan, Fenglong Ma, and S Yu Philip. 2019. Multi-grained named entity recognition. In *Proceedings of the ACL*, pages 1430–1440.

Jie Yang, Shuailong Liang, and Yue Zhang. 2018. Design challenges and misconceptions in neural sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3879–3889.

Meishan Zhang, Yue Zhang, and Guohong Fu. 2016. Transition-based neural word segmentation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 421–431.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on EMNLP*, pages 2205–2215.

Changmeng Zheng, Yi Cai, Jingyun Xu, Ho-fung Leung, and Guandong Xu. 2019. A boundary-aware neural model for nested named entity recognition. In *Proceedings of the 2019 Conference on EMNLP*, pages 357–366.

Hao Zhou, Yue Zhang, Shujian Huang, and Jiajun Chen. 2015. A neural probabilistic structured-prediction model for transition-based dependency parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1213–1222.

Method	CLEF	CLEF-Dis
Word2Vec	68.5	43.5
Word2Vec+BiLSTM	78.6	56.4
ELMo	74.2	48.1
ELMo+BiLSTM	77.1	55.8
BERT	82.5	59.0
BERT+BiLSTM	83.2	63.3

Table 6: Results using different word representation methods.

A Comparing Different Settings in the Word Representation Layer

The word representation layer addresses the problem that how to transform a word into a vector for the usage of upper layers. In this paper, we investigate several common word encoders in recent NLP research to generate word representations, namely Word2Vec (Mikolov et al., 2013) (or its variants such as Glove (Pennington et al., 2014)), ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019). Given an input sentence $\boldsymbol{x} = \{x_1, x_2, ..., x_N\}$, we use different methods to represent them as vectors based on which word encoders we utilize:

- If Word2Vec is used, each word x_i will be directly transformed into a vector $\boldsymbol{h}_i$ according to the pretrained embedding lookup table. Therefore, all the words in the sentence $\boldsymbol{x}$ correspond to a matrix $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N\} \in \mathbb{R}^{N \times d_h}$, where d_h denotes the dimension of $\boldsymbol{h}_i$.

- If ELMo is used, each word x_i will first be split into characters and then input into character-level convolutional networks to obtain character-level word representations. Finally, all word representations in the sentence will be input into 3-layer BiLSTMs to generate contextualized word representations, which can also be denoted as $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N\}$

- If BERT is used, each word x_i will be converted into word pieces and then fed into a pretrained BERT module. After the BERT calculation, each sentential word may involve vectorial representations of several pieces. Here we employ the representation of the beginning word piece as the final word representation following (Wadden et al., 2019). For instance, if "fevers" is split into "fever" and "##s", the representation of "fever" is used as the whole word representation. Therefore, all the words in the sentence $\boldsymbol{x}$ can also be represented as a matrix $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N\}$

In addition, a bidirectional LSTM (BiLSTM) layer can be stacked on word encoders to further capture contextual information in the sentence, which is especially helpful for non-contextualized word representations such as Word2Vec. Concretely, the word representations $\boldsymbol{H} = \{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_N\}$ will be input into the BiLSTM layer and consumed in the forward and backward orders. Assuming that the outputs of the forward and backward LSTMs are $\overrightarrow{\boldsymbol{H}} = \{\overrightarrow{\boldsymbol{h}}_1, \overrightarrow{\boldsymbol{h}}_2, ..., \overrightarrow{\boldsymbol{h}}_N\}$ and $\overleftarrow{\boldsymbol{H}} = \{\overleftarrow{\boldsymbol{h}}_1, \overleftarrow{\boldsymbol{h}}_2, ..., \overleftarrow{\boldsymbol{h}}_N\}$ respectively. Thus, they can be concatenated (e.g., $\hat{\boldsymbol{h}}_i = [\overrightarrow{\boldsymbol{h}}_i, \overleftarrow{\boldsymbol{h}}_i]$) to compose the final word representations $\hat{\boldsymbol{H}} = \{\hat{\boldsymbol{h}}_1, \hat{\boldsymbol{h}}_2, ..., \hat{\boldsymbol{h}}_N\}$.

We investigate the effects of different word encoders and the BiLSTM layer in the experiments. As shown in Table 6, we compare the effects of different word representation methods in the CLEF and CLEF-Dis datasets, where the size of the former one is much bigger than that of the latter, in order to also investigate the impact of the data size on word representations. From the table, the first observation is that BERT is the most effective word representation method. Surprisingly, Word2Vec is more effective than ELMo, which may be because ELMo is exclusively based on characters and cannot effectively capture the whole meanings of words. Therefore, this suggests that it is better to use ELMo with Word2Vec.

Second, we find that BiLSTM is helpful in all cases, especially for Word2Vec. This may be because Word2Vec is a kind of non-contexualized word representations, which particularly needs the help of BiLSTM to capture contexual information. In contrast, BERT is not very sensitive to the help of BiLSTM as Word2Vec and ELMo, which may be because the transformer in BERT has already captured contexual information.

Third, we observe that the effect of BiLSTM is more obvious for the CLEF-Dis dataset. Considering the data sizes of the CLEF and CLEF-Dis datasets, it is more likely that small datasets need the help of BiLSTM, while big datasets are less sensitive to the BiLSTM and BERT is usually enough for them to build word representations.

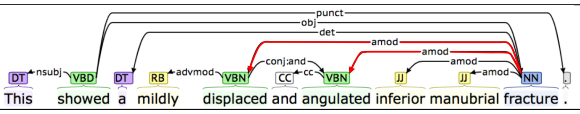
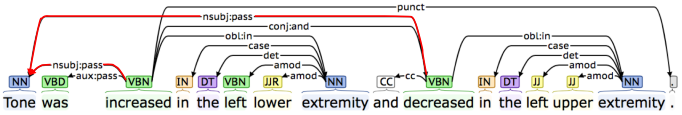
Examples	Dependency Graphs
This showed a mildly [**displaced**]$_1$ and [**angulated**]$_2$ inferior manubrial [[**fracture**]$_1$]$_2$.	
[[**Tone**]$_1$]$_2$ was [**increased**]$_1$ in the left lower extremity and [**decreased**]$_2$ in the left upper extremity.	

Table 7: Case Studies. Bold words with the same number belong to the same entity.

Method	P	R	F1
EFR	81.2	79.6	80.4
EFR(+FRP)	81.4	80.1	80.7

Table 8: Effect of joint training between entity fragment recognition (EFR) and fragment relation prediction (FRP) on the CLEF-Dis dataset. P, R and F1 are the results for EFR.

B Case Studies

To understand how syntax information helps our model to identify discontinuous or overlapped entities, we offer two examples in the CLEF dataset for illustration, as shown in Table 7. Both the two examples are failed in the model without using dependency information, but are correctly recognized in our final model. In the first example, the fragments "displaced" and "fracture" of the same entity are far away from each other in the original sentence, while they are directly connected in the dependency graph. Similarly, in the second example, the distance between "Tone" and "decreased" is 9 in the sentence, while their dependency distance is only 1. These dependency connections can be directly modeled in dependency-guided GCN, thus, resulting in strong clues for the NER, which makes our final model work.

C Effect of Joint Training

As mentioned in Section 3.5, we employ multi-task learning to jointly train our model between two tasks, namely entity fragment recognition and fragment relation prediction. Therefore, it is interesting to show the effect of joint training by observing the performance changes of the entity fragment recognition (EFR) task before and after adding the fragment relation prediction (FRP) task. As seen in Table 8, the F1 of entity fragment recognition increases by 0.3% after adding the FRP task, which shows that the FRP task could improve the EFR task. This suggests that the interaction between entity fragment recognition and fragment relation prediction could benefit our model, which also indicates that end-to-end modeling is more desirable.

	CLEF	CADEC	GENIA	ACE05
d_h	400	400	768	768
N_{head}	4	4	4	4
l	2	2	2	1
d_f	20	20	64	64
d_s	860	860	1,684	1,684
MLP Layer	1	1	2	2
MLP Size	150	150	150	150
α	1.0	1.0	1.0	1.0
β	1.0	1.0	1.0	0.6

Table 9: Main hyper-parameter settings in our model for all the datasets. d_h–Section 3.1; N_{head}, l and d_f–Section 3.2; d_s–Section 3.3; α and β–Section 3.5. Note that the hyper-parameter settings in the CLEF-Dis dataset is the same as those in the CLEF dataset.

D Implementation Details

Our model is implemented based on AllenNLP (Gardner et al., 2018). The number of parameters is about 117M plus BERT. We use one GPU of NVIDIA Tesla V100 to train the model, which occupies about 10GB memories. The training time for one epoch is between 2∼6 minutes on different datasets.

Table 9 shows the main hyper-parameter values in our model. We tune the hyper-parameters based on the results of about 5 trials on development sets. Below are the ranges tried for the hyper-parameters: the GCN layer l (1, 2), the GCN head N_{head} (2, 4), the GCN output size d_f (20, 48, 64), the MLP layer (1, 2), the MLP size (100, 150, 200), the loss weight α and β (0.6, 0.8, 1.0). Since we employ the BERT$_{BASE}$, the dimension d_h of word representations is 768 except in the CLEF and CADEC

datasets, where we use a BiLSTM layer on top of BERT to obtain word representations since we observe performance improvements. We try 200 and 400 hidden units for the BiLSTM layer.

Considering the domains of the datasets, we employ clinical BERT[1] (Alsentzer et al., 2019), SciBERT[2] (Beltagy et al., 2019) and Google BERT[3] (Devlin et al., 2019) for the CLEF (and CADEC), GENIA and ACE05 datasets, respectively. In addition, since our model needs syntax information for dependency-guided GCN, but the datasets do not contain gold syntax annotations, we utilize the Stanford CoreNLP toolkit (Manning et al., 2014) to perform dependency parsing.

[1] https://github.com/EmilyAlsentzer/clinicalBERT
[2] https://github.com/allenai/scibert
[3] https://github.com/google-research/bert