The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21)

Rethinking Boundaries: End-To-End Recognition of Discontinuous Mentions with Pointer Networks

Hao Fei,[1] Donghong Ji,[1] Bobo Li,[1] Yijiang Liu,[1] Yafeng Ren,[2] Fei Li[1]*

[1] Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, Wuhan, China

[2] Guangdong University of Foreign Studies, Guangzhou, China

{hao.fei, dhji, boboli, cslyj, renyafeng}@whu.edu.cn, foxlf823@gmail.com

Abstract

A majority of research interests in irregular (e.g., nested or discontinuous) named entity recognition (NER) have been paid on nested entities, while discontinuous entities received limited attention. Existing work for discontinuous NER, however, either suffers from decoding ambiguity or predicting using token-level local features. In this work, we present an innovative model for discontinuous NER based on pointer networks, where the pointer simultaneously decides whether a token at each decoding frame constitutes an entity mention and where the next constituent token is. Our model has three major merits compared with previous work: (1) The pointer mechanism is memory-augmented, which enhances the mention boundary detection and interactions between the current decision and prior recognized mentions. (2) The encoder-decoder architecture can linearize the complexity of structure prediction, and thus reduce search costs. (3) The model makes every decision using global information, i.e., by consulting all the input, encoder and previous decoder output in a global view. Experimental results on the CADEC and ShARe13 datasets show that our model outperforms flat and hypergraph models as well as a state-of-the-art transition-based model for discontinuous NER. Further in-depth analysis demonstrates that our model performs well in recognizing various entities including flat, overlapping and discontinuous ones. More crucially, our model is effective on boundary detection, which is the kernel source to NER.

Introduction

Named Entity Recognition (NER), which aims to detect the span as well as the semantic category of an entity mention from text, has long been a fundamental task in natural language processing (NLP) (Florian et al. 2004; Sutton, McCallum, and Rohanimanesh 2007; Collobert et al. 2011; Lample et al. 2016; Yu, Bohnet, and Poesio 2020). Most traditional methods formalize NER as a sequence labeling task (Lafferty, Mccallum, and Pereira 2001; Collobert et al. 2011; Lample et al. 2016), assigning a single label to each token. Those methods, however, only solve flat NER where a token can only be assigned to one mention, while they are incapable of handling irregular NER where entity mentions may be nested, overlapped or discontinuous (Lu and Roth 2015; Li et al. 2018; Wang and Lu 2019).

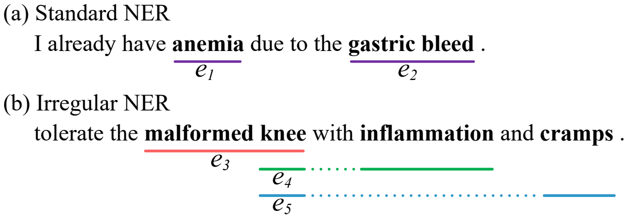

(a) Standard NER
I already have **anemia** due to the **gastric bleed** .
e_1
e_2
(b) Irregular NER
tolerate the **malformed knee** with **inflammation** and **cramps** .
e_3
e_4
e_5

Figure 1: Examples for standard NER (a) and irregular NER (b). Different entity mentions (e_i) are in distinct colors. e_1 and e_2 are regular entity mentions. e_3 overlaps with two discontinuous mentions e_4 and e_5 at the token “knee”.

Nested NER (Kim et al. 2003), as one of common irregular NER problems, has drawn much research attention (Kim et al. 2003; Alex, Haddow, and Grover 2007; Finkel and Manning 2009; Lu and Roth 2015; Katiyar and Cardie 2018; Yu, Bohnet, and Poesio 2020). Yet discontinuous NER, where entities may consist of a discontinuous sequence of words, has been neglected by most of previous work. Discontinuous NER is ubiquitous in many practical scenarios, especially in biomedical and clinical domain (Tang et al. 2013; Xu et al. 2015; Tang et al. 2018). Another character of discontinuous entity mentions is that there certain words may be overlapped, and thus recognizing discontinuous entities also needs to handle the nested mentions. In Figure 1, we exemplify some cases of irregular entity mentions. For example, the mentions e_4 and e_5 are discontinuous, and they share a common word “knee” with the mention e_3. Apparently, discontinuous NER poses more challenges than traditional NER (Muis and Lu 2016; Wang and Lu 2019; Dai et al. 2020).

Existing works for discontinuous NER can be roughly divided into several categories. The first one is still based on the sequence labeling architecture but extending the *BIO* label scheme to more complex label schemes such as *BIOHD* (Tang et al. 2013) to represent discontinuous and overlapping structures (Xu et al. 2015; Metke-Jimenez and Karimi 2016; Tang et al. 2018). Other approaches aim to build more effective inference systems by modeling the structure as a whole graph, e.g. such as hypergraph models

[*] Corresponding author

(Muis and Lu 2016), the two-stage method (Wang and Lu 2019) and the transition system (Dai et al. 2020). Nevertheless, label-scheme-extension approaches (Tang et al. 2013) or hypergraph models (Muis and Lu 2016) may suffer from decoding ambiguity, while the two-stage method (Wang and Lu 2019) disables the interactions between the entity segment recognition and combination stages. The current state-of-the-art transition system (Dai et al. 2020) may suffer from the long-distance dependency issue (McDonald and Nivre 2011; Kurita and Søgaard 2019) due to its local and incremental decision-making mechanism.

In this work, we present a novel solution for discontinuous NER[1] using pointer networks (Vinyals, Fortunato, and Jaitly 2015), which have been demonstrated to be effective for a wide range of NLP tasks (See, Liu, and Manning 2017; Ma et al. 2018; Li, Ye, and Shang 2019). As shown in Figure 2, we establish our model based on the encoder-decoder architecture, where the decoder is equipped with a pointer network to indicate whether a token at each decoding frame constitute a mention and where the next constituent token is.

Unlike previous methods that exclusively make token-level predictions, our pointer-network-based architecture makes decisions by consulting all the input elements in a global view. Moreover, our model linearizes the complexity of structure prediction, resulting in a significant decrease in search costs compared with previous methods. Last but not least, we believe that the most difficult problem for discontinuous NER lies in recognizing the complex boundaries of entities. We thus propose a memory-augmented pointer mechanism (cf. Figure 3), where each pointer decision can be made based on prior recognized mentions cached in a memory, allowing to better capture informative cues for the boundary detection of the current partial mention.

We evaluate our model on two benchmark datasets for discontinuous NER, i.e., CADEC (Karimi et al. 2015) and ShARe13 (Pradhan et al. 2013). Results show that our model achieves much better performance than all the baselines (Muis and Lu 2016; Tang et al. 2018; Wang and Lu 2019; Dai et al. 2020), demonstrating its effectiveness on recognizing discontinuous mentions as well as regular and overlapped mentions. Further analyses reveal that our model is more powerful on detecting mention boundaries. We summarize the contributions of this work and the strengths of our method as follows:

• We are the first to introduce a pointer-network-based model for discontinuous NER. Our model can avoid making decisions only based on local information and fully utilize the global information via the pointer network (Vinyals, Fortunato, and Jaitly 2015).

• Benefiting from the linear complexity of the encoder-decoder architecture (Sutskever, Vinyals, and Le 2014), our model has a lower decoding complexity than those of hypergraph-based methods (Muis and Lu 2016; Wang and Lu 2019). In addition, it can detect both regular and irregular entity mentions in an entire end-to-end fashion without any constraint used in previous work (Muis and Lu 2016; Wang and Lu 2019).

• We propose a memory-augmented pointer mechanism to encourage the current pointer to interact with prior recognized mentions, in order to effectively capture informative clues for improving the boundary detection.

• Our framework wins state-of-the-art performances on two benchmark datasets, which demonstrates that it is effective for both regular and irregular NER.

Related Work

Named Entity Recognition (NER) as the fundamental NLP task has drawn much research attention (Florian et al. 2004; Sutton, McCallum, and Rohanimanesh 2007; Collobert et al. 2011; Lample et al. 2016; Yu, Bohnet, and Poesio 2020). Traditional NER (i.e., flat or regular NER) casts the task as sequential labeling, i.e., assigning each token with a label (e.g., *BIO* tagging scheme). On the other hand, there can be irregular NER, which cannot be solved by those flat NER methods. One of the common tasks for irregular NER is nested NER, which aims to extract the overlapping entities (Kim et al. 2003; Alex, Haddow, and Grover 2007; Finkel and Manning 2009; Lu and Roth 2015; Katiyar and Cardie 2018; Fei, Ren, and Ji 2020a,b) Nevertheless, another important task for irregular NER, namely discontinuous NER has not received much attention yet (Pradhan et al. 2013). In addition, since discontinuous entity mentions are often partially overlapped with each other, discontinuous NER covers nested NER but is more challenging to some extent (Tang et al. 2013).

Existing methods for discontinuous NER can be mainly categorized into two classes: label-extension methods and hypergraph-based methods. One representative work of label-extension methods is proposed by Tang et al. (2013), in which the *BIO* label scheme is extended to the *BIOHD* label scheme with special labels to represent discontinuous structures. This approach is then widely extended by several other studies (Xu et al. 2015; Metke-Jimenez and Karimi 2016; Tang et al. 2018). By contrast, hypergraph-based methods (Lu and Roth 2015) cast the task as structural prediction, detecting the combination of mentions within a sentence. For example, Muis and Lu (2016) proposed a hypergraph model for extracting the discontinuous and overlapping mentions. Unfortunately, all the above methods suffer from more or less decoding ambiguity (Tang et al. 2013; Muis and Lu 2016). Recently, Wang and Lu (2019), introduce a two-stage method, where the first stage is hypergraph-based entity segment recognition and the second is relation-based segment combination. Because such design makes their method not end-to-end, two separated stages may result in interaction deficiency.

The latest attempt dealing with discontinuous NER is Dai et al. (2020). They designed a transition-based model for extracting irregular mentions in an end-to-end manner, achieving higher performance than prior models. However, there are two main limitations in their model. First, due to the nature of transition-based systems (i.e., incremental decision-making by local features), their model may suffer from the long-range dependency issue (McDonald and Nivre 2011;

[1] Our system also support the recognition of regular, nested or overlapped entities.

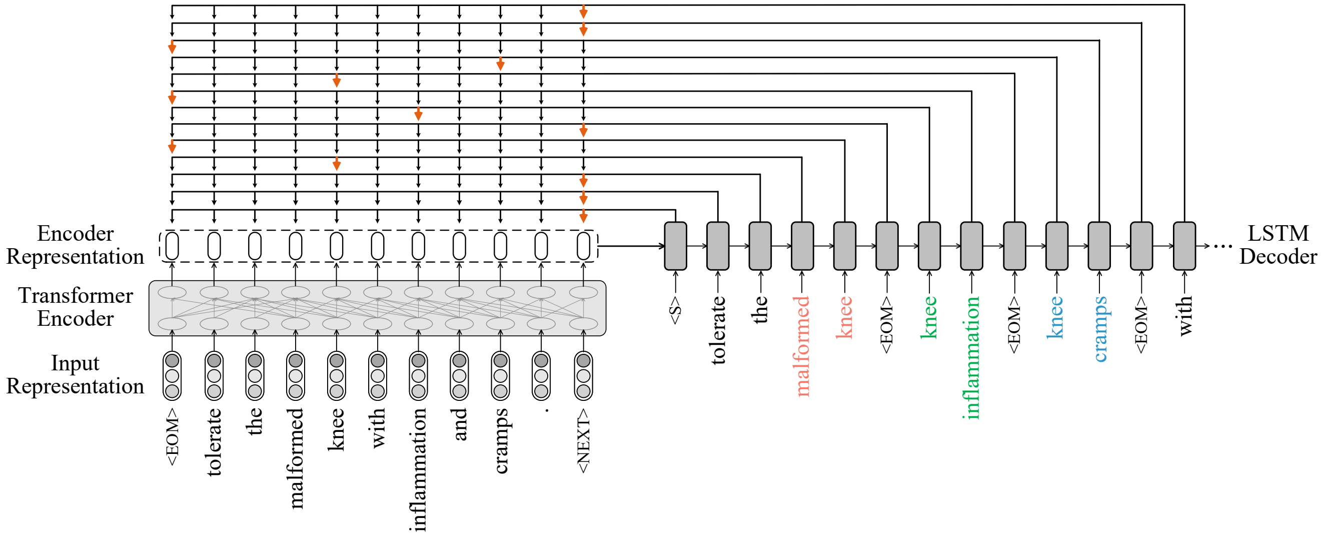


Figure 2: The main architecture of our model, which includes an input layer and a transformer layer in the encoder side, and a LSTM-based (memory-augmented) pointer network in the decoder side. The decoding process starts with <S>. If the pointer network points to <NEXT>, the next token will be the input of the decoder (e.g., if <S>→<NEXT>, the next input is "tolerate"). If the pointer network points to a real word (e.g., "malformed"→"knee"), the input token and pointed token belong to an entity mention (e.g., "malformed knee"). If the pointer network points to <EOM> (e.g., "knee"→<EOM>), the input token is the end token of a mention. Through such method, our model in this case can decode out three entity mentions by the following pointers: "malformed"→"knee"→<EOM>, "knee→"inflammation"→<EOM> and "knee"→"cramps"→<EOM>, respectively.

Kurita and Søgaard 2019). Second, limited by the designing of the shift-reduce system, their model may fail to detect some irregular mentions. For example, their model can simultaneously recognize e_4 and e_5 in Figure 1 but not e_3 or e_4 simultaneously. In this work, we present a better solution for fully end-to-end discontinuous NER using pointer networks. Our model is able to make predictions in a global view, and meanwhile recognize all types of irregular entity mentions without any constraint.

Framework

Method Overview

Given an input sentence $s = \{w_1, \cdots, w_n\}$, our system outputs a list of mentions $Y = \{y_1, \cdots, y_m\}$, where each mention $y_k = [a, \cdots, b]$ $(1 \leq a < b \leq n)$ is represented as a list of ordered token indexes.[2] We design two sentinel tokens, namely <EOM> (indicating the end of the current mention) and <NEXT> (indicating the next token will be the decoder input), to insert into the head and tail of the sentence s.

We adopt an encoder-decoder paradigm (Sutskever, Vinyals, and Le 2014), which ensures the flexibility that the output sequence can be variable-length. In Figure 2, we illustrate the overall encoder-decoder framework of our model. First, input tokens are projected into vectorial representations. Then the encoder generates the contextual representation for each token. Afterwards, the decoder takes a special token <S> as input at the first-time frame and all the tokens in the input sentence s at other time frames sequentially. During decoding, the pointer of the decoder will direct to a token w_t in the input sentence s.[3]

- If w_t is a word token (not the sentinel tokens such as <EOM> or <NEXT>), our system will create a partial mention if there is no previous created mention. When there has already existed a created mention y_k, our system will add the index t of the token w_t into y_k. At the next decoding frame, the decoder input will become w_t.
- If w_t is <EOM>, the current mention y_k will be finished and stored into the decoding output set Y. At the next decoding frame, the decoder input will be <EOM>.
- If w_t is <NEXT>, it means that w_t does not belong to any mention. Therefore the decoder input for the next decoding frame will become 1) the next token of the input token where the resulting pointer is <NEXT>, or 2) the second token of the recognized mention.
- The decoding procedure will terminate if all the tokens in the input sentence have been consumed. Finally, the mentions stored in Y are used as decoding outputs.

Input Representation and Contextual Encoder

The input representations are derived from three sources. We first obtain the vectorial representation $\boldsymbol{x}_t^w$ of each word w_t from pre-trained embeddings (Bojanowski et al. 2017). We then represent the absolute position information (Zeng et al. 2014) for each word as an embedding $\boldsymbol{x}_t^p$. Moreover,

[2] Following (Dai et al. 2020), only entity mention boundaries are necessary for discontinuous NER, while entity mention types are not required. Note that it is convenient for our model to support entity mention type recognition by adding a softmax classifier on the mention representation layer.

[3] The pointer for <S> is forced to direct to <NEXT>.

a convolutional neural network (CNN) is used to encode the characters inside each word into a character-level word representation $\boldsymbol{x}_t^c$. Finally, the total input representation is the concatenation of all above representations:

$$\boldsymbol{x}_t = [\boldsymbol{x}_t^w; \boldsymbol{x}_t^p; \boldsymbol{x}_t^c] \, . \tag{1}$$

To encode contextual information into word representations, we leverage the Transformer (Trm) (Vaswani et al. 2017) that has shown to be prominent on learning the interaction between each pair of input words, leading to better contextualized word representations. Formalized, the input and output of the Transformer encoder can be defined as:

$$\boldsymbol{h}_1, \cdots, \boldsymbol{h}_n = \mathrm{Trm}(\boldsymbol{x}_1, \cdots, \boldsymbol{x}_n) \, . \tag{2}$$

Pointer-Network-Based Decoder

Backbone Decoder We employ the LSTM (Hochreiter and Schmidhuber 1997) as the decoder of our model. At each decoding time frame i, the LSTM cell produces the decoding representation based on three parts of inputs:

$$\boldsymbol{s}_i = \mathrm{LSTM}(\boldsymbol{x}_i \oplus \boldsymbol{h}_i \oplus \boldsymbol{s}_{i-1}) \, , \tag{3}$$

where $\oplus$ refers to the concatenating operation. $\boldsymbol{x}_i$ and $\boldsymbol{h}_i$ are the corresponding input token representation and encoder representation. $\boldsymbol{s}_i$ and $\boldsymbol{s}_{i-1}$ are the current and last decoding representation, respectively.

Basic Pointer Mechanism The basic pointer mechanism in our framework is based on the vanilla pointer network (Vinyals, Fortunato, and Jaitly 2015). Technically, given the encoder representations $[\boldsymbol{h}_1, \cdots, \boldsymbol{h}_n]$ of input tokens and the current decoding representation $\boldsymbol{s}_i$, we calculate and normalize the relatedness score between $\boldsymbol{s}_i$ and each $\boldsymbol{h}_j$:

$$\begin{aligned} v_{ij} &= \mathrm{Score}(\boldsymbol{s}_i, \boldsymbol{h}_j), \\ &= \mathrm{Tanh}(\boldsymbol{s}_i^T \boldsymbol{W} \boldsymbol{h}_j + \boldsymbol{U}_1^T \boldsymbol{s}_i + \boldsymbol{U}_2^T \boldsymbol{h}_j + b) \, , \\ o_{ij} &= \mathrm{Softmax}(v_{ij}) \, , \quad j = [1, \cdots, n] \, . \end{aligned} \tag{4}$$

We then take the position j^* with the maximal relatedness probability o_{ij^*} as the output of the i-th decoding, formalized as:

$$P_i = j^* = \underset{1 \le j \le n}{\mathrm{Argmax}}(o_{i1}, ..., o_{in}) \, , \tag{5}$$

where P_i denotes the position that the current pointer directs to. Note that since each pointer decision is made by consulting all input tokens, our model can utilize the global information.

Memory-Augmented Pointer Mechanism

Overview To enhance the boundary detection for NER, we introduce a memory-augmented pointer mechanism as illustrated in Figure 3. Our motivation is to enrich the input of the pointer network by considering the prior recognized mentions. Specifically, we build a memory to store the representations of prior recognized mentions and encourage the current pointer to make interactions with the representations in the memory.

Mention Representation For each mention $y_k = [a, \cdots, b]$ $(1 \le a < b \le n)$, we can construct its representation $\boldsymbol{r}_k$ as below:

$$\boldsymbol{r}_k = \mathrm{Att}([\boldsymbol{h}_a, \cdots, \boldsymbol{h}_b]) \oplus \mathrm{Att}([\boldsymbol{s}_a, \cdots, \boldsymbol{s}_b]) \, , \tag{6}$$

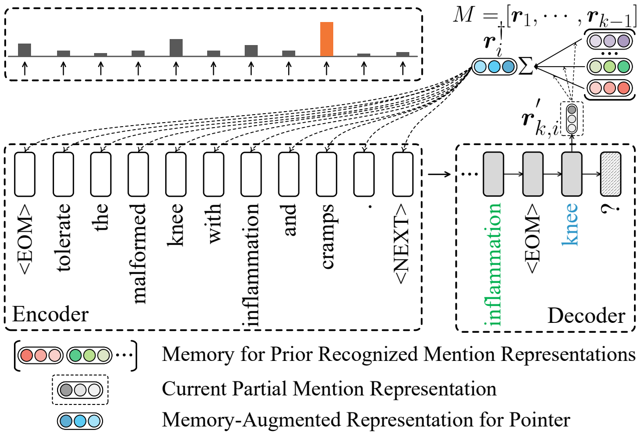

$M = [\boldsymbol{r}_1, \cdots, \boldsymbol{r}_{k-1}]$
$\boldsymbol{r}_i^\dagger$
Σ
$\boldsymbol{r}'_{k,i}$
<EOM>
tolerate
the
malformed
knee
with
inflammation
and
cramps
.
<NEXT>
Encoder
inflammation
<EOM>
knee
?
Decoder
Memory for Prior Recognized Mention Representations
Current Partial Mention Representation
Memory-Augmented Representation for Pointer

Figure 3: Illustration of our memory-augmented pointer mechanism. The orange bar in the histogram denotes the current pointing.

where $\mathrm{Att}(\cdot)$ refers to the attention mechanism (Bahdanau, Cho, and Bengio 2015; Luong, Pham, and Manning 2015):

$$\mu_i = \boldsymbol{v}^T \mathrm{Tanh}(\boldsymbol{W} \boldsymbol{h}_i) \, , \tag{7}$$

$$\alpha_i = \mathrm{Softmax}(\mu_i), \quad i = [a, \cdots, b] \, , \tag{8}$$

$$\mathrm{Att}([\boldsymbol{h}_a, \cdots, \boldsymbol{h}_b]) = \sum \alpha_i \boldsymbol{h}_i \, , \tag{9}$$

where $\boldsymbol{v}$ and $\boldsymbol{W}$ are learnable parameters. Note that no matter a mention is partially or fully recognized, we can create its representation though Equation 6. For a recognized mention $y_k = [a, \cdots, b]$, we store its representation $\boldsymbol{r}_k$ into the memory M. For a partial mention $y_k = [a, \cdots, i]$ that starts at a and ends at i, we define its representation as $\boldsymbol{r}'_{k,i}$.

Updating Representations via Memory With the memory that stores the representations of prior recognized mentions, namely $M = [\boldsymbol{r}_1, \cdots, \boldsymbol{r}_{k-1}]$, the current partial mention representation $\boldsymbol{r}'_{k,i}$ can make interactions with the representations in the memory as below:

$$u_m = \mathrm{Tanh}(\boldsymbol{r}_m^T \boldsymbol{W} \boldsymbol{r}'_{k,i}) \, , \tag{10}$$

$$\beta_m = \mathrm{Softmax}(u_m) \, , \quad m = [1, \cdots, k-1] \, , \tag{11}$$

$$\boldsymbol{r}_i^\dagger = \textstyle\sum_{m=1}^{k-1} \beta_m \boldsymbol{r}'_{k,i} \, , \tag{12}$$

where $\boldsymbol{W}$ is the weight matrix, u_m and β_m are the attention score and probability for the m-th mention in the memory, and $\boldsymbol{r}_i^\dagger$ is the updated representation for the partial mention $y_k = [a, \cdots, i]$ that exactly ends at the position i. Since $\boldsymbol{r}_i^\dagger$ can be aligned with the decoding representation $\boldsymbol{s}_i$ at the position i, we could replace $\boldsymbol{s}_i$ with $\boldsymbol{r}_i^\dagger$ in Equation 4 as below:

$$v_{ij} = \mathrm{Score}(\boldsymbol{r}_i^\dagger, \boldsymbol{h}_j) \, . \tag{13}$$

With the memory-augmented pointer mechanism, the model may capture informative cues for the boundary detection of the current partial mention. For example, in Figure 3, the pointer at the position "*knee*" will not direct to "*inflammation*" again, if the pointer module knows that there exists a recognized mention "*knee inflammation*". Thus, such awareness helps the pointer direct to the correct word "*cramps*".

Dataset	Document				Sentence		Mention		Discontinuous Mention		
	#All	#Train	#Dev	#Test	#All	Avg.Len.	#All	Avg.Len.	#All	#Ovlp.	Avg.Itv.Len.
CADEC	1,250	875	187	188	7,597	14.2	6,318	2.7	675(10.7%)	594(88.0%)	3.3
ShARe13	298	180	19	99	18,767	12.9	11,161	1.8	1,090(9.7%)	508(46.6%)	3.0

Table 1: Dataset statistics. '#' denotes the amount. 'Ovlp.' denotes the overlapping mentions among the discontinuous ones. 'Avg.Itv.Len.' means the averaged interval length of the spans in discontinuous mentions.

Training Details

Training Objective For each sentence s=$\{w_1, \cdots, w_n\}$, our training target is to minimize the negative log-likelihood losses with regards to the corresponding gold pointers, formalized as:

$$\mathcal{L} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{n+2}\hat{o}_{ij}\text{log}o_{ij} \, , \tag{14}$$

where N is the number of the decoding frame for the sentence s, $\hat{o}_{ij}$ and o_{ij} are the gold and predicted pointer probabilities (cf. Equation 4) at the i-th decoding frame. Note that there are $n+2$ candidates for the pointer at each decoding frame, since we add two sentinel tokens `<EOM>` and `<NEXT>` to the input sentence.

Teacher Forcing and Dynamic Sampling Following previous encoder-decoder work, we adopt the teacher forcing strategy (Williams and Zipser 1989), maximizing the likelihood of each predicted pointer under the gold standard. That is, we feed the gold-standard input into the decoder at each decoding step. During inference, the input for the next decoding step will come from the last prediction. However, since the model may produce incorrect predictions, the decoder inputs during training and inference are inconsistent, thus yielding biases between training and inference. We therefore employ the dynamic sampling strategy (Wang et al. 2017; Yu, Zhang, and Fu 2018) to alleviate this problem. We define an initial threshold $\gamma \in [0, 1]$ which gradually decreases within the training process. At each decoding step, we generate a random value $\tau \in [0, 1]$. If τ is less than γ, the input will be the gold standard, and otherwise the input will be the predicted one. Thus, the training process can be gradually changed from a "gold-biased" process towards a "predicted-biased" process.

Experiment

Setup

Datasets and Resources We experiment on two datasets for discontinuous NER, namely CADEC (Karimi et al. 2015) and ShARe13 (Pradhan et al. 2013), both of which are derived from biomedical or clinical domain documents. These datasets contain about 10% discontinuous entity mentions, and a large proportion of discontinuous mentions are overlapped with each other. In Table 1, we present the detailed statistics of two datasets. In terms of word representations, we use the pre-trained Fasttext embeddings (Bojanowski et al. 2017). Moreover, we use the contextualized word embeddings, ELMo (Peters et al. 2018), which is also used in our baseline (Dai et al. 2020). We further employ BioBERT (Lee et al. 2020; Fei et al. 2020), a BERT model (Devlin et al. 2019) pre-trained using biomedical text.

Hyper-Parameters The dimensions of word embeddings, position embeddings and character representations are 300, 30 and 50 respectively. We use the 3-layer Transformer with a 768-dimension hidden size as encoder. The dimensions of all the other intermediate representations are set as 300. The kernel sizes of CNN are [3,4,5]. We adopt the Adam optimizer with an initial learning rate as 1e-4. The mini-batch size is set as 16. Moreover, the initial value of γ is set as 0.85 according to the development experiments.

Baselines and Evaluation Metrics We make comparisons with three types of prior approaches: (1) Flat NER (Lample et al. 2016), which uses the BiLSTM-CRF model and *BIO* label scheme. Thus, it does not support discontinuous NER. We follow Dai et al. (2020), replacing a discontinuous mention with the shortest span that fully covers it and merging overlapping mentions into a single mention that covers them all. (2) *BIOHD* labeling methods (Metke-Jimenez and Karimi 2016; Tang et al. 2018), which also uses the BiLSTM-CRF model but extends the *BIO* label scheme to the *BIOHD* label scheme. (3) Hypergraph-based approaches (Muis and Lu 2016; Wang and Lu 2019). (4) Transition-based approach (Dai et al. 2020).

Precision (P), recall (R) and F1 scores are used as the measurement metrics. We run each experiment for 5 times and report the averaged value. We denote our model with the basic pointer as *Ptr*, and the one with the memory-augmented pointer as *MAPtr*.

Results and Discussion

Main Results In Table 2, we find that when flattening these irregular mentions into flat ones, the BiLSTM+CRF model can achieve competitive results compared with the discontinuous NER models. This implies that irregular mentions bring more difficulties on boundary detection. Moreover, we find that the *BIOHD* labeling methods can obtain better recall, the hypergraph methods achieve higher precision, and the transition model obtains better overall results with the help of ELMo or BioBERT. Furthermore, our models can significantly outperform all these baselines. Especially, our memory-augmented pointer model (MAPtr) wins the best performances on all metrics, i.e., with 64.8 F1 % and 76.3% F1 on CADEC and ShARe13 datasets. We next examine the effects in the help of contextual pre-trained word representation. Compared with Dai et al. (2020), we see that their model receives the biggest improvement with ELMo,

		CADEC			ShARe13			Decode
		P	R	F1	P	R	F1	*word/sec*
• Flat NER	BiLSTM+CRF	65.3*	58.5*	61.8*	78.5*	66.6*	70.0*	**1,432**
• BIOHD Labelling	Metke-Jimenez and Karimi (2016)	64.4*	56.5*	60.2*	-	-	-	-
	Tang et al. (2018)	67.8*	64.9*	66.3*	78.9	69.2	73.7	-
• Graph Modeling	Muis and Lu (2016)	**72.1***	48.4*	58.0*	**83.9***	60.4*	70.3*	540
	Wang and Lu (2019)	71.8	50.2	60.1	82.0	68.6	73.7	318
• Transition System	Dai et al. (2020)	63.2	56.8	61.3	76.5	70.2	74.0	727
	+ELMo	68.9*	**69.0***	**69.0*(+7.7)**	78.9*	**73.0***	**77.7*(+3.7)**	705
	+BioBERT	69.0	67.8	68.4 (+7.1)	79.3	71.7	75.6 (+1.6)	680
• Ours	Ptr	71.1	54.5	61.9	82.2	69.4	74.7	1,050
	MAPtr	73.5	59.8	64.8	84.7	72.6	76.3	983
	+ELMo	74.3	70.6	71.0 (+6.2)	86.7	75.9	79.5 (+3.2)	920
	+BioBERT	**75.5**	**71.8**	**72.4 (+7.6)**	**87.9**	**77.2**	**80.3 (+4.0)**	887

Table 2: Main results. The values with * are retrieved from Dai et al. (2020) and others are based on our implementations. The values above underlines are the best results without using contextualized word representations. In the brackets are the improvements by contextualized word representations.

	CADEC	ShARe13
Ours(MAPtr)	**64.8**	**76.3**
w/o Char	64.1	76.0
w/o Position	63.2	75.4
w/o Memory-augmented Ptr	61.9	74.7
BiLSTM encoder	62.2	75.0
w/o Dynamic sampling	64.0	75.8

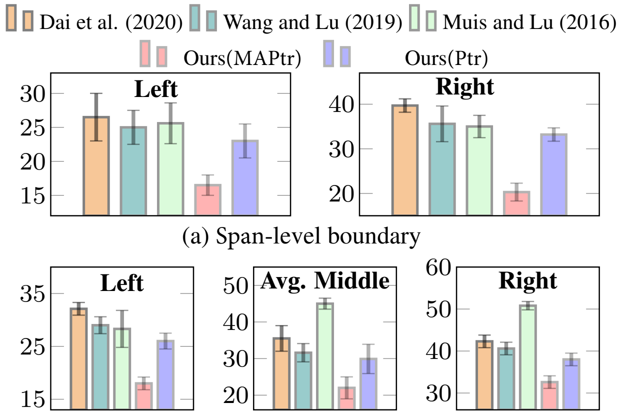
Table 3: The results (F1) of ablation studies.

	CADEC			ShARe13		
	P	R	F1	P	R	F1
Tang et al. (2018)	60.0	32.6	51.2	53.6	40.5	46.3
Muis and Lu (2016)	69.5	43.2	53.3	82.3	47.4	60.2
Wang and Lu (2019)	67.2	49.6	58.2	81.0	55.7	61.8
Dai et al. (2020)	53.2	50.1	52.7	67.2	53.8	56.3
+ELMo	66.5	64.3	65.4	70.5	56.8	62.9
+BioBERT	64.2	58.3	62.4	71.2	54.4	61.0
Ours(MAPtr)	70.6	53.5	61.2	80.4	56.1	62.4
+ELMo	71.3	65.8	66.7	82.6	58.3	63.8
+BioBERT	**72.2**	**66.3**	**68.0**	**83.1**	**60.4**	**64.8**

Table 4: Results on sentences that have at least one discontinuous mention.

while our model benefits the most from BioBERT. Last but not least, our models show higher decoding speeds.

Ablation Studies We use MAPtr as the target for ablation studies. As shown in Table 3, both character representations and position embeddings are effective, but the latter ones are more prominent since intuitively the positions are more informative for pointer networks. After we remove the memory-augmented pointer mechanism, the performance drops drastically. This suggests that the memory-augmented pointer mechanism is very crucial for our model and task. Besides, when replacing the Transformer encoder with BiLSTM, there are considerable drops in F1. Last, we



(a) Span-level boundary

(b) Mention-level boundary

Figure 4: Error rates for boundary detection on ShARe13.

observe that our model also benefits from the dynamic sampling strategy.

Effectiveness for Discontinuous NER Following Dai et al. (2020), we experiment on the subset of the original test set where each sentence has at least one discontinuous mention. As shown in Table 4, we see some distinct trends against those in Table 2. First, the graph modeling methods show better performances than the *BIOHD* labeling method (Tang et al. 2018). Besides, the transition method (Wang and Lu 2019) obtains higher F1 scores among all baselines. Notably, our model achieves the best results against all baselines in terms of all (nearly) metrics.

Effectiveness for Boundary Detection It is a long-reached understanding that boundary detection is the most difficult part for NER. Since the boundaries of irregular mentions are more complex, discontinuous NER entails more troubles. We now explore the abilities of our model and the baselines on detecting the boundaries in two levels as shown in Figure 4. The span-level boundary refers to

	CADEC		ShARe13	
	Not Ovlp.	Ovlp.	Not Ovlp.	Ovlp.
Tang et al. (2018)	0.0	15.7	8.9	24.0
Muis and Lu (2016)	0.0	23.8	33.4	30.6
Wang and Lu (2019)	8.3	27.0	31.5	34.8
Dai et al. (2020)	0.0	25.6	37.2	22.7
Ours(MAPtr)	**20.2**	**33.3**	**53.2**	**41.9**

Table 5: Results (F1) of recognizing overlapped (Ovlp.) and non-overlapped mentions among discontinuous ones.

the inner-boundaries of the continuous spans within a discontinuous mention, and the mention-level boundary means the boundaries of an entire discontinuous mention. Both of them contains left and right boundaries, and the latter also contains the boundaries of those spans in the middle of a discontinuous mention, denoted as "Avg.Middle".

In Figure 4, we observe that both of our models have lower error rates on different types of boundary evaluations. Particularly, MAPtr prominently reduces the error rates, which explains its higher performances. Besides, we find that for both types of boundary evaluations, the accuracy for right boundary detection are lower than that for left boundary detection. In the span-level boundary evaluation, all baselines are almost in the same level, while our Ptr model achieves slightly lower error rates. For the mention-level boundary evaluation, we find that the baselines (Wang and Lu 2019; Dai et al. 2020) achieve higher accuracy in the middle or right boundary detection.

Effectiveness for Overlapped NER In CADEC and ShARe13, there are 88.0% and 46.6% discontinuous mentions overlapped with each other. We now examine the influences of discontinuous and overlapping structures for models In Table 5, we find that, in the CADEC dataset, the recognition for overlapped mentions is more successful than that for non-overlapped ones. Most of the baselines cannot recognize any non-overlapped mention, but our model achieves a much higher F1. For the ShARe13 dataset, most of the models can recognize certain overlapped and non-overlapped mentions. However, our model still shows the best overall results.

Impact of Discontinuous Mention Intervals We study the impact of interval lengths for discontinuous mentions. As shown in Figure 5, the trends of line graphs for two datasets are distinct. For the CADEC dataset, the F1s for recognizing the mentions with the interval length 2 are universally higher. By contrast, for the ShARe13 dataset, the shorter the interval lengths are, the higher the performances become. In addition, our MAPtr model wins the best performance against all baselines under any setting.

Case Study for Pointer Visualization To understand the working mechanism of the pointer network, we select a case and visualize the pointer distribution at each decoding frame in Figure 6. The sentence contains four overlapped mentions: "*legs started going numb*", "*legs started tingling*", "*arms started going numb*", "*arms started tingling*", which are all recognized successfully. Figure 6 shows that all the correct positions have the largest pointer weights. The pointer can correctly detect all the boundaries of each text span within a discontinuous mention. For example, given the input "*legs*", the pointer directs to "*started*" as the next component, followed by "*going*" and "*numb*".

The visualization also partially reveals the effectiveness of the memory-augmented pointer mechanism. For instance, when detecting "*legs started tingling*" with the current decoding input "*started*", our model points to the correct word "*tingling*" rather than "*going*", by consulting the prior recognized mention "*legs started going numb*" in the memory. Furthermore, we can see that the word "*going*" receives a much lower pointer weight than that of "*tingling*".

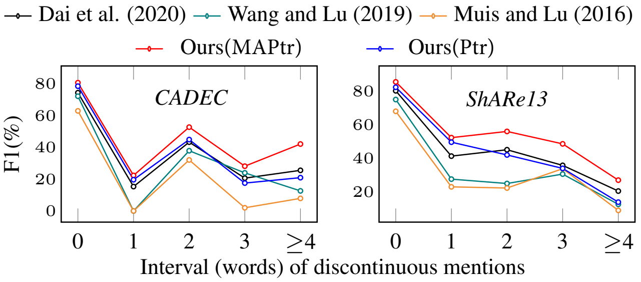


Figure 5: Impact of Discontinuous Mention Intervals. 0 indicates continuous mentions.

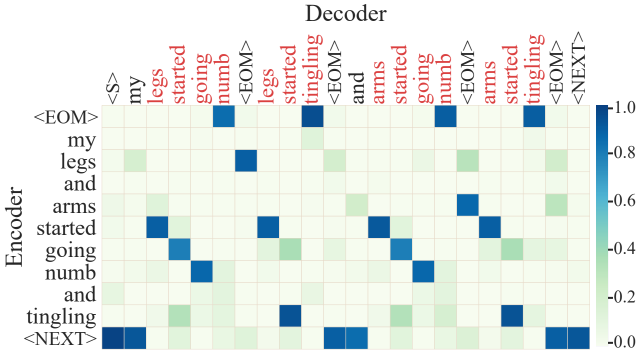


Figure 6: A case study for pointer visualization. Continuous words in red constitute a mention.

Conclusion

We present a pointer-network-based model for discontinuous entity mention recognition. Unlike prior methods, our framework makes each decision by consulting all the input elements and thus benefits from global information. Moreover, our framework is able to detect discontinuous, overlapped and flat entity mentions in an end-to-end fashion simultaneously. We further propose a memory-augmented pointer mechanism to enhance boundary detection of entity mentions. Experimental results on two benchmark datasets show that our model outperforms all previous state-of-the-art models. Further in-depth analyses show that our proposed mechanisms are effective for various NER problems.

Acknowledgments

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

Alex, B.; Haddow, B.; and Grover, C. 2007. Recognising nested named entities in biomedical text. In *Proceedings of the BioNLP*, 65–72.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of the ICLR*.

Bojanowski, P.; Grave, E.; Joulin, A.; and Mikolov, T. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5: 135–146.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. P. 2011. Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research* 12: 2493–2537.

Dai, X.; Karimi, S.; Hachey, B.; and Paris, C. 2020. An Effective Transition-based Model for Discontinuous NER. In *Proceedings of the ACL*, 5860–5870.

Devlin, J.; Chang, M.-W.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the NAACL*, 4171–4186.

Fei, H.; Ren, Y.; and Ji, D. 2020a. Boundaries and edges rethinking: An end-to-end neural model for overlapping entity relation extraction. *Information Processing & Management* 57(6): 102311.

Fei, H.; Ren, Y.; and Ji, D. 2020b. Dispatched attention with multi-task learning for nested mention recognition. *Information Science* 513: 241–251.

Fei, H.; Ren, Y.; Zhang, Y.; Ji, D.; and Liang, X. 2020. Enriching contextualized language model from knowledge graph for biomedical information extraction. *Briefings in Bioinformatics* .

Finkel, J. R.; and Manning, C. D. 2009. Nested named entity recognition. In *Proceedings of the EMNLP*, 141–150.

Florian, R.; Hassan, H.; Ittycheriah, A.; Jing, H.; Kambhatla, N.; Luo, X.; Nicolov, N.; and Roukos, S. 2004. A Statistical Model for Multilingual Entity Detection and Tracking. In *Proceedings of the NAACL*, 1–8.

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

Karimi, S.; Metke-Jimenez, A.; Kemp, M.; and Wang, C. 2015. Cadec: A corpus of adverse drug event annotations. *J. Biomed. Informatics* 55: 73–81.

Katiyar, A.; and Cardie, C. 2018. Nested named entity recognition revisited. In *Proceedings of the NAACL*, 861–871.

Kim, J.-D.; Ohta, T.; Tateisi, Y.; and Tsujii, J. c. 2003. GENIA corpus semantically annotated corpus for biotextmining. *Bioinformatics* 19: 180–182.

Kurita, S.; and Søgaard, A. 2019. Multi-Task Semantic Dependency Parsing with Policy Gradient for Learning Easy-First Strategies. In *Proceedings of the ACL*, 2420–2430.

Lafferty, J. D.; Mccallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the ICML*, 282–289.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural Architectures for Named Entity Recognition. In *Proceedings of the NAACL*, 260–270.

Lee, J.; Yoon, W.; Kim, S.; Kim, D.; Kim, S.; So, C. H.; and Kang, J. 2020. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics* 36(4): 1234–1240.

Li, F.; Zhang, M.; Tian, B.; Chen, B.; Fu, G.; and Ji, D. 2018. Recognizing irregular entities in biomedical text via deep neural networks. *Pattern Recognit. Lett.* 105: 105–113.

Li, J.; Ye, D.; and Shang, S. 2019. Adversarial Transfer for Named Entity Boundary Detection with Pointer Networks. In *Proceedings of the IJCAI*, 5053–5059.

Lu, W.; and Roth, D. 2015. Joint mention extraction and classification with mention hypergraphs. In *Proceedings of the EMNLP*, 857–867.

Luong, T.; Pham, H.; and Manning, C. D. 2015. Effective Approaches to Attention-based Neural Machine Translation. In *Proceedings of the EMNLP*, 1412–1421.

Ma, X.; Hu, Z.; Liu, J.; Peng, N.; Neubig, G.; and Hovy, E. 2018. Stack-Pointer Networks for Dependency Parsing. In *Proceedings of the ACL*, 1403–1414.

McDonald, R. T.; and Nivre, J. 2011. Analyzing and Integrating Dependency Parsers. *Computational Linguistics* 37(1): 197–230.

Metke-Jimenez, A.; and Karimi, S. 2016. Concept Identification and Normalisation for Adverse Drug Event Discovery in Medical Forums. In *Proceedings of the BMDID-ISWC*.

Muis, A. O.; and Lu, W. 2016. Learning to Recognize Discontiguous Entities. In *Proceedings of the EMNLP*, 75–84.

Peters, M. E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; and Zettlemoyer, L. 2018. Deep Contextualized Word Representations. In *Proceedings of the NAACL*, 2227–2237.

Pradhan, S.; Elhadad, N.; South, B. R.; Martínez, D.; Christensen, L. M.; Vogel, A.; Suominen, H.; Chapman, W. W.; and Savova, G. K. 2013. Task 1: ShARe/CLEF eHealth Evaluation Lab 2013. In *Proceedings of the CLEF*.

See, A.; Liu, P. J.; and Manning, C. D. 2017. Get To The Point: Summarization with Pointer-Generator Networks. In *Proceedings of the ACL*, 1073–1083.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the NeurIPS*, 3104–3112.

Sutton, C. A.; McCallum, A.; and Rohanimanesh, K. 2007. Dynamic Conditional Random Fields: Factorized Probabilistic Models for Labeling and Segmenting Sequence Data. *Journal of Machine Learning Research* 8: 693–723.

Tang, B.; Cao, H.; Wu, Y.; Jiang, M.; and Xu, H. 2013. Recognizing clinical entities in hospital discharge summaries using Structural Support Vector Machines with word representation features. *BMC Med. Inf. & Decision Making* 13(S-1): S1.

Tang, B.; Hu, J.; Wang, X.; and Chen, Q. 2018. Recognizing Continuous and Discontinuous Adverse Drug Reaction Mentions from Social Media Using LSTM-CRF. *Wireless Communications and Mobile Computing* 2018.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Proceedings of the NeurIPS*, 5998–6008.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer Networks. In *Proceedings of the NeurIPS*, 2692–2700.

Wang, B.; and Lu, W. 2019. Combining Spans into Entities: A Neural Two-Stage Approach for Recognizing Discontiguous Entities. In *Proceedings of the EMNLP*, 6216–6224.

Wang, S.; Che, W.; Zhang, Y.; Zhang, M.; and Liu, T. 2017. Transition-Based Disfluency Detection using LSTMs. In *Proceedings of the EMNLP*, 2785–2794.

Williams, R. J.; and Zipser, D. 1989. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Computation* 1(2): 270–280.

Xu, J.; Zhang, Y.; Wang, J.; Wu, Y.; Jiang, M.; Soysal, E.; and Xu, H. 2015. UTH-CCB: The Participation of the SemEval 2015 Challenge-Task 14. In *Proceedings of the SemEval*, 311–314.

Yu, J.; Bohnet, B.; and Poesio, M. 2020. Named Entity Recognition as Dependency Parsing. In *Proceedings of the ACL*, 6470–6476.

Yu, N.; Zhang, M.; and Fu, G. 2018. Transition-based Neural RST Parsing with Implicit Syntax Features. In *Proceedings of the COLING*, 559–570.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; and Zhao, J. 2014. Relation classification via convolutional deep neural network. In *Proceedings of the COLING*, 2335–2344.