Joint Extraction of Entities and Overlapping Relations Using Position-Attentive Sequence Labeling

Dai Dai,[1] **Xinyan Xiao,**[1] **Yajuan Lyu,**[1] **Shan Dou,**[2] **Qiaoqiao She,**[1] **Haifeng Wang**[1]
[1]Baidu Inc., Beijing, China
[2]Department of Automation, Tsinghua University, Beijing, China
{daidai,xiaoxinyan,lvyajuan,sheqiaoqiao,wanghaifeng}@baidu.com
ds16@mails.tsinghua.edu.cn

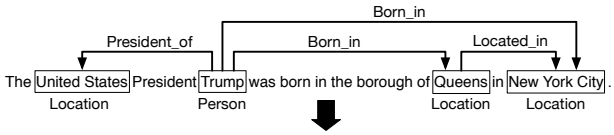
Abstract

Joint entity and relation extraction is to detect entity and relation using a single model. In this paper, we present a novel unified joint extraction model which directly tags entity and relation labels according to a query word position p, i.e., detecting entity at p, and identifying entities at other positions that have relationship with the former. To this end, we first design a tagging scheme to generate n tag sequences for an n-word sentence. Then a position-attention mechanism is introduced to produce different sentence representations for every query position to model these n tag sequences. In this way, our method can simultaneously extract all entities and their type, as well as all overlapping relations. Experiment results show that our framework performances significantly better on extracting overlapping relations as well as detecting long-range relation, and thus we achieve state-of-the-art performance on two public datasets.

Introduction

Relation extraction (RE) aims to detect the semantic relation between entities in unstructured text. Traditional RE systems separate this task into pipelined subtasks: first detecting entities and then classifying the relation types between candidate entity pairs. Such a framework makes the task easy to conduct, but it ignores the underlying interdependencies and error propagation between these two subtasks (Li and Ji 2014; Gupta, Schütze, and Andrassy 2016).

Different from the pipelined methods, joint extraction is to detect entities together with their relations using a joint model. Recent studies show that joint learning approaches can effectively integrate the information of entity and relation, and therefore achieve better performance in both subtasks. Most previous joint models are based on feature-based structured learning (Kate and Mooney 2010; Li and Ji 2014; Miwa and Sasaki 2014; Ren et al. 2017). These methods heavily depend on hand-crafted features and other NLP toolkits. Recently, several neural architectures have been applied, most of which utilize parameter sharing for joint modeling, but still require explicit separate components for entity recognition and relation classification (Miwa and Bansal 2016; Gupta, Schütze, and Andrassy 2016). In contrast, Zheng et al. (2017b) proposes a special tagging scheme



Extracted Relation Triplets

Entity 1	Relation Type	Entity2
Trump	President_of	United States
Trump	Born_in	Queens
Trump	Born_in	New York City
Queens	Located_in	New York City

Figure 1: An example sentence that contains overlapping relations which share the same entity in the sentence. For example, the first three relations in the table are overlapping because they share the same entity "Trump". Similarly, the last two relations are also overlapping due to the shared entity "New York City". Such overlapping relations are very common in datasets of relation extraction (see Table 1).

to convert joint extraction to a sequence labeling problem, solving the task in a unified way. However, their method cannot identify overlapping relations, which may lead to poor recall when processing a sentence with overlapping relations (see Figure 1). Another unified method proposed by Zeng et al. (2018) employs sequence-to-sequence learning with copy mechanism. Although it can extract overlapping relations, their model fails to identify multi-word entities. Overall, it is still challenging to jointly extract entities and overlapping relations using a single unified model.

In this paper, we present a new unified method to solve the joint extraction by tagging entity and relation labels simultaneously according to a query word position p. Given a sentence and a query position p, our model is to answer two pseudo questions: "What is the entity and its type at p?" and "Which entities have relationship with the the one at p?" To this end, we design a special tagging scheme that annotates entity labels at query position p, as well as relation labels at other positions in the sentence (see Figure 2). Thus, it actually transforms the joint relation extraction problem to a list of sequence labeling problems, e.g., for an n-word sentence, we annotate n different tag sequences according to n query positions. To model these n tag sequences in a single unified model, a novel position-attention mechanism is introduced

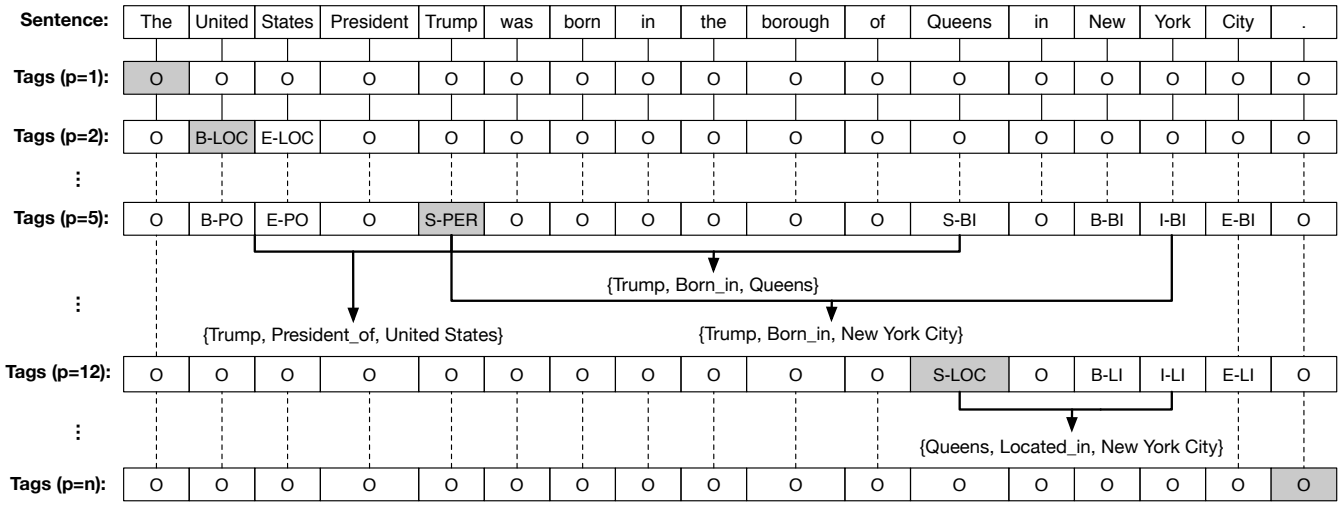


Figure 2: An example of our tagging scheme, where n denotes the sentence length, $p \in [1, n]$ is the query word position. For a query p, we build an n-tag sequence to represent all possible overlapping relations that correspond to the entity at p. Thus, entity type is labeled at p if it is the start of an entity, and relation types are labeled at the rest of words if they have relationship with the entity at p. In this way, all entities and overlapping relations can be annotated using this tagging scheme. In this example, "LOC" in "B-LOC" is short for entity type *LOCATION*, "PER" in "S-PER" is short for *PERSON*, "PO" in "B-PO" is short for relation type *President_of*, "BI" in "B-BI" is short for *Born_in* and "LI" in "B-LI' is short for *Located_in*.

into the sequence tagging model (see Figure 3) to produce n different position-aware sentence representations. These representations are then used to decode different tagging results, from which we can extract all entities, their types and all overlapping relations. In addition, the proposed attention mechanism can build direct connection between words (entities), which might contribute to extraction of long-range relation (the two entities have a long distance from each other).

The key contribution of this paper is the newly proposed unified model for joint extraction of entities and overlapping relations. In detail:

1. We design a tagging scheme which can simultaneously represent type of entities and overlapping relations.
2. We propose a position-attention mechanism to produce different position-aware sentence representations according to query position p, which can be used to decode different tag sequences and extract overlapping relations.
3. We demonstrate the effectiveness of our method using two public datasets and achieve state-of-the-art results. Furthermore, the analysis shows that our model performance better on extracting long-range relations, which are usually more difficult.

Methodology

In this section, we first introduce our tagging scheme which transforms overlapping relation extraction to a list of sequence labeling problems. Then we detail the position-attentive sequence labeling model based on this tagging scheme.

Tagging Scheme

As shown in Figure 2, for an n-word sentence, n different tag sequences are annotated based on our tagging scheme according to different query position p. In each tag sequence, entity type is labeled at the current query position p if this position is the start of an entity, and other entities, which have relationship to the entity at p, are labeled with relation types. The rest of tokens are assigned label "O" (Outside), indicating that they do not correspond to the entity that is attended to. Thus, relations, which are represented by a triplet $(Entity_1, RelationType, Entity_2)$, can be extracted based on a tag sequence. Here, $Entity_1$ is the first argument of the relation and can be obtained from the detected entity at the query position. $Entity_2$, the second argument, and $RelationType$ can be extracted from other detected entities and their labels in the same tag sequence. Obviously, the first entity can be used multiple times to form overlapping relations.

For example, when the query position p is 5 and the token at this position is "Trump", the label of "Trump" is *PERSON*. Other entities, such as "United States", "Queens" and "New York City", which are corresponding to "Trump", are labeled as *President_of*, *Born_in* and *Born_in*. The first entity "Trump" will be used for three times to form three different triplets in this situation. If the query p is 12 and the token is "Queens", its tag is *LOCATION*, and the corresponding entity "New York City" is labeled as *Located_in*. For p is 2, there is no corresponding entity and thus only the entity type of "United States" is labeled (notice that the relation is unidirectional). All of the tokens are labeled as "O" when p is 1 because there is no entity at the position p which is attended to. For both entity and relation type annotation, we

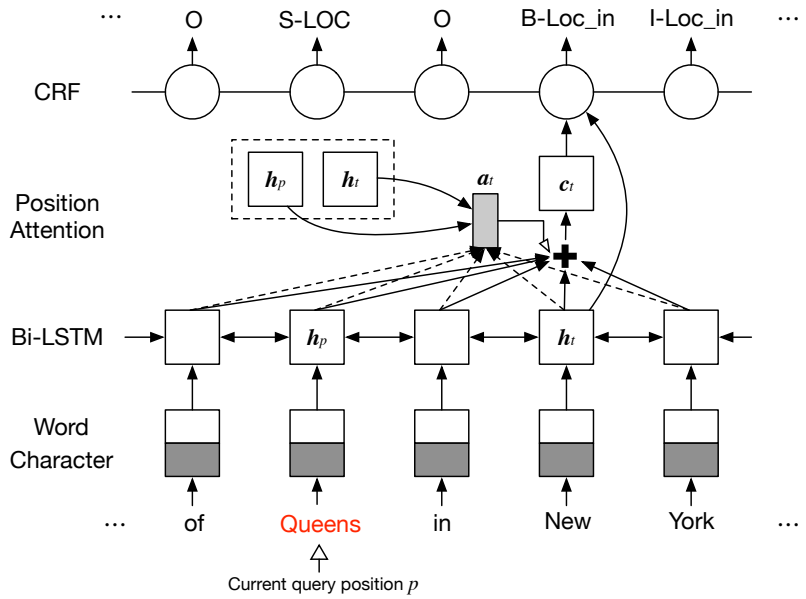


Figure 3: The architecture of our position-attentive sequence labeling model. It receives the same sentence input and a different query position p to extract all overlapping relations. Here, the red "Queens" is the token at query position p, $\mathbf{h}_p$ is the hidden state of time-step p, $\mathbf{h}_t$ is the hidden vector of time-step t, $\mathbf{a}_t$ is the attention weights and $\mathbf{c}_t$ is the attention-pooling vector.

use "BIES" (Begin, Inside, End, Single) signs to indicate the position information of tokens in the entity, and therefore we can extract multi-word entities. Through our tagging scheme, all of overlapping relations in an n-word sentence, together with all entity mentions and their entity types, can be represented in n tag sequences.

Note that our tagging scheme is quite different from table filling method (Miwa and Sasaki 2014). It uses only half of the table and hence cannot represent reverse relation, which is also a kind of overlapping form, e.g., if the relation of entity pair $(Entity_1, Entity_2)$ is *Capital_of*, the reverse pair $(Entity_2, Entity_1)$ may have relation of *Contains*. In addition, the best search order "close-first" actually equals to first detecting entities and then classifying relations. Most joint neural methods following table filling (Gupta, Schütze, and Andrassy 2016; Zhang, Zhang, and Fu 2017) also use this search order and usually require explicit RC components in the network.

End-to-End Sequence Labeling Model with Position-Attention

With our tagging scheme, we build an end-to-end sequence labeling neural architecture (Figure 3) to jointly extract entities and overlapping relations. Our architecture first encodes the n-word sentence using a RNN encoder. Then, we use a position-attention mechanism to produce different position-aware sentence representations for every query position p. Based on these position-aware representations, we finally use Conditional Random Field (CRF) to decode the n tag sequences to extract entities and overlapping relations.

Bi-LSTM Encoder RNNs have been shown powerful to capture the dependencies of input word sequence. In this work, we choose bidirectional Long Short Term Memory (Bi-LSTM) as the encoding RNN. Consider a sentence that consists of n words $S = \{w_t\}_{t=1}^{n}$, we first convert the words to their word-level representations $\{\mathbf{w}_t^w\}_{t=1}^{n}$, where $\mathbf{w}_t \in \mathbb{R}^d$ is the d-dimensional word vector corresponding to the t-th word in the sentence. As out of vocabulary (OOV) word is common for entity, we also augment word representation with character-level information. Character-level representations $\{\mathbf{w}_t^c\}_{t=1}^{n}$ are extracted by a convolution neural network (CNN), where $\mathbf{w}_t^c \in \mathbb{R}^k$ is the k-dimensional outputs of the CNN with k-filters. This CNN, similar to the one applied on words, receives character embeddings as input and generates representation which effectively captures the morphological information of the word. The final representations of words are concatenation of word-level and character-level representation $[\mathbf{w}_t^w; \mathbf{w}_t^c]$. Then, the Bi-LSTM is implemented to produce forward state $\overrightarrow{\mathbf{h}_t}$ and backward state $\overleftarrow{\mathbf{h}_t}$ for each time step:

$$\overrightarrow{\mathbf{h}_t} = \overrightarrow{LSTM}(\overrightarrow{\mathbf{h}}_{t-1}, [\mathbf{w}_t^w, \mathbf{w}_t^c]) \quad (1)$$

$$\overleftarrow{\mathbf{h}_t} = \overleftarrow{LSTM}(\overleftarrow{\mathbf{h}}_{t+1}, [\mathbf{w}_t^w, \mathbf{w}_t^c]) \quad (2)$$

These two separate hidden states capture both past (forward) and future (backward) information of the word sequence. Finally, we concatenate $\overrightarrow{\mathbf{h}_t}$ and $\overleftarrow{\mathbf{h}_t}$ as the encoding output of the t-th word, donated as $\mathbf{h}_t = [\overrightarrow{\mathbf{h}_t}; \overleftarrow{\mathbf{h}_t}]$, to obtain the final sentence representations $\mathbf{H} = \{\mathbf{h}_t\}_{t=1}^{n}$. However, such representations are not enough for decoding the n tag sequences produced by our tagging scheme. Because position information is lacking where to detect $Entity_1$ and other components in overlapping triplets.

Position-Attention Mechanism The key information for detecting an entity and its relationship with another entity include: (1) the words inside the entity itself; (2) the depended entity; (3) the context that indicates the relationship. Under these considerations, we propose position-attention, which can encode the entity information at query position as well as the context information of the whole sentence to generate position-aware and context-aware representations $\{\mathbf{u}_t\}_{t=1}^n$:

$$\mathbf{u}_t = [\mathbf{h}_t; \mathbf{c}_t] \tag{3}$$

where $\mathbf{c}_t = att(\mathbf{H}, \mathbf{h}_p, \mathbf{h}_t)$ is an attention-pooling vector of the whole sentence ($\mathbf{H}$):

$$\begin{aligned} s_{tj} &= \mathbf{v}^T \tanh(\mathbf{W}^H \mathbf{h}_j + \mathbf{W}^p \mathbf{h}_p + \mathbf{W}^h \mathbf{h}_t) \\ a_{tj} &= \exp(s_{tj}) \Big/ \sum\nolimits_{k=1}^{n} \exp(s_{tk}) \\ \mathbf{c}_t &= \sum\nolimits_{j=1}^{n} a_{tj}\, \mathbf{h}_j \end{aligned} \tag{4}$$

where $\mathbf{W}^H, \mathbf{W}^p, \mathbf{W}^h, \mathbf{v}$ are parameters to be learned, $\mathbf{h}_j, \mathbf{h}_p, \mathbf{h}_t$ are the hidden states at position j, p and t respectively, s_{tj} is the score computed by comparing $\mathbf{h}_p$ and $\mathbf{h}_t$ with each of the sentence state $\mathbf{h}_j$, and a_{tj} is the attention weight produced by normalization of s_{tj}. It means that $\mathbf{h}_p$, the state at the position that we attend to, is used for comparing with the sentence representations to encode position information, and $\mathbf{h}_t$ is used for matching the sentence representations against itself (self-matching) to collect information from the context (Wang et al. 2017). The position-attention mechanism produces different sentence representations according to the query position p, and thus solves the problem for modeling different tag sequences of a sentence. The following tag decoder can predict completely distinct labels given the same sentence and different query positions.

CRF Decoder It is shown beneficial for sequence labeling model to consider the correlations between labels in neighborhoods and jointly decode the best chain of labels. Thus, we use CRF for jointly decoding, instead of independently decoding each label. We consider $\mathbf{Z} = \{\mathbf{z}_t\}_{t=1}^n$ to be the input sequence scores, which is generated from position-aware sentence representation $\mathbf{u}_t$:

$$\mathbf{z}_t = \mathbf{W}^u \mathbf{u}_t \tag{5}$$

where $\mathbf{z}_t \in \mathbb{R}^{N_t}$ is the tag scores of the t-th word, and N_t is the number of distinct tags. Consider $Z_{t,j}$ as the score of the j-th tag at position t. For a sequence of labels $\mathbf{y} = \{y_t\}_{t=1}^n$, we define the decoding score as:

$$score(\mathbf{Z}, \mathbf{y}) = \sum_{t=0}^{n} A_{y_t, y_{t+1}} + \sum_{t=1}^{n} Z_{t, y_t} \tag{6}$$

where $\mathbf{A}$ is transition matrix such that $A_{i,j}$ represents the transition score from the tag i to tag j. Then we get the conditional probability over all possible label sequences $\mathbf{y}$ with the following form:

$$p(\mathbf{y}|\mathbf{Z}) = \frac{\exp(score(\mathbf{Z}, \mathbf{y}))}{\sum_{\mathbf{y}' \in Y_\mathbf{Z}} \exp(score(\mathbf{Z}, \mathbf{y}'))} \tag{7}$$

Datasets	NYT	Wiki-KBP
#Relation types	24	14
#Training sentences	66,335	75,325
#Training sentences with ORs	24,183	26,628
#Training triplets	116,747	113,540
#Test sentences	395	289
#Test sentences with ORs	1	23
#Test triplets	3878	2002
#Test triplets (without "None")	396	316

Table 1: Statistics of the datasets in our experiments. Here, "ORs" is short for overlapping relations.

where $Y_\mathbf{Z}$ represents the set of possible label sequences for $\mathbf{Z}$. During training, we maximize the log-likelihood of the correct tag sequence on the the training set $\{(\mathbf{Z}_i, \mathbf{y}_i)\}$:

$$L = \sum_i \log\ p(\mathbf{y}|\mathbf{Z}) \tag{8}$$

Decoding is to search for the tag sequence that obtains the maximum score given by:

$$\mathbf{y}^* = \underset{\mathbf{y} \in Y_\mathbf{Z}}{\operatorname{argmax}}\ score(\mathbf{Z}, \mathbf{y}) \tag{9}$$

The best tag sequence $\mathbf{y}^*$ can be computed using the Viterbi algorithm.

Extracting Entities and Overlapping Relations from Tag Sequences

From our tagging scheme, we know that the first entity of the triplet and its entity type can be extracted from the labels at the query position, and the second corresponding entity, if existed, as well as the relation type, can be obtained from the labels at other positions (see Figure 2). The overlapping relation extracting problem is solved because the first entity is allowed to belong to multiple triplets in a tag sequence. Through n tag sequence results considering different query positions, we can extract all overlapping relations in a sentence, as well as all entity mentions and their entity types. Moreover, the extracted entity types can be used to validate the triplets, for example, if the relation type is *Born_in*, the entity type of the first argument must be *PERSON*.

Experiments

Experiment Settings

Datasets We use two public datasets to demonstrate the effectiveness of our method: (1) **NYT** (Riedel, Yao, and McCallum 2010) is a news corpus sampled from 294k 1989-2007 New York Times news articles. We use the same dataset[1] published by (Ren et al. 2017). The training data are automatically labeled using distant supervision, while 395 sentences are annotated by the author of (Hoffmann et al. 2011) and used as test data. (2) **Wiki-KBP** (Xiao and Weld 2012) utilizes 1.5M sentences sampled from 780k Wikipedia articles as training corpus, while test set consists of 289 sentences selected by the author of (Ren et al. 2017) from the

[1] https://github.com/shanzhenren/CoType

Method	NYT			Wiki-KBP		
	Prec.	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
MultiR (Hoffmann et al. 2011)	0.338	0.327	0.333	0.301	0.530	0.380
DS-Joint (Li and Ji 2014)	0.574	0.256	0.354	-	-	-
Cotype (Ren et al. 2017)	0.423	0.511	0.463	0.311	**0.537**	0.388
ReHession (Liu et al. 2017)	0.412	0.573	0.480	0.367	0.493	0.421
LSTM-CRF (Zheng et al. 2017b)	**0.693**	0.310	0.428	**0.596**	0.256	0.358
LSTM-LSTM-Bias (Zheng et al. 2017b)	0.615	0.414	0.495	0.536	0.303	0.387
PA-LSTM-CRF	0.494	**0.591**	**0.538**	0.511	0.393	**0.444**

Table 2: Comparison of our model and baseline methods on NYT and Wiki-KBP datasets. PA-LSTM-CRF denotes our sequence labeling model with position-attention.

manual annotations in 2013 KBP slot filling assessment results (Ellis et al. 2012). We use the public training data[2] which are automatically labeled using distant supervision and handcrafted patterns by the author of (Liu et al. 2017). Statistics of the datasets are shown in Table 1.

Evaluation We mainly focus on overlapping relation extraction in this paper. Because our model directly extracts relations without detecting entities and their entity types first, we only evaluate the results of extracted triplets. We use the F1 metric computed from Precision (Prec.) and Recall (Rec.) for evaluation. A triplet is marked correct when its relation type and two corresponding entities are all correct, where the entity is considered correct if the head and tail offsets are **both** correct. We exclude all triplets with relation type of "None" (because we do not require them as negative samples) and create a validation set by randomly sampling 10% sentences from test set as previous studies (Ren et al. 2017; Zheng et al. 2017b) did.

Implementation Details For both datasets, the word embeddings are randomly initialized with 100 dimensions and the character embeddings are randomly initialized with 50 dimensions. The window size of CNN is set to 3 and the number of filters is 50. For Bi-LSTM encoder, the hidden vector length is set to 200. We use $l2$ regularization with a parameter of 0.001 to avoid overfitting. Parameter optimization is performed using Adam (Kingma and Ba 2014) with learning rate 0.001 and batch size 16. In addition, we randomly sample 10% negative tag sequences in which all words are labeled as "O" to reduce the training samples.

Baselines We compare our method on NYT and Wiki-KBP datasets with the following baselines: (1) **MultiR** (Hoffmann et al. 2011) models training label noise based on multi-instance multi-label learning; (2) **DS-Joint** (Li and Ji 2014) jointly extracts entities and relations using structured perceptron; (3) **Cotype** (Ren et al. 2017) learns jointly the representations of entity mentions, relation mentions and type labels; (4) **ReHession** (Liu et al. 2017) employs heterogeneous supervision from both knowledge base and heuristic patterns. (5) **LSTM-CRF** and **LSTM-LSTM-Bias** (Zheng et al. 2017b), the most related work to our method, converts the joint extraction task to a sequence labeling problem based on a novel tagging scheme. However, it cannot detect overlapping relations. Note that we do not compare our method with (Zeng et al. 2018) for two reasons. First, their model can decode only the first word of multi-word entity, while ours can detect the whole. In this paper, we conduct a stricter evaluation in which an entity is considered correct only if the head and tail offsets are both correct, which makes the task more challenging. Second, they do not report the result on manually labeled NYT test set. Instead, they use test set split from training data which is generated by distant supervision.

[2]https://github.com/LiyuanLucasLiu/ReHession

Experimental Results and Analyses

Main Results We report our experimental results on NYT and Wiki-KBP datasets in Table 2. It is shown that our model, position-attentive LSTM-CRF (PA-LSTM-CRF), outperforms all the baselines and achieves state-of-the-art F1 score on both datasets. Specially, compared to LSTM-LSTM-Bias (Zheng et al. 2017b), our method achieves significant improvements of 5.7% in F1 Wiki-KBP dataset, which is mainly because our model can extract overlapping relations. For NYT dataset, although no overlapping relation is manually labeled, our model also outperforms LSTM-LSTM-Bias by 4.3% in F1 due to a large improvement in Recall. We consider that it is because our model our model is capable of identifying more long-range relations, which will be further discussed in section of attention analysis. We also notice that the Precision of our model drops compared with LSTM-LSTM-Bias. This is mainly because many overlapping relations are not annotated in the manually labeled test data. We will discuss it in the following paragraph.

Effect on Overlapping Relation Extraction As Table 1 shows, there are about a third of sentences that contain overlapping relations in training data of both datasets, but much less in test sets. In fact, we find that many overlapping relations are omitted from the manually labeled test data in both datasets, especially for the relations of reverse pair of entities, e.g., "per:parents" and "per:children" in Wiki-KBP, or "/location/country/administrative_divisions" and "/location/administrative_division/country" in NYT. This may significantly affect the performance of our model on overlapping relation detection, especially the Precision. Thus, to discover the capability of our model to identify overlapping triplets, we simply add some gold triplets into test set of Wiki-KBP. For example, we add a ground-truth reverse triplet with "per:parents" type if "per:children" is originally labeled and vice versa. This will increase the number of sen-

Method	Wiki-KBP			
	Prec.	Rec.	F1	F1*
LSTM-LSTM-Bias	0.581	0.290	0.386	0.310
PA-LSTM-CRF	**0.623**	**0.436**	**0.513**	**0.637**

Table 3: Evaluation for overlapping relation extraction on Wiki-KBP, where F1* denotes the F1 score on sentences with overlapping relations. By simply adding some gold triplets using the reverse relations, our model achieves a large improvement of 6.9% in F1 and 11.2% in Precision compared with the results in Table 2.

tences with overlapping relations in test set to about 50 from 23, but still much less in proportion to training data. We report the evaluation results compared with LSTM-LSTM-Bias in Table 3. It can be seen that our model achieves an large improvement of 6.9% in F1 and 11.2% in Precision compared with the results in Table 2, while the performance of LSTM-LSTM-Bias basically remain the same in F1. Moreover, for overlapping relations, our model significantly outperforms LSTM-LSTM-Bias by about 30%, which demonstrates the effectiveness of our method on extraction of overlapping relations.

Ablation Study We also conduct ablation experiments to study the effect of components of our model. As shown in Table 4, all the components play an important roles in our model. Consistent with previous work, the character-level representations are helpful to capture the morphological information and deal with OOV words in sequence labeling task (Ma and Hovy 2016). Introducing position attention mechanism for generating the position-aware representations seems an efficient way to incorporate the information of the query position compared with directly concatenating the the hidden vector at the query position to each state of the BiLSTM encoder. In addition, the self-matching in our position-attention mechanism also contributes to the final results for the reason of extracting more information from the context.

Comparison of Running Time While LSTM-LSTM-Bias or LSTM-CRF runs sequence tagging only once to extract non-overlapping relations, our model tags the same sentence for another $n-1$ times in order to recognize all overlapping relations. This means our model is more time-consuming ($O(n^2)$ vs. $O(n)$). For instance, LSTM-CRF only predicts about 300 samples and consumes 2s on Wiki-KBP test set, while our model decodes about 7000 tag sequences and takes about 50s. However, testing can be speeded up by sharing the sentence representation before position attention because it is identical for the other $n-1$ times decoding. In this way, the running time of our model reduces to 16s. Moreover, we may also prune some query positions to further accelerate in real application because it is impossible for some words to be the head of an entity.

Further Analysis for Attention

Detecting Long-Range Relations It is shown in previous work that attention mechanism is helpful to capturing the long range dependencies between arbitrary tokens, which

Wiki-KBP	Prec.	Rec.	F1
PA-LSTM-CRF	**0.511**	**0.393**	**0.444**
- Character embeddings	0.470	0.359	0.407
- Position attention	0.461	0.372	0.412
- Self-matching	0.459	0.392	0.423

Table 4: F1 results of ablation experiments on Wiki-KBP. For model without position attention, we directly concatenate the hidden vector at the query position to each hidden state of the sentence to generate position-aware representations. For model without self-matching, the third item $\mathbf{W}^h\mathbf{h}_t$ is removed from Equation 4.

is very important for detecting triplets composed of entities that have a long distance from each other. To further prove the effectiveness of position attention, we analyze the F1 score on triplets with different distances between entities on Wiki-KBP dataset. As shown in Figure 4, we find that the performance of our model remains stable as the distance between entities increases, while that of LSTM-LSTM-Bias drops significantly. It means that our model can effectively model the dependencies between entities despite a long distance between them.

Case Study for Attention Weights We select two sentences from the test set of Wiki-KBP to show the alignment of our position-attention for case study. As shown in Figure 5, the information of the first entity at the query position, together with the context evidence for recognizing relations, is encoded into the position-aware representations regardless of distance between entities. For instance, in Figure 5(a), the second entity "Albert" pays more attention to the possible corresponding entities "Anderson" and "Carol", as well as the key context word "sons" that contains information of relation. The model also produces reasonable alignment at other query position as Figure 5(b) shows. In Figure 5(c), "Thousand Oaks" can still attend to the first entity "Sparky Anderson" despite a long distance between them.

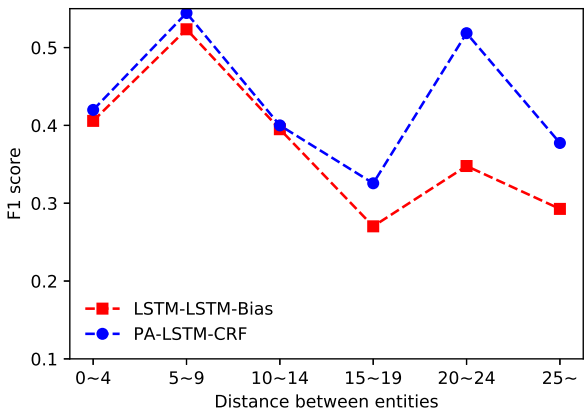


Figure 4: Model performance on triplets with different distances between entities on Wiki-KBP dataset.

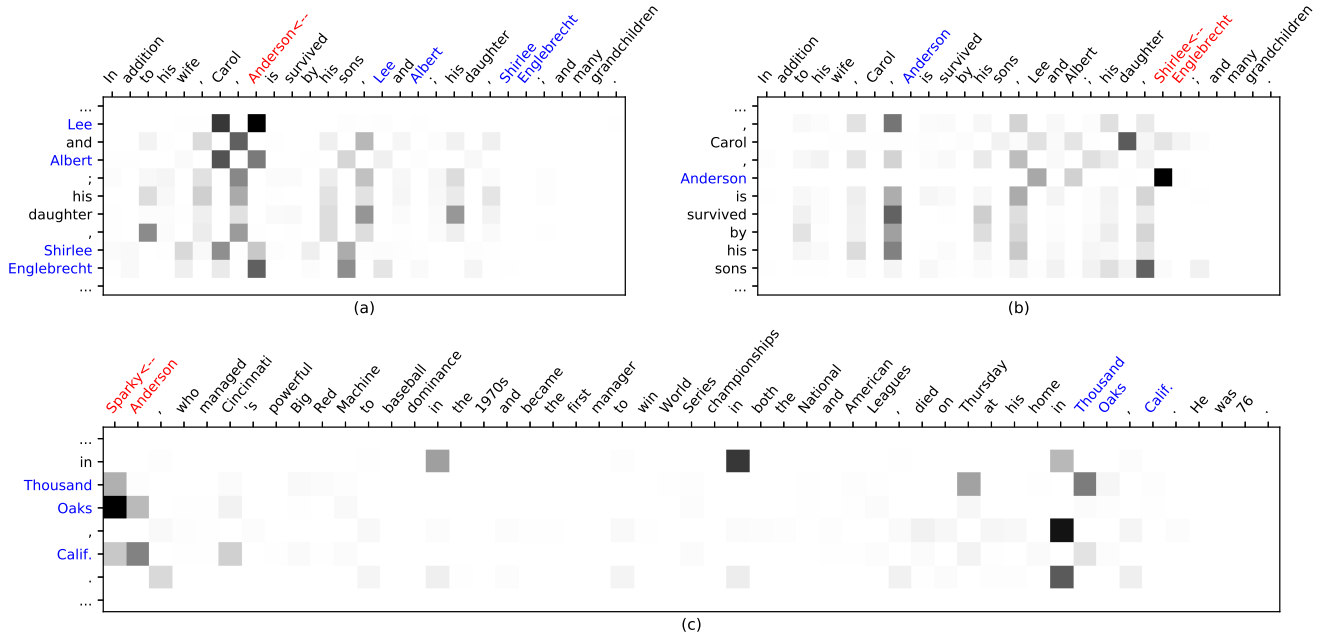


Figure 5: Part of attention matrices for position-attention. Each row is the attention weights of the whole sentence for the current token. The query position is marked by an arrow, the red tokens indicate the first entity extracted at the query position, and the blue tokens indicates the second corresponding entities. (a) and (b) are the attention matrices of different query positions for the same sentence, and (c) is the attention matrix for a sentence with long-range entity pairs.

Related Work

Traditional relation extraction frameworks divide this task into two separate steps: first performing named entity recognition (NER) and then conducting relation classification (RC). RC is normally treated as a problem of multi-label classification. Feature-based methods are first introduced to tackle this problem (Rink and Harabagiu 2010), and recently several neural architectures are applied, including CNNs (Zeng et al. 2014; 2015; Santos, Xiang, and Zhou 2015; Lin et al. 2016; Adel and Schütze 2017) and RNNs (Zhang and Wang 2015; Zhou et al. 2016; Zhang et al. 2017). All of these approaches utilize position embedding (or position indicator) to integrate entity position information and finally generate a sentence representation for identifying relation type of the input entity pair. Zhou et al. (2016) employs attention mechanism to decide the contribution of each word to the final representation. Zhang et al. (2017) proposes a position-aware attention mechanism to refine the information of position embedding. Besides, CRF is introduced to model the interdependency between entity type and relation type (Adel and Schütze 2017). However, all these methods require preprocessing step such as NER, and therefore may suffer from error propagation.

In contrast, joint extraction approaches detect entities and identify their relations using a joint model. Most of the joint models are based on feature-based structured learning (Kate and Mooney 2010; Singh et al. 2013; Li and Ji 2014; Miwa and Sasaki 2014; Ren et al. 2017; Wang et al. 2018). Recently, several end-to-end neural architectures are applied to joint extraction. Most of them have explicit separate components for NER and RC subtask (Miwa and Bansal 2016; Gupta, Schütze, and Andrassy 2016; Katiyar and Cardie 2017; Zhang, Zhang, and Fu 2017; Zheng et al. 2017a; Verga, Strubell, and McCallum 2018), and usually RC component is still behind NER component and depends on NER results. On the contrary, Zheng et al. (2017b) proposes a unified model which utilizes a special tagging scheme to convert joint extraction task to a sequence tagging problem. However, their model cannot recognize overlapping relations in the sentence. Zeng et al. (2018) proposes another unified model using sequence-to-sequence learning with copy mechanism to solve overlapping relation extraction, but their method fails to identify multi-word entities.

In this paper, we present a novel unified model from joint extraction of entities and overlapping relations. We design a tagging scheme which simultaneously annotates entity type and relation type in n tag sequences for an n-word sentence, and propose a position-attention mechanism to model them in a unified sequence tagging model. Attention mechanisms are used to model dependencies of input and output sequences, and are successfully applied in various NLP tasks (Bahdanau, Cho, and Bengio 2014; Vaswani et al. 2017; Cheng, Dong, and Lapata 2016; Wang et al. 2017). We also incorporate self-attention into our position-attention mechanism to capture the context information. It is a special case which is computed inside a sequence to capture the long-range dependencies between tokens, and is also widely used in many NLP tasks (Lin et al. 2017; Tan et al. 2017).

Conclusion

In this paper, we propose a unified position-attentive sequence labeling framework for joint extraction of entities and overlapping relations. Experiments show that our method can effectively extract overlapping relations and achieves the start-of-the-art results on two public datasets. Besides, we find that attention mechanism is helpful to modeling the long range dependencies and improves the model performance on long-range relation detection.

References

Adel, H., and Schütze, H. 2017. Global normalization of convolutional neural networks for joint entity and relation classification. In *Proceedings of the 2017 Conference on EMNLP*. Copenhagen, Denmark: ACL.

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Cheng, J.; Dong, L.; and Lapata, M. 2016. Long short-term memory-networks for machine reading. *arXiv preprint arXiv:1601.06733*.

Ellis, J.; Li, X.; Griffitt, K.; Strassel, S.; and Wright, J. 2012. Linguistic resources for 2013 knowledge base population evaluations. In *TAC*.

Gupta, P.; Schütze, H.; and Andrassy, B. 2016. Table filling multi-task recurrent neural network for joint entity and relation extraction. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, 2537–2547.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of ACL: Human Language Technologies-Volume 1*, 541–550. ACL.

Kate, R. J., and Mooney, R. J. 2010. Joint entity and relation extraction using card-pyramid parsing. In *Proceedings of the 14th Conference on Computational Natural Language Learning*, 203–212. ACL.

Katiyar, A., and Cardie, C. 2017. Going out on a limb: Joint extraction of entity mentions and relations without dependency trees. In *Proceedings of the 55th Annual Meeting of ACL*, volume 1, 917–928.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Li, Q., and Ji, H. 2014. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of ACL*, volume 1, 402–412.

Lin, Y.; Shen, S.; Liu, Z.; Luan, H.; and Sun, M. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of ACL*, volume 1, 2124–2133.

Lin, Z.; Feng, M.; Santos, C. N. d.; Yu, M.; Xiang, B.; Zhou, B.; and Bengio, Y. 2017. A structured self-attentive sentence embedding. *arXiv preprint arXiv:1703.03130*.

Liu, L.; Ren, X.; Zhu, Q.; Zhi, S.; Gui, H.; Ji, H.; and Han, J. 2017. Heterogeneous supervision for relation extraction: A representation learning approach. *arXiv preprint arXiv:1707.00166*.

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

Miwa, M., and Bansal, M. 2016. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv preprint arXiv:1601.00770*.

Miwa, M., and Sasaki, Y. 2014. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on EMNLP*, 1858–1869.

Ren, X.; Wu, Z.; He, W.; Qu, M.; Voss, C. R.; Ji, H.; Abdelzaher, T. F.; and Han, J. 2017. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on WWW*, 1015–1024.

Riedel, S.; Yao, L.; and McCallum, A. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 148–163. Springer.

Rink, B., and Harabagiu, S. 2010. Utd: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, 256–259. ACL.

Santos, C. N. d.; Xiang, B.; and Zhou, B. 2015. Classifying relations by ranking with convolutional neural networks. *arXiv preprint arXiv:1504.06580*.

Singh, S.; Riedel, S.; Martin, B.; Zheng, J.; and McCallum, A. 2013. Joint inference of entities, relations, and coreference. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, 1–6. ACM.

Tan, Z.; Wang, M.; Xie, J.; Chen, Y.; and Shi, X. 2017. Deep semantic role labeling with self-attention. *arXiv preprint arXiv:1712.01586*.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, 6000–6010.

Verga, P.; Strubell, E.; and McCallum, A. 2018. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. *arXiv preprint arXiv:1802.10569*.

Wang, W.; Yang, N.; Wei, F.; Chang, B.; and Zhou, M. 2017. Gated self-matching networks for reading comprehension and question answering. In *Proceedings of the 55th Annual Meeting of ACL*, volume 1, 189–198.

Wang, S.; Zhang, Y.; Che, W.; and Liu, T. 2018. Joint extraction of entities and relations based on a novel graph scheme. In *IJCAI*, 4461–4467.

Xiao, L., and Weld, D. S. 2012. Fine-grained entity recognition. In *AAAI Conference on Artificial Intelligence*.

Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; and Zhao, J. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th Interna-*

tional Conference on Computational Linguistics: Technical Papers, 2335–2344.

Zeng, D.; Liu, K.; Chen, Y.; and Zhao, J. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on EMNLP*, 1753–1762.

Zeng, X.; Zeng, D.; He, S.; Liu, K.; and Zhao, J. 2018. Extracting relational facts by an end-to-end neural model with copy mechanism. In *Proceedings of the 56th Annual Meeting of ACL*, volume 1, 506–514.

Zhang, D., and Wang, D. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Zhang, Y.; Zhong, V.; Chen, D.; Angeli, G.; and Manning, C. D. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on EMNLP*, 35–45.

Zhang, M.; Zhang, Y.; and Fu, G. 2017. End-to-end neural relation extraction with global optimization. In *Proceedings of the 2017 Conference on EMNLP*, 1730–1740.

Zheng, S.; Hao, Y.; Lu, D.; Bao, H.; Xu, J.; Hao, H.; and Xu, B. 2017a. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing* 257:59–66.

Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; and Xu, B. 2017b. Joint extraction of entities and relations based on a novel tagging scheme. *arXiv preprint arXiv:1706.05075*.

Zhou, P.; Shi, W.; Tian, J.; Qi, Z.; Li, B.; Hao, H.; and Xu, B. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of ACL*, volume 2, 207–212.