Bioinformatics, 37(11), 2021, 1581–1589
doi: 10.1093/bioinformatics/btaa993
Advance Access Publication Date: 27 November 2020
Original Paper

Data and text mining

A span-graph neural model for overlapping entity relation extraction in biomedical texts

Hao Fei[1], Yue Zhang[2], Yafeng Ren[3,*] and Donghong Ji[1]

[1]School of Cyber Science and Engineering, Wuhan University, Wuhan 430072, China, [2]School of Engineering, Westlake University, Hangzhou 310024, China and [3]Laboratory of Language and Artificial Intelligence, Guangdong University of Foreign Studies, Guangzhou 510420, China

*To whom correspondence should be addressed.

Associate Editor: Jonathan Wren

Received on March 13, 2020; revised on October 25, 2020; editorial decision on November 13, 2020; accepted on November 17, 2020

Abstract

Motivation: Entity relation extraction is one of the fundamental tasks in biomedical text mining, which is usually solved by the models from natural language processing. Compared with traditional pipeline methods, joint methods can avoid the error propagation from entity to relation, giving better performances. However, the existing joint models are built upon sequential scheme, and fail to detect overlapping entity and relation, which are ubiquitous in biomedical texts. The main reason is that sequential models have relatively weaker power in capturing long-range dependencies, which results in lower performance in encoding longer sentences. In this article, we propose a novel span-graph neural model for jointly extracting overlapping entity relation in biomedical texts. Our model treats the task as relation triplets prediction, and builds the entity-graph by enumerating possible candidate entity spans. The proposed model captures the relationship between the correlated entities via a span scorer and a relation scorer, respectively, and finally outputs all valid relational triplets.

Results: Experimental results on two biomedical entity relation extraction tasks, including drug–drug interaction detection and protein–protein interaction detection, show that the proposed method outperforms previous models by a substantial margin, demonstrating the effectiveness of span-graph-based method for overlapping relation extraction in biomedical texts. Further in-depth analysis proves that our model is more effective in capturing the long-range dependencies for relation extraction compared with the sequential models.

Availability and implementation: Related codes are made publicly available at http://github.com/Baxelyne/SpanBioER.

Contact: renyafeng@whu.edu.cn

1 Introduction

Detecting entities and their relations is the initial step toward extracting structured knowledge from raw texts. As a hot research topic in biomedical text mining community, automatic extraction of entities and relations can facilitate a line of biomedical tasks (Fei *et al.*, 2020b). In recent years, many related tasks have been proposed, such as adverse drug event (ADE) extraction (Gurulingappa *et al.*, 2012), protein–protein interaction (PPI) detection (Pyysalo *et al.*, 2007), drug–drug interaction (DDI) detection (Segura Bedmar *et al.*, 2013) and the bacteria biotope detection (Deléger *et al.*, 2016), etc.

Natural language processing (NLP) techniques are extensively employed for both biomedical entity and relation extraction. Existing work can largely be divided into two categories: pipeline methods and joint methods. Pipeline methods divide the task into two separate subtasks, containing entity mention recognition and relation classification. The entity mentions are first recognized by named entity recognition (NER) techniques. Then, relations between entity pair are examined by classification models. Pipeline methods have long been explored, but they suffer from the error propagation problem from entity recognition to relation classification (Li *et al.*, 2017; Wang *et al.*, 2018).

To avoid the error propagation between two subtasks and fully capture the relationship between two subtasks, joint methods are proposed for the end-to-end extraction, which can better integrate the information of entities and relations, achieving improved performances (Katiyar and Cardie, 2017; Ren *et al.*, 2017). For example, Li *et al.* (2017) employ a bi-directional long short-term memory (BiLSTM) network with syntax information for entity mention detection, and then learn relation representations of two target entities with their shortest dependency path (SDP) for ADE task. Zhang *et al.* (2017) leverage a table-filling method for jointly decoding relation triplets. However, these methods fail to identify overlapping entities and relations, which are ubiquitous in biomedical texts. Taking as the example in Figure 1, total eight pairs of overlapping drug-to-drug interaction relational triplets co-exist in one sentence. Furthermore, we can find that the entity pair of *sympathomimetics* and *antidiabetic drugs* is far away in distance.

To solve this problem, recent efforts are paid to dealing with the overlapping relations (Fei *et al.*, 2020a; Takanobu *et al.*, 2018; Wang *et al.*, 2018; Zeng *et al.*, 2018). For instance, Zeng *et al.* (2018) model relation extraction as a triplet generation task. Takanobu *et al.* (2018) introduce a reinforcement learning method to detect nested relations. However, these models fail to give satisfactory results when dealing with biomedical texts. The main reason is that these systems have relatively weaker power in capturing long-range dependencies between entity pairs, which results in lower performance in encoding longer sentences of biomedical texts.

In this article, inspired by the success of neural span-graph models in a range of NLP tasks, such as coreference resolution (Lee *et al.*, 2018), semantic role labeling (Fei *et al.*, 2020c), machine reading comprehension (Li *et al.*, 2019) and overlapping entity mention extraction (Luan *et al.*, 2019), we propose to build a novel span-graph neural model for jointly extracting overlapping entity and relation in biomedical texts. Unlike traditional sequential models, our method treats the task as relation triplets prediction. In particular, we investigate three different relation scorers, in order to mitigate the long-distance dependencies and enhance the communication of entity pairs. In addition, two pruning strategies are exploited during decoding for controlling the model complexity. Note that, Dixit and Al-Onaizan (2019) first construct a span-level graph model for detecting the overlapping entities and relations. Compared with their work, our motivations and network structure are highly different. Specifically, their method employs a simple feed-forward network (FFN) to measure the relation between entities, and the model is very limited in capturing long-distance dependencies between the entity pairs, while our model considers more thorough communication on the relation scorers for detecting the relations.

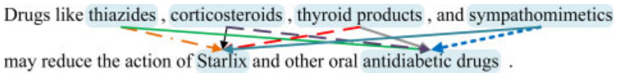
Drugs like thiazides , corticosteroids , thyroid products , and sympathomimetics
may reduce the action of Starlix and other oral antidiabetic drugs .

Fig. 1. An example of DDI. The entities are in the blue boxes, and the directed arrows in different colors indicate the relations

We conduct experiments on two benchmarks (DDI and PPI) where the overlapping entity relations are overwhelming. Results show that our method outperforms previous methods and baseline systems by a large margin. Specifically, we obtain 68.02% $F1$ score in DDI and 80.04% $F1$ score in PPI for relation extraction, respectively. Further in-depth analysis indicates that our model is more effective in capturing the long-distance dependencies for relation extraction compared with the existing methods.

2 Model

We model the overlapping relation extraction as a quintuple prediction (to generalize the problem, we also consider the entity type label. Therefore, the prediction becomes a quintuple) via a span-graph neural model. Given an input sentence $\boldsymbol{S} = \{w_1, w_2, \ldots, w_n\}$, the model aims to output a set of quintuples:

$$\boldsymbol{Y} = \{(e_s, l_s, r, e_t, l_t) | e \in \boldsymbol{E}, r \in \boldsymbol{R}, l \in \boldsymbol{L}\}, \quad (1)$$

where subscripts s and t represent the source and target, respectively. $\boldsymbol{E} = \{(e_i, \ldots, e_j) | 1 \leq i \leq j \leq n\}$ is the set of candidate entity spans, and $\boldsymbol{R}$ is the set of the relation labels including a null label ϵ indicating no relation between an entity pair, $\boldsymbol{L}$ is the set of the entity labels. l_s and l_t are the corresponding entity type. Note that

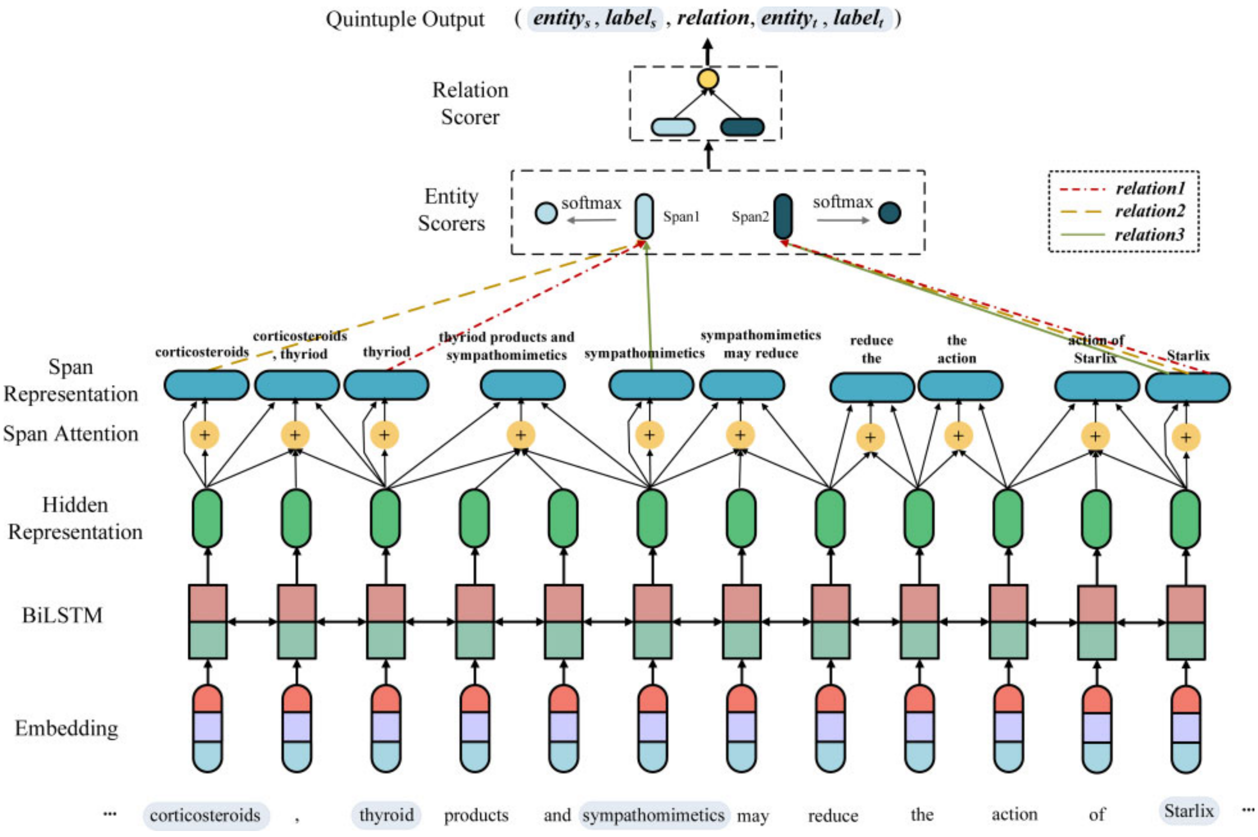

Fig. 2. Overall framework of the proposed span-graph based model for overlapping entity relation extraction. For simplifying the illustration, we only show a small subset of the spans

$(e_a, l_a, r, e_b, l_b) \neq (e_b, l_b, r, e_a, l_a)$ when the relation between two entities is strictly directed.

The overall framework of the proposed method is illustrated in Figure 2. The model first takes as input the features embedding, and then a BiLSTM layer is used to encode token representation into contextualized hidden representation. Afterwards, the model enumerates all possible candidate entity span representations via a span attention layer, and builds the span-graph. Then, the span and relation scorers dynamically measure all possible relation triplets. Finally, once the relation between an entity pair is decided, the model outputs the relational quintuple.

2.1 Input representation

Given an input sequence $\boldsymbol{S}$, for each word w_i, we use a look-up table $\boldsymbol{E} \in \mathbb{R}^{L \times V}$ (L is the dimension of embedding, V is the vocabulary size) to obtain its embedding $\boldsymbol{e}_i \in \mathbb{R}^L$. Character-level features (such as the prefix or suffix of a word) have been shown to be effective for neural NER models (Ren *et al.*, 2018). For example, the suffix 'roid' is a sharp morphological information to indicate a kind of hormone entity, such as the biomedical entity 'corticosteroid' or 'thyroid'. We use a convolutional neural network (CNN) (Kim, 2014) to encode morphological information from characters inside a word $\boldsymbol{w}_i$ into character-level representation $\boldsymbol{c}_i$.

Previous work also shows that syntactic features are useful for relation extraction. We further employ the corresponding part-of-speech (PoS) labels $\{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_n\}$, dependency labels $\{\boldsymbol{d}_1, \boldsymbol{d}_2, \ldots, \boldsymbol{d}_n\}$. We use two separate look-up table $\boldsymbol{E}_p$ and $\boldsymbol{E}_d$ to obtain the corresponding embedding vectors $\boldsymbol{p}_i$ and $\boldsymbol{d}_i$ for each word. Besides, the position information can help to better inform the relative contribution of each token under global scope. We obtain the position representation $posi_i$ via a trainable embedding $\boldsymbol{E}_{posi}$. Finally, we concatenate the above word-level features into a unified embedding vector as follows:

$$\boldsymbol{x}_i = [\boldsymbol{e}_i; \boldsymbol{c}_i; \boldsymbol{p}_i; \boldsymbol{d}_i; posi_i]. \tag{2}$$

2.2 Encoder

We use a 3-layer BiLSTM to encode the input embedding $\boldsymbol{x}$ to fully generalize and contextualize the representation. BiLSTM encodes the input from both directions, yielding hidden representations $\boldsymbol{h}_{f_i}$ and $\boldsymbol{h}_{b_i}$, respectively. We concatenate two directional hidden states at each time step i as the sequence representation $\boldsymbol{h}_i = \{\boldsymbol{h}_1, \boldsymbol{h}_2, \ldots, \boldsymbol{h}_n\}$. The contextualized representation is intended to capture the information shared for all the downstream span learning.

2.3 Span representation

Thereafter, the model iteratively builds all entity span representation $\boldsymbol{x}_{sp}$ over the following representation:

$$\boldsymbol{x}_{sp} = [\boldsymbol{h}_{start}; \boldsymbol{h}_{end}; \boldsymbol{h}_{sp}; \mathbf{size}(sp)], \tag{3}$$

where $\boldsymbol{h}_{start}$ and $\boldsymbol{h}_{end}$ are boundary representation of the start and the end token, respectively. sp denotes a span, $\mathbf{size}(sp)$ is the embedding vector standing for the span width, and $\boldsymbol{h}_{sp}$ is the span attention representation over the tokens involved in current span sp, which can be obtained as follows:

$$\begin{aligned} \boldsymbol{v}_t &= \boldsymbol{V} \cdot \tanh(\boldsymbol{W}_{att} \cdot x_t), \\ \boldsymbol{\alpha}_t &= softmax(v_t), \boldsymbol{h}_{sp} = \sum_{t=start}^{end} \alpha_t \cdot x_t, \end{aligned} \tag{4}$$

where $\boldsymbol{W}_{att}$ and $\boldsymbol{V}$ are attention parameters.

2.4 Scorers

We build separate output layers to decide whether a pair of candidate entity spans sp_s, sp_t is valid, and their relation r. We reach the goal by measuring the corresponding scores via a span scorer and a relation scorer, respectively. First, we measure the possible candidate entity span via a span scorer, which is a FFN (He *et al.*, 2018):

$$\Phi_{sp} = \boldsymbol{W}_{sp} \cdot \mathrm{FFN}(\boldsymbol{x}_{sp}). \tag{5}$$

One key problem in entity and relation extraction is the long-distance dependency problem. When the distance between two interrelated entities becomes large, the difficulty to identify their relation significantly increases. An intuitive solution is to allow more comprehensive and sufficient communication between entity pairs. Here, we explore three types of relation scorers: the biaffine attention relation scorer, SDP relation scorer and graph convolutional network (GCN) relation scorer.

2.4.1 Biaffine attention relation scorer

The biaffine attention scorer is first proposed for better parsing dependency (Dozat and Manning, 2016), it measures the relation between two input representations via a biaffine transformation function:

$$\boldsymbol{\Phi}_{\mathrm{r}}^{\mathrm{BA}}(\mathrm{e_s}, \mathrm{e_t}) = \boldsymbol{x}_{\mathrm{s}}^{\mathrm{T}} \cdot \boldsymbol{W}_1 \cdot \boldsymbol{x}_{\mathrm{t}} \quad + \boldsymbol{W}_2 \cdot [\boldsymbol{x}_{\mathrm{s}}; \boldsymbol{x}_{\mathrm{t}}] + \boldsymbol{b}, \tag{6}$$

where $\boldsymbol{W}_1$, $\boldsymbol{W}_2$ and b are parameters.

2.4.2 SDP relation scorer

Previous work also shows the usefulness of SDP syntax structure for relation classification (Miwa and Bansal, 2016). Specifically, we split SDP into left sub-path and right sub-path, each from an entity span to the common ancestor node. The input of the scorer is the embeddings of each token word in the path between two target entities (note that, the input of the first left or last right entity span is the span representation x_{sp}, instead of its word embedding x_i.). A Tree-LSTM model (Miwa and Bansal, 2016) is used to encode each path according to the direction of the dependency labels. Based on the representations, the max pooling operation yields the representation of two sub-paths. Finally, we concatenate the pooling results of two paths as follows:

$$\boldsymbol{\Phi}_{\mathrm{r}}^{SDP}(\mathrm{e_s}, \mathrm{e_t}) = [\underset{\mathrm{i} \in left_{\mathrm{p}}(\mathrm{e_s}, \mathrm{a})}{\mathrm{MaxPool(i)}}; \underset{\mathrm{i} \in right_{\mathrm{p}}(\mathrm{e_t}, \mathrm{a})}{\mathrm{MaxPool(i)}}], \tag{7}$$

where $left_p(e_s, a)$ and $right_p(e_t, a)$ are the left and right paths, respectively, and a is the shortest common ancestor node.

2.4.3 GCN relation scorer

GCN (Duvenaud *et al.*, 2015) has been used for modeling the underlying dependencies of nodes in the graph while maintaining the semantic information (Marcheggiani and Titov, 2017). For the dependency graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ constructed between start span node sp$_s$ and target span node sp$_t$, where $\mathcal{V}$ and $\mathcal{E}$ are sets of nodes and dependency edges, respectively, a GCN layer encodes nodes representation as follows:

$$GCN(\mathcal{G}) = ReLU(\sum_{i \in \mathcal{N}(i)} (\boldsymbol{W}_{gcn} \cdot x_i + b)), \tag{8}$$

where $\boldsymbol{W}_{gcn}$ and b are parameters, $\mathcal{N}(i)$ are neighbors of i and ReLU is a non-linear activation function. Specifically, we use a two-layer GCN architecture for yielding a relation score:

$$\boldsymbol{\Phi}_{\mathrm{r}}^{GCN}(\mathrm{e_s}, \mathrm{e_t}) = GCN(\mathcal{G}). \tag{9}$$

2.5 Training

During training, we use a cross-entropy loss, optimizing the probability $P_\theta(\hat{\boldsymbol{Y}}|\boldsymbol{S})$ of the quintuple $y_{(e_s, l_s, r, e_t, l_t)}$, given a sentence $\boldsymbol{S}$. Specifically,

$$\begin{aligned} P_\theta(\boldsymbol{Y}|\boldsymbol{S}) &= \prod_{e \in E, l \in L, r \in R} P_\theta(y_{(e_s, l_s, r, e_t, l_t)}|\boldsymbol{S}) \\ &= \prod_{e \in E, l \in L, r \in R} \frac{\boldsymbol{\Phi}(\mathrm{e_s}, \mathrm{l_s}, \mathrm{r}, \mathrm{e_t}, \mathrm{l_t})}{\sum_{\hat{l} \in L, \hat{r} \in R} \boldsymbol{\Phi}(\mathrm{e_s}, \hat{\mathrm{l}}_{\mathrm{s}}, \hat{\mathrm{r}}, \mathrm{e_t}, \hat{\mathrm{l}}_{\mathrm{t}})}, \end{aligned} \tag{10}$$

where θ is the parameters of the model, $\boldsymbol{\Phi}(\mathrm{e_s}, \mathrm{l_s}, \mathrm{r}, \mathrm{e_t}, \mathrm{l_t})$ represents the total score:

$$\boldsymbol{\Phi}(\mathrm{e_s},\mathrm{l_s},\mathrm{r},\mathrm{e_t},\mathrm{l_t})=\boldsymbol{\Phi}_{\mathrm{sp}}(\mathrm{e_s})+\boldsymbol{\Phi}_{\mathrm{sp}}(\mathrm{e_t})+\boldsymbol{\Phi}_{\mathrm{r}}(\mathrm{e_s},\mathrm{e_t})+\mathrm{FFN}(\boldsymbol{x}_{\mathrm{sp}}(\mathrm{s}))+\mathrm{FFN}(\boldsymbol{x}_{\mathrm{sp}}(\mathrm{t})), \tag{11}$$

where the additional FFN($\cdot$) measures the scores for entity label l_s and l_t based on their span representations, as the span scorer [Equation (5)] is used only to measure the probability of a span to be an entity mention.

Finally, the objective is to minimize the negative log likelihood of the golden structure:

$$\ell = -logP_\theta(\boldsymbol{Y}|\boldsymbol{S}). \tag{12}$$

Note that, the score $P_\theta(y_{(e_s,l_s,\epsilon,e_t,l_t)}|\boldsymbol{S})$ of null relation label ϵ is assigned to 0, which represents an invalid relational triplet. If the relation scorers find the relational type $r \neq \epsilon$, there exists a certain relation type between the pair of entities. In other words, the relation scorers relate simultaneously to the relation existence and the relational type.

2.6 Inference

In inference stage, the well-trained model will output all possible entity relation quintuples $(sp_s, l_s, r, sp_t, l_t)$ in Equation (1). In particular, after the measurement by relation scorer, a softmax classifier is used to predict whether there is a relation label r between two entity spans:

$$r = \mathrm{softmax}(\boldsymbol{\Phi}_{\mathrm{r}}). \tag{13}$$

If the relation label is not null ϵ, the candidate span sp_s and sp_t will be output according to their start and end index. Meanwhile, another softmax classifier is used to decide the entity labels l_s and l_t for the corresponding entities, respectively:

$$l = \mathrm{softmax}([\boldsymbol{\Phi}_{\mathrm{sp}}; \boldsymbol{x}_{sp}]). \tag{14}$$

The decoding process is shown in Algorithm 1.

Algorithm 1 Decoding procedure of the proposed model.

```
Input:
Given input sentence $S = \{w_1, w_2, \ldots, w_n\}$, the corresponding PoS labels $\{P_1, P_2, \ldots, P_n\}$ and dependency labels $\{D_1, D_2, \ldots, D_n\}$.
Output:
for each word $w_i$ in sequence $S$ do
    Convert word $w_i$ into embedding $e_i$.
    Calculate char representation $c_i$ for $w_i$.
    Convert POS label $P_i$ into embedding $p_i$.
    Convert dependency label $D_i$ into embedding $d_i$.
end for
Concatenate all embeddings into $x_i$.
Use BiLSTM to encode $x$ and output contextualized representation $h_i$.
Build all the entity span representation $x_{\mathrm{sp}}$.
Measure the likelihood for candidate entity span $x_{\mathrm{sp}}$ by entity scorer $\Phi_{sp}$.
Measure the relation score for each candidate entity span pair by relation scorer $\Phi_r(e_s, e_t)$.
Prune the span-graph:
1) Drop the spans whose score $\Phi$ is out of the beam $\beta \cdot n$.
2) Cache the spans whose width $size_{sp} < K$.
for each pair of candidate entities $(e_s, e_t)$ do
    Compute the relation label $r$ for $(e_s, e_t)$ via Equation (13).
    if $r \neq \epsilon$ then
        Compute the entity labels $l_s$ and $l_t$ for $e_s$ and $e_t$ via Equation (13).
        Output the triplet $(e_s, l_s, r, l_t, e_t)$.
    end if
end for
```

2.7 Pruning and optimization

As our model enumerates all possible entity relation quintuples, its time complexity is proportional to the number of spans given a sentence of n words, i.e. $N = \frac{n(n+1)}{2}$. We take two pruning strategies for optimizing the computation efficiency.

- We define a beam for storing the span candidates, and the sizes of the beams are limited by $\beta \cdot n$ (n is the length of sentence). All candidate spans are ranked by their unary score Φ, and only the spans within the beam are stored.
- We limit the maximum width of a span with a fixed number K, to reduce the computation for scorers.

Besides, for the non-overlapping relation, once a pair of candidate entities is assigned with a relation label, they will be removed from the candidate set. In addition, if the relation extraction task does not consider the order between the entity pair, we can reduce the entity pairs into half before scoring their relations.

3 Experimental settings

3.1 Datasets

We conduct experiments on the DDI and PPI datasets, both of which are the benchmark for biomedical relation extraction. The DDI dataset was first published for DDI detection task on SemEval 2013 (Segura Bedmar *et al.*, 2013), including two sub datasets: *DrugBank* and *Medline*. There are four types of relations between drug entities, including *Advice*, *Mechanism*, *Effect* and *Int*. The PPI dataset includes five widely used sub sets: *AIMed* (Bunescu *et al.*, 2005), *BioInfer* (Pyysalo *et al.*, 2007), *IEPA* (Ding *et al.*, 2002), *HPRD50* (Fundel *et al.*, 2007) and *LLL* (Nédellec, 2005). There is either *True* or *False* relational label between the protein entities in the PPI dataset. The statistics of the datasets are shown in Table 1. We divide the DDI dataset into training set and test set by following previous work (Sun *et al.*, 2018). Likewise, we follow previous settings for the PPI dataset (Ahmed *et al.*, 2019).

Table 1. Summary of datasets

Dataset		# Sent.		# E–R pairs		# Rel.
		Total	OL.	Total	OL.	
DDI	DrugBank	5675	596	3805	1827	4
	Medline	1301	37	232	59	—
	Total	6976	633	4037	1886	—
PPI	LLL	77	42	330	253	2
	AIMed	1943	685	5775	4613	—
	BioInfer	1100	817	9666	8576	—
	IEPA	486	111	817	331	—
	HPRD50	145	76	433	288	—
	Total	3751	1731	17 021	14 061	—

Note: *OL.* denotes *overlapping* and # Rel. denotes the number of relation labels.

Table 2. Parameters setting of networks

Param.	Value	Param.	Value
dim(e_i)	300	dim(p_i)	50
dim(c_i)	50	dim(d_i)	80
dim($posi_i$)	25	dim(sp)	200
dim(BiLSTM)	300	dim(Biaffine)	200
dim(TreeLSTM)	200	dim(GCN)	200
dropout(word)	0.4	filter(cnn)	50
dropout(char, pos, dep, posi)	0.1	size(cnn)	[3,4,5]
Optimizer	Adam	Batch size	16
Learning rate	0.0001	Iterations	200

3.2 Parameter settings

Word embeddings are randomly initialized with uniform samples from $[-\sqrt{\frac{6}{r+c}}, +\sqrt{\frac{6}{r+c}}]$, where r and c are the number of rows and columns in the structure, respectively. The dimension of character representations is set to 50. The convolutions in character CNN use three different window sizes [3, 4, 5], each consisting of 50 filters. BiLSTM is employed with 200 dimensional hidden states. During training, we use the Adam (Kingma and Ba, 2014) optimization method with an initial learning rate of 0.0001. The span beam width β and the maximum width of span K are determined by fine-tuning based on the development sets. We use the mini-batch with a size of 16, training 10 k iteration with early stopping. We apply a 0.5 dropout ratio for word embeddings and character CNN, and a 0.2 dropout ratio for all hidden layers and feature embeddings. All experiments are conducted with a NVIDIA GeForce GTX 1080Ti GPU and 11 GB graphic memory. The detailed setting of parameters is listed in Table 2.

The SDP and GCN relation scorers take as input the dependency tree features over the input sentence. To this end, we adopt the state-of-the-art BiAffine dependency parser (Dozat and Manning, 2016) to parse the relevant dataset. Being trained on English Penn Treebank corpus (Marcus *et al.*, 1993), the dependency parser has 95.2% UAS and 93.4% LAS, respectively. Moreover, a PoS tagger is used to produce PoS tags for sentences, which is trained on the Universal Dependency Treebank v1.4 dataset (https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-1827), with an accuracy of 95.15%.

3.3 Evaluation metrics

We adopt the precision, recall and $F1$ score to evaluate the models with respect to the entity (Ent.) and relation (Rel.). The prediction of the relation is considered as correct only when the relation label and two entities are both correct. Following previous work on these two datasets, we do not consider the entity labels, which mean that the task becomes the triplets prediction. We test the performances of our method 30 times on all the corresponding test set, and results are presented after significance test with $P \leq 0.015$.

We measure the performances for joint extraction of the entity and relation. For joint extraction task, we combine the sub datasets for DDI and PPI, respectively. Since pipelined baselines only perform the relation classification given the gold entities in sentences, we also conduct experiments where our model only measures the performances on the relation classification for fair comparisons with them. Technically, given an input sentence $\{w_1, w_2, \ldots, w_n\}$, we use *BIO* labels to tag the tokens as $\{l_1, l_2, \ldots, l_n\}$ $(l_i \in \{B, I, O\})$, by which we let the model understand which tokens are the corresponding gold entities. We only consider the metrics for the relation classification tasks on the corresponding test set.

3.4 Baseline systems

To show the effectiveness of the proposed model, we compare our model with two types of baselines: the models that consider relation classification in pipeline procedures and the joint models for entity and relation co-detection.

3.4.1 Pipeline methods

(i) Liu *et al.* (2016b) use a CNN model for DDI classification. (ii) Sahu and Anand (2018) employ LSTM for DDI task. (iii) Yi *et al.* (2017) use LSTM with attention mechanism. (iv) Liu *et al.* (2016a) propose a dependency-based CNN model for DDI. (v) Dewi *et al.* (2017) use a deepened CNN model for DDI. (vi) Sun *et al.* (2018) deepen the CNN layers based on the work of Dewi *et al.* (2017) for DDI. (vii) Zhang *et al.* (2018) combine recurrent neural network and CNN for PPI classification. (viii) Chang *et al.* (2016) present an interaction pattern tree kernel method for PPI. (ix) Peng *et al.* (2015) build dependency graph for better classification of PPI task. (x) Hsieh *et al.* (2017) use a BiLSTM model for PPI. (xi) Ahmed *et al.* (2019) construct a tree LSTM and integrate structured attention for improving the performance of PPI classification.

3.4.2 Joint methods

(i) LSTM-CRF is taken as a baseline sequential labeling model (Lample *et al.*, 2016). (ii) LSTM-LSTM is a sequential joint model which uses LSTM as encoder, and CRF or LSTM as decoder respectively (Vaswani *et al.*, 2016). We employ them with end-to-end tagging scheme. (iii) LSTM-SDP is a joint model, using sequential LSTM to decode entities, and tree-LSTM based on SDP to decode relations (Li *et al.*, 2017). (iv) GlobalE is a table-filling method with global optimization and syntax information for end-to-end neural relation extraction (Zhang *et al.*, 2017). (v) Tagging is an end-to-end method by integrating the entity and relation extraction tasks into one unified *BIO* tagging scheme (Zheng et al., 2017b). (vi) BiLSTM-ED is a joint method, including encoder–decoder BiLSTM for entity extraction, and CNN for relation classification (Zheng *et al.*, 2017a). (vii) DAG is an end-to-end method, which constructs a directed graph for entities and relations by using a transition-based parsing framework (Wang *et al.*, 2018). (viii) CopyR is a joint method by using a sequence-to-sequence network with copy mechanism (Zeng *et al.*, 2018). (ix) HRL is a joint model, which applies a hierarchical reinforcement learning framework to enhance the interaction between two subtasks (Takanobu *et al.*, 2018). (x) SpanRE is a span-graph model for joint overlapping detection (Dixit and Al-Onaizan, 2019), which is also similar to ours in span-level encoding, but uses different mechanism in measuring relation between entity pair.

3.5 Development experiments

We introduce pruning strategies during decoding for improving the efficiency. Based on development experiments, we study the complexity and efficiency of the model on the DDI dataset.

3.5.1 Entity coverage

Since our model can enumerate all possible entity spans, so the most nested entity texts can be detected. Table 3 shows the entity and relation coverage under different setting of maximum width of spans. With a max length of 6, the model covers 99.96% entities and 99.78% relations on the dataset. When the maximum width is set to 10, the entities and relations coverage both increase to 99.99%.

3.5.2 Maximum spans width

As shown in Figure 3, when the maximum width of spans $K = 4$, the model obtains a trade-off between the efficiency and the decoding speed of around 230 sentence per minute. This is plausible because the span width of majority of entities is <4.

3.5.3 Span beam width

Figure 4 shows the $F1$ score and decoding speed against different span beam width ratio β. With the ratio =0.4, the model can maintain a close-to-full performance, yet keeping an acceptable decoding speed of about 135 sentence per minute. While our model exhaustively creates representations for candidate entity span, it achieves the dual goals of being memory efficient and capturing most of the

Table 3. Entity and relation coverage under different setting of maximum width of spans

Max. width K	Ent. (%)	Rel. (%)
10	99.99	99.99
8	99.98	99.85
6	99.96	99.78
4	99.56	97.20
2	95.36	93.20
1	87.24	79.06

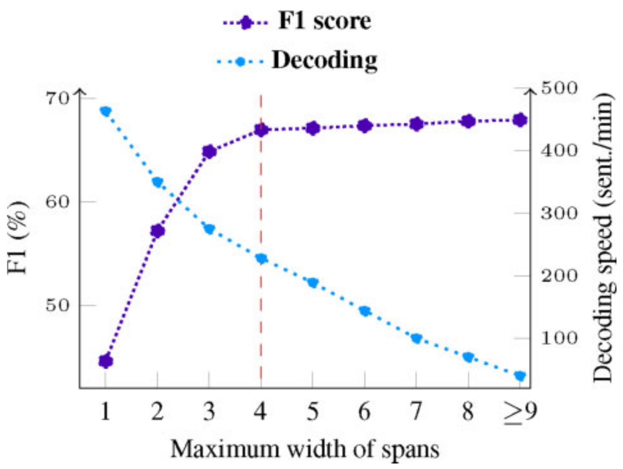

Fig. 3. Performance and decoding speed under different maximum width of spans K

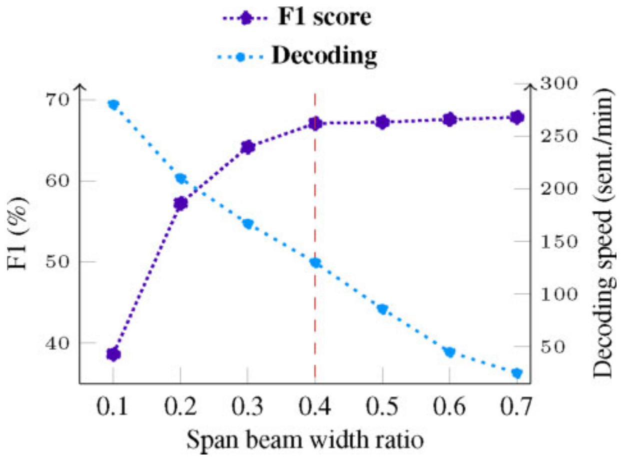

Fig. 4. Performance and decoding speed under different span beam width ratio β

entities and relations in the space of the spans considered. The observation coincides with the study of Dixit and Al-Onaizan (2019).

4 Main results

4.1 Joint entity relation extraction

We first report the performances on joint entity relation extraction. For the flat extraction baselines, we preprocess the datasets into non-overlapping by keeping only one randomly chosen triplet while removing the rest. As shown in Table 4, our model (ensemble model) gives the highest $F1$ scores on both entity and relation extraction against all baseline systems, with 75.05% and 68.02% for entity and relation extraction on DDI dataset, respectively, and 90.08% and 80.04% on PPI dataset, respectively. Second, all models that handle overlapping relational triplets significantly outperform flat extraction methods. Notably, ours outperforms the best flat baseline BiLSTM-ED, with an improvement of 7.83% on DDI and 10.72% on PPI on overlapping relation extraction. Furthermore, we can find that the SpanRE model gives an overall better performance, thanks to its exhaustive span-graph architecture. This demonstrates the importance of span-graph framework for addressing the overlapping problem in biomedical entity relation extraction.

Besides, compared with the SpanRE, our model is stronger on the relation detection tasks by a large margin. In particular, the model with the GCN scorer is the most powerful. This comparison demonstrates the effectiveness of our method on modeling the long-distance dependencies between entity pairs for relation classification. As a result, the improvements of our model on relation extraction are more significant than entity recognition. The above analysis shows the effectiveness of the proposed joint model for the task.

Table 4. Joint extraction results

System		DDI		PPI	
		Ent.	Rel.	Ent.	Rel.
Flat	LSTM-CRF	58.51	49.22	72.16	58.36
	LSTM-LSTM	59.34	50.64	74.31	60.18
	LSTM-SDP	62.03	54.98	79.67	65.21
	Tagging	63.38	55.34	78.64	65.92
	GlobalE	67.34	59.75	80.34	67.82
	BiLSTM-ED	67.86	60.19	81.32	69.32
Overlap	DAG	70.91	62.87	84.27	72.84
	CopyR	71.64	63.87	83.69	74.92
	HRL	71.25	63.11	85.18	73.37
	SpanRE	73.05	64.40	87.25	75.36
	Ours(Φ_r^{BA})	73.32	65.86	88.62	76.74
	Ours(Φ_r^{SDP})	73.97	66.40	89.92	78.86
	Ours(Φ_r^{GCN})	74.83	67.61	89.32	79.86
	Ours(†)	75.05	68.02	90.08	80.04

Note: Flat means the models for flat extraction and *Overlap* means the models for overlapping extraction. (†) represents the ensemble model by integrating three scorers.

Table 5. Results of relation classification on DDI

System		DDI-ALL		
		Precision	Recall	$F1$
Pipeline	Liu *et al.* (2016b)	75.72	64.66	69.75
	Sahu and Anand (2018)	73.41	69.66	71.48
	Yi *et al.* (2017)	73.67	70.79	72.20
	Liu *et al.* (2016a)	77.21	64.35	70.19
	Dewi *et al.* (2017)	86.18	87.20	86.27
	Sun *et al.* (2018)	87.99	82.73	84.50
Joint	DAG	91.35	87.82	89.22
	CopyR	92.04	89.03	90.43
	HRL	91.98	88.34	89.55
	SpanRE	92.51	90.31	91.65
	Ours(†)	94.85	92.04	93.42

4.2 Relation classification

We further compare the performances on the separate relation classification task. The results on the DDI dataset are shown in Table 5. First, there are a salient performance gaps between joint models and pipeline models. This coincides with the fact that joint methods can effectively leverage information of entities and relations, and also reduce the error propagation. Second, our model achieves the best performance compared with all previous systems, obtaining a gain of

Table 6. Results on separate PPI

System		AIMed	BioInfer	IEPA	HPRD50	LLL	Avg.
Pipeline	Zhang *et al.* (2018)	56.40	61.30	75.10	63.40	76.50	66.54
	Chang *et al.* (2016)	60.60	69.40	71.40	71.50	80.60	70.70
	Peng *et al.* (2015)	61.10	58.70	72.90	79.90	84.60	71.44
	Hsieh *et al.* (2017)	76.90	87.20	76.31	80.51	78.30	79.84
	Ahmed *et al.* (2019)	81.60	89.10	78.50	82.00	84.80	83.20
Joint	DAG	87.27	89.47	83.51	84.67	86.13	86.21
	CopyR	84.57	92.26	81.26	86.19	90.88	87.03
	HRL	85.84	93.07	81.75	84.45	89.08	86.84
	SpanRE	86.71	94.74	82.49	87.60	91.30	88.56
	Ours(†)	88.27	96.21	83.90	89.57	92.86	90.16

Note: The performances are measured by *F*1-score.

Table 7. Results on ablation study

Feature	DDI	PPI
# word	65.21	75.10
+ charCNN	65.81	76.98
+ PoS label	66.34	77.35
+ position	67.08	78.61
+ dependency label	66.67	78.02
All syntax	68.02	80.04

Note: # word represents using only randomly initialized word embedding.

1.77% in *F*1 score compared with SpanRE. The main reason is that the proposed model can enumerate all possible entity spans, detecting all potential relation triplets between them.

The same observation can be found in the results on the PPI datasets, as shown in Table 6. First, our model yields the best performance on all sub sets, with an average of 90.16% *F*1 score. For the *BioInfer* dataset, which accounts for the largest number of overlapping relation triplets in all datasets, our model also gives the best result (95.21% *F*1 score), evidently proving the extraordinary capability for overlapping relation extraction. Finally, the joint methods universally outperform pipeline models.

4.3 Ablation study

We investigate the contributions from relation scorers and input features, respectively.

4.3.1 Relation scorers

From Table 4, we can know that the three relation scorers vary in performances. Generally, the SDP and GCN scorers are better than the Biaffine scorer. This is reasonable because the SDP and GCN scorers can encode more underlying relations between nodes in global scope. Additionally, the GCN scorer helps to achieve better performances for determining the relations. By integrating three scorers, the best performances can be obtained.

4.3.2 Input features

Table 7 shows the results where we add one of these items, respectively. First, the performance can further be improved by integrating the char and PoS label information. Second, the dependency label gives a more significant performance boost than other syntax features, which coincides with the fact that syntax information is crucial for entity relation extraction. By integrating all these features, our model achieves the state-of-the-art performances.

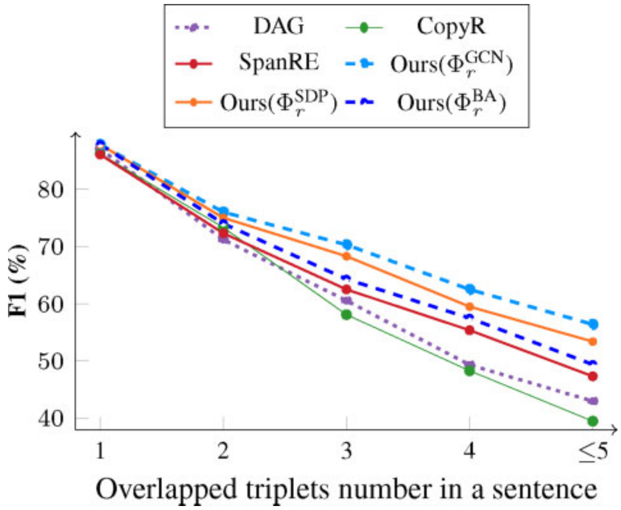

Fig. 5. Performances against varying numbers of overlapping triplets

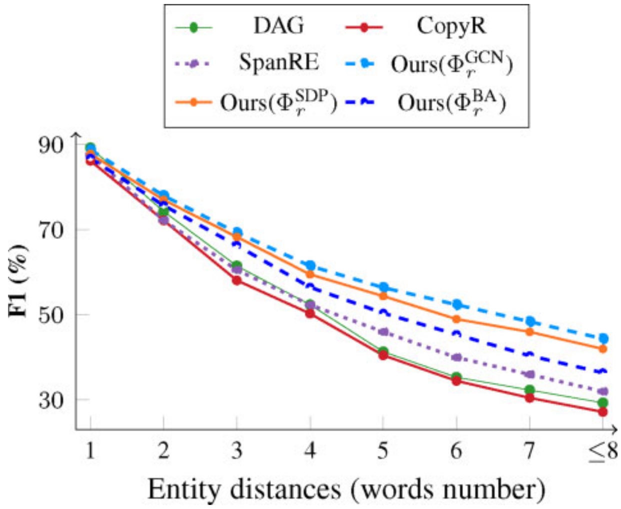

Fig. 6. Performances against varying entity distances

4.4 Overlapping relation extraction

We study the ability of our model on overlapping relation extraction, by calculating *F*1 score on the sentences under varying numbers of overlapping triplets, based on the PPI dataset. As shown in Figure 5, with increasing numbers of overlapping relation triplets, our model gradually outperforms other systems. Furthermore, the more the overlapping triplets are in a sentence, the higher improvements our model can yield, compared with other systems. We can find that SpanRE is much closer to our model (with Biaffine scorer), and far better than other sequential baselines, showing the superiority of span-graph methods for overlapping relations.

4.5 Impact of entity distance

We compare the performances by differing entity distances on the PPI test set. Figure 6 shows the results. For all different entity distances, our model achieves consistently better results than the baselines. Note that even the distance is increased to five, our model can still accurately detect the relations compared with other baseline systems. We can find that the performance of the SpanRE model declines when the distance grows, which indicates that our relation scorers can partially address the problem of long-distance dependency. Besides, scorers with rich structure information (i.e. GCN and SDP) can achieve better performance than the biaffine scorer, showing the necessity on providing sufficient interaction between entity pairs.

5 Conclusion

We proposed a span-graph neural model for jointly detecting overlapping entity relation in biomedical texts, treating the extraction task as relational triplets prediction. Three types of relation scorers were proposed for offering more sufficient communication on measuring the relations between entity pairs. Results on two benchmarks showed that the model outperformed all the strong baselines, demonstrating the effectiveness of the proposed method for overlapping entity relation extraction in biomedical texts.

Funding

This work was supported by the National Natural Science Foundation of China [grant numbers 61702121, 61772378]; the Research Foundation of Ministry of Education of China [No. 18JZD015]; and the Natural Science Foundation of Guangdong Province [No. 2020A151501705].

Conflict of Interest

none declared.

References

Ahmed,M. *et al.* (2019) Identifying protein-protein interaction using tree LSTM and structured attention. In: *Proceedings of 2019 IEEE International Conference on Semantic Computing*. Newport Beach, California, USA, pp. 224–231.

Bunescu,R. *et al.* (2005) Comparative experiments on learning information extractors for proteins and their interactions. *Artif. Intell. Med.*, **33**, 139–155.

Chang,Y.-C. *et al.* (2016) PIPE: a protein–protein interaction passage extraction module for biocreative challenge. Berlin, Germany, *Database*, **2016**, baw101.

Deléger,L. *et al.* (2016) Overview of the bacteria biotope task at BioNLP shared task 2016. In: *Proceedings of the 4th BioNLP Shared Task Workshop*. pp. 12–22.

Dewi,I.N. *et al.* (2017) Drug-drug interaction relation extraction with deep convolutional neural networks. In: *Proceedings of 2017 IEEE International Conference on Bioinformatics and Biomedicine*. Kansas City, USA, pp. 1795–1802.

Ding,J. *et al.* (2002) Mining medline: abstracts, sentences, or phrases? In: *Proceedings of Pacific Symposium on Biocomputing*. Kauai, Hawaii, USA, pp. 326–337.

Dixit,K. and Al-Onaizan,Y. (2019) Span-level model for relation extraction. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy, pp. 5308–5314.

Dozat,T. and Manning,C.D. (2016) Deep biaffine attention for neural dependency parsing. *arXiv Preprint arXiv: 1611.01734*.

Duvenaud,D.K. *et al.* (2015) Convolutional networks on graphs for learning molecular fingerprints. In: *Advances in Neural Information Processing Systems*. Montreal, Quebec, Canada, pp. 2224–2232.

Fei,H. *et al.* (2020a) Boundaries and edges rethinking: an end-to-end neural model for overlapping entity relation extraction. *Inf. Process. Manag.*, **57**, 102311.

Fei,H. *et al.* (2020b) Enriching contextualized language model from knowledge graph for biomedical information extraction. *Brief. Bioinform*.

Fei,H. *et al.* (2020c) High-order refining for end-to-end Chinese semantic role labeling. *arXiv Preprint arXiv: 2009.06957*.

Fundel,K. *et al.* (2007) Relex–relation extraction using dependency parse trees. *Bioinformatics*, **23**, 365–371.

Gurulingappa,H. *et al.* (2012) Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *J. Biomed. Inform.*, **45**, 885–892.

He,L. *et al.* (2018) Jointly predicting predicates and arguments in neural semantic role labeling. *arXiv Preprint arXiv: 1805.04787*.

Hsieh,Y.-L. *et al.* (2017) Identifying protein-protein interactions in biomedical literature using recurrent neural networks with long short-term memory. In: *Proceedings of the 8th International Joint Conference on Natural Language Processing*. Taipei, Taiwan, pp. 240–245.

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 the Association for Computational Linguistics*. Vancouver, Canada. pp. 917–928.

Kim,Y. (2014) Convolutional neural networks for sentence classification. *arXiv Preprint arXiv: 1408.5882*.

Kingma,D.P. and Ba,J. (2014) Adam: a method for stochastic optimization. *arXiv Preprint arXiv: 1412.6980*,

Lample,G. *et al.* (2016) Neural architectures for named entity recognition. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*. San Diego, CA, USA, pp. 260–270.

Lee,K. *et al.* (2018) Higher-order coreference resolution with coarse-to-fine inference. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. New Orleans, Louisiana, USA, pp. 687–692.

Li,F. *et al.* (2017) A neural joint model for entity and relation extraction from biomedical text. *BMC Bioinformatics*, **18**, 198–208.

Li,Z. *et al.* (2019) Teaching machines to extract main content for machine reading comprehension. In: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*. Honolulu, Hawaii, USA, pp. 9973–9974.

Liu,S. *et al.* (2016a) Dependency-based convolutional neural network for drug-drug interaction extraction. In: *Proceedings of 2016 IEEE International Conference on Bioinformatics and Biomedicine*. pp. 1074–1080.

Liu,S. *et al.* (2016b) Drug-drug interaction extraction via convolutional neural networks. *Comput. Math. Methods Med.*, Shenzhen, China, **2016**, 1–8.

Luan,Y. *et al.* (2019) A general framework for information extraction using dynamic span graphs. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Minneapolis, USA, pp. 3036–3046.

Marcheggiani,D. and Titov,I. (2017) Encoding sentences with graph convolutional networks for semantic role labeling. *arXiv Preprint arXiv: 1703.04826*.

Marcus,M.P. *et al.* (1993) Building a large annotated corpus of English: the penn treebank. *Comput. Linguist.*, **19**, 313–330.

Miwa,M. and Bansal,M. (2016) End-to-end relation extraction using LSTMs on sequences and tree structures. In: *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*. Berlin, Germany, pp. 1105–1116.

Nédellec,C. (2005) Learning language in logic-genic interaction extraction challenge. In: *Proceedings of the 4th Learning Language in Logic Workshop*. Lisboa, Portugal, pp. 1–7.

Peng,Y. *et al.* (2015) An extended dependency graph for relation extraction in biomedical texts. In: *Proceedings of BioNLP Shared Task 2015 Workshop*. Berlin, Germany, pp. 21–30.

Pyysalo,S. *et al.* (2007) BioInfer: a corpus for information extraction in the biomedical domain. *BMC Bioinformatics*, **8**, 50.

Ren,X. *et al.* (2017) Cotype: joint extraction of typed entities and relations with knowledge bases. In: *Proceedings of the 26th International Conference on World Wide Web*. Perth, Australia, pp. 1015–1024.

Ren,Y. *et al.* (2018) Neural networks for bacterial named entity recognition. In: *2018 IEEE International Conference on Bioinformatics and Biomedicine*. Madrid, Spain, pp. 2797–2799.

Sahu,S.K. and Anand,A. (2018) Drug-drug interaction extraction from biomedical texts using long short-term memory network. *J. Biomed. Inform.*, **86**, 15–24.

Segura Bedmar,I. *et al.* (2013) SemEval-2013 task 9: extraction of drug-drug interactions from biomedical texts. In: *Proceedings of the Seventh International Workshop on Semantic Evaluation*. Atlanta, Georgia, pp. 341–350.

Sun,X. *et al.* (2018) Deep convolution neural networks for drug-drug interaction extraction. In: *Proceedings of 2018 IEEE International Conference on Bioinformatics and Biomedicine*. Madrid, Spain, pp. 1662–1668.

Takanobu,R. *et al.* (2018) A hierarchical framework for relation extraction with reinforcement learning. *arXiv Preprint arXiv: 1811.03925*,

Vaswani,A. *et al.* (2016) Supertagging with LSTMs. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*. San Diego, CA, USA, pp. 232–237.

Wang,S. *et al.* (2018) Joint extraction of entities and relations based on a novel graph scheme. In: *Proceedings of 27th International Joint Conference on Artificial Intelligence*. Stockholm, Sweden, pp. 4461–4467.

Yi,Z. *et al.* (2017) Drug-drug interaction extraction via recurrent neural network with multiple attention layers. In: *Proceedings of International Conference on Advanced Data Mining and Applications*. Singapore, pp. 554–566.

Zeng,X. *et al.* (2018) Extracting relational facts by an end-to-end neural model with copy mechanism. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. Melbourne, Australia, pp. 506–514.

Zhang,M. *et al.* (2017) End-to-end neural relation extraction with global optimization. In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Copenhagen, Denmark, pp. 1730–1740.

Zhang,Y. *et al.* (2018) A hybrid model based on neural networks for biomedical relation extraction. *J. Biomed. Inform.*, **81**, 83–92.

Zheng,S. *et al.* (2017a) Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing*, **257**, 59–66.

Zheng,S. *et al.* (2017b) Joint extraction of entities and relations based on a novel tagging scheme. *arXiv Preprint arXiv: 1706.05075*.