A Relation-Specific Attention Network for Joint Entity and Relation Extraction

Yue Yuan[1,2] , **Xiaofei Zhou**[1,2*] , **Shirui Pan**[3] , **Qiannan Zhu**[1,2] , **Zeliang Song**[1,2] and **Li Guo**[1,2]

[1]Institute of Information Engineering, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences, School of Cyber Security
[3]Faculty of Information Technology, Monash University
{yuanyue,zhouxiaofei}@iie.ac.cn, shirui.pan@monash.edu

Abstract

Joint extraction of entities and relations is an important task in natural language processing (NLP), which aims to capture all relational triplets from plain texts. This is a big challenge due to some of the triplets extracted from one sentence may have overlapping entities. Most existing methods perform entity recognition followed by relation detection between every possible entity pairs, which usually suffers from numerous redundant operations. In this paper, we propose a relation-specific attention network (RSAN) to handle the issue. Our RSAN utilizes relation-aware attention mechanism to construct specific sentence representations for each relation, and then performs sequence labeling to extract its corresponding head and tail entities. Experiments on two public datasets show that our model can effectively extract overlapping triplets and achieve state-of-the-art performance. Our code is available at https://github.com/Anery/RSAN

1 Introduction

Jointly extracting entities and relations is to capture structural knowledge in the form of *(head, relation, tail)* from unstructured texts. The process can promote many graph-based tasks in data mining and NLP fields, such as knowledge graph construction [Luan *et al.*, 2018] and graphical dialogue system [Liu *et al.*, 2018]. Traditional pipelined extraction systems [Zelenko *et al.*, 2003; Chan and Roth, 2011] treat entity and relation extractions as two separate tasks, which perform relation classification after the recognition of all the entities in the text. Such models suffer from error propagation and ignore the relevance between the two subtasks [Li and Ji, 2014]. Thus, many researchers focus on building joint models to simultaneously extract entities and relations.

Prior joint learning methods [Kate and Mooney, 2010; Miwa and Sasaki, 2014] depend heavily on complex feature engineering and other off-the-shelf NLP tools. The later studies concentrate more on learning neural network-based models, and some of them apply parameter sharing strategy for jointly training [Miwa and Bansal, 2016; Katiyar and Cardie, 2017]. Although these neural methods perform better than the former, they still make predictions separately on extracting entities and relations, and the connections between the two subtasks are not fully utilized. Recently, a NovelTagging model [Zheng *et al.*, 2017] combines the two tasks as a single sequence labeling problem. However, a word cannot be assigned with more than one tag, so that the model fails to extract the triplets with overlapped entities (see the examples in Figure 1).

	Texts	Triplets
Normal	The [United States] president [Donald Trump] will visit [Beijing], [China].	(Donald Trump, *President_of*, United States) (China, *Contains*, Beijing)
SEO	The [United States] president [Donald Trump] was born in [New York City].	(**Donald Trump**, *President_of*, United States) (**Donald Trump**, *Born_in*, New York City)
EPO	Martin went to [Tokyo] last week, which is the capital of [Japan].	(**Japan**, *Contains*, **Tokyo**) (**Japan**, *Capital*, **Tokyo**)

Figure 1: Examples of the *Normal*, *SingleEntityOverlap (SEO)* and *EntityPairOverlap (EPO)* cases. The overlapping entities are marked in bold. The first example belongs to *Normal* class which has no overlapped entities. The second one with triplets sharing one single entity *Donald Trump* belongs to *SEO* class. The last one that has triplets with overlapped entity pair *(Japan, Tokyo)* belongs to *EPO* class.

To address the overlapping issue, many entity-guided joint learning methods, such as PA-LSTM [Dai *et al.*, 2019] and ETL-Span [Yu *et al.*, 2020] are proposed. They perform head entities recognition as the first step, and develop some joint decoding strategies for extracting the corresponding tail entities and relations. On the contrary, CopyRE [Zeng *et al.*, 2018] and HRL [Takanobu *et al.*, 2019] present a relation-guided joint extraction process, which takes relation classification as the first step of their models. It is because relations are usually triggered by the context of sentences rather than target entities. For example, descriptions like 'was born in' in the sentence will directly lead to the `place_of_birth` relation. Thus, the relation information can be first introduced as prior knowledge and reduce the model's focus on semantically unrelated entities, which avoids the redundant extraction operations on them. However, CopyRE and HRL simply utilize the results of relation classification as the guidance of entity extraction, ignoring the fine-grained semantic

*Corresponding Author

connections between relations and the words in the sentence. We argue that the words should have different contributions to the underlying semantic expression of the sentence under different relations. Based on this assumption, we use attention mechanism for assigning higher weights to the relation-related words in the sentence.

In this paper, we propose a relation-specific attention network (RSAN) for joint entity and relation extraction. We use relation-based attention to construct the specific sentence representation under each relation, and then perform sequence labeling to extract its corresponding entities. Our model can not only capture fine-grained semantic features from the words, but also effectively solve the overlapping problem by decomposing the extraction task into separate entity tagging processes for different relations. Moreover, we employ a relational gate to reduce the noise brought by the unrelated relations in entity recognition. During training, we further use a relation-level negative sampling strategy to avoid most of the redundant decoding processes. In summary, the main contribution of this paper are as follows:

- We present a joint entity and relation extraction model named RSAN, which incorporates the relation fine-grained semantic information to guide the entity recognition process. Our RSAN is suitable for extracting the overlapping triplets as it performs entity extraction for different relations separately.
- We apply relation-based attention mechanism to construct different sentence representations under different relations, and propose a relational gated mechanism to adaptively control the relation information provided for entity decoding.
- Training with a relational negative sampling strategy, our model achieves state-of-the-art results on two public datasets, which proves its effectiveness.

2 Related Work

Researchers have made great efforts in relational facts extraction, which can be directly used for knowledge graph construction, or supporting downstream text mining tasks. Early methods [Zelenko *et al.*, 2003; Chan and Roth, 2011] regard entity and relation extractions as two separate subtasks, which apply pipelined approach to perform relation classification after extracting all the entities. To construct the bridge between the two subtasks, building joint model that extracts entities together with relations simultaneously has attracted much attention. The prior feature-based models [Kate and Mooney, 2010; Miwa and Sasaki, 2014] rely on other NLP tools to do feature engineering, and suffer from the error propagation. The later works are mainly based on neural architectures, which can be roughly divided into Table-filling, Tagging and Seq2Seq methods.

Table-filling methods [Miwa and Sasaki, 2014; Gupta *et al.*, 2016] construct a table for each sentence and specify an order of the table cells, incrementally filling the table with entity or relation tags. The recent work GraphRel [Fu *et al.*, 2019] can also be seen as table-filling methods, which applies 2-phrase Graph Convolutional Network (GCN) to predict word entities and relations for each word pair. Due to the sparsity of the tables, these methods suffer from plenty of redundant predictions. Tagging methods usually use well-designed tagging strategies to construct connections between entities and relations. Among these methods, NovelTagging [Zheng *et al.*, 2017] first treats entity types and relation roles as different parts of the tag, and models the joint extraction task as a single sequence labeling problem. However, it cannot handle the overlapping cases. As the improvement, [Takanobu *et al.*, 2019; Dai *et al.*, 2019; Yu *et al.*, 2020] perform the tagging process for multiple turns. The Seq2Seq methods try to directly generate all triplets sequentially. CopyRE [Zeng *et al.*, 2018] generates the relation followed by its two corresponding entities with copy mechanism, but only the last word of the entity can be generated. Therefore, CopyMTL [Zeng *et al.*, 2019a] applies a multi-task learning framework to extract multi-token entities. WDec [Nayak and Ng, 2020] designs a new representation scheme for the triplets, and then generates triplets as word sequences. OrderRL [Zeng *et al.*, 2019b] applies reinforcement learning to optimize the extraction order of triplets.

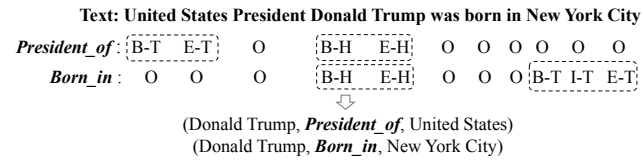

Figure 2: An example for our relation-specific tagging scheme. For different given relations, we will generate a specific tag sequence for each of them.

In this paper, we propose a new tagging method named RSAN to decode the specific entities for each relation, which is helpful for dealing with overlapping problem. Different from the above mentioned models, we use attention mechanism to incorporate fine-grained relation information as the guidance of the entity tagging process.

3 Problem Formulation

We describe the relational triplets as $\{\pi = (h, r, t) \mid h, t \in E, r \in R\}$ and a sentence as $S = \{w_1, w_2, ..., w_n\}$, where E and R are the entity and relation sets respectively, a triplet π indicates entity pair (h, t) and relation r between them, and w_i is the i-th word in the sentence. In this paper, given a sentence S and a predefined relation set R, the purpose of the joint entity and relation extraction task is to identify all existing triplets π from S.

Note that the extracted triplets may share the same entities or relations, i.e. the overlapping problem. Thus designing a joint extraction model to overcome such issue is a big challenge in this task.

4 Methodology

In this section, we will first introduce our tagging scheme which transforms overlapping triplets extraction task to several sequence labeling problems. Then we elaborate the details of our relation-specific attention network based on a certain relation.

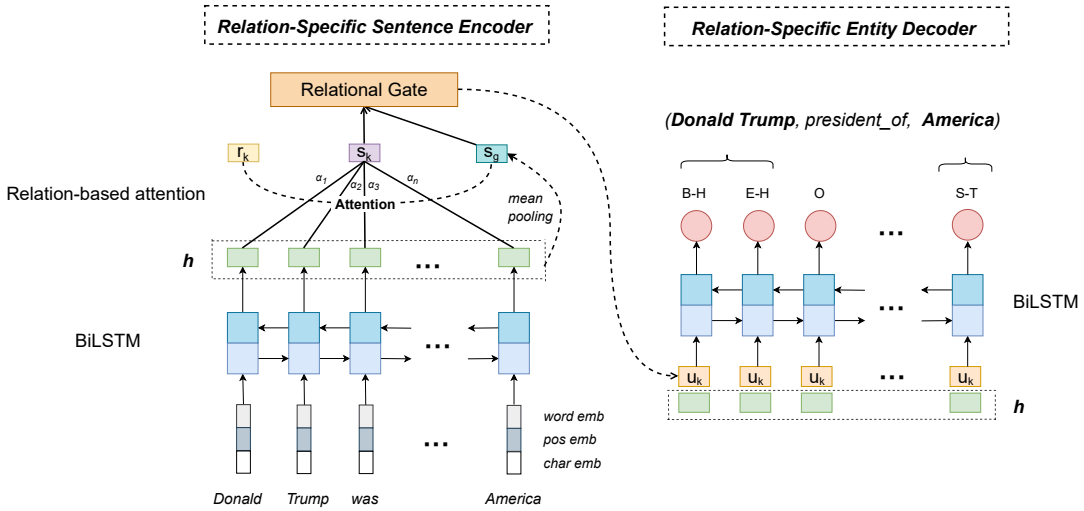

Figure 3: The overall structure of RSAN under the given relation r_k (*president_of* in this example). The left part is relation-specific sentence encoder, and the right is entity decoder. The sentence encoder incorporates relation-based attention mechanism and gated mechanism on bi-directional LSTM to generate relation-guided sentence representation. Based on the sentence representation, the entity decoder aims to extract the corresponding entities of r_k. Here, the entity decoder extracts the head entity *Donald Trump* and the tail entity *America*, RSAN will combine this entity pair with r_k to return the triplets *(Donald Trump, president_of, America)*.

4.1 Tagging Scheme

We incorporate head and tail roles {H,T} in the triplets into the typical BIES signs (Begin, Inside, End, Single) as our entity tags. For a sentence with multiple triplets, we will generate separate tag sequences according to different relations. In the tag sequence of a certain relation, only its corresponding head and tail entities will be annotated, while the rest of words are assigned with label O. Figure 2 shows an example of our extracting method. There are two triplets in the sentence: *(Donald Trump, President_of, United States)* and *(Donald Trump, Born_in, New York City)*, we will perform sequence labeling for the relation *President_of* and *Born_in* separately. As we can see, the two triplets have the overlapped entity *Donald Trump*, and they can be extracted without conflict based on the separate labeling operations.

Besides, when multiple triplets share the same relation, i.e., the relation overlapping cases, we follow [Zheng *et al.*, 2017] and apply the heuristic nearest principle to combine the entity pairs. Concretely, the nearest head and tail entities will be combined into a triplet.

4.2 Relation-Specific Attention Network

Figure 3 gives an overview of RSAN under a certain relation r_k. Note that the extracted entities will be directly combined with the current relation r_k, thus there is no extra relation classification operations in our model. We first encode the input sentence with a bi-directional Long Short Term Memory (BiLSTM) network [Hochreiter and Schmidhuber, 1997], and then apply attention mechanism to construct the specific sentence representation of r_k. After filtering by relational gate, the final representation of the sentence will be used for the sequence labeling process to extract the corresponding entities.

BiLSTM Layer

Given a sentence $S = \{w_1, w_2, ..., w_n\}$ of length n, we denote $\mathbf{x}_i = [\mathbf{w}_i^w; \mathbf{w}_i^p; \mathbf{w}_i^c]$ as the representation of the i-th word, where $\mathbf{w}_i^w \in \mathbb{R}^{d_w}$ is randomly initialized word embedding, $\mathbf{w}_i^p \in \mathbb{R}^{d_{pos}}$ is the part-of-speech (POS) embedding, and $\mathbf{w}_i^c \in \mathbb{R}^{d_c}$ is character-based word features. The character-level word features are extracted by a convolution neural network (CNN) running on the character sequence of w_i. Then we choose BiLSTM to capture the dependencies of the words. The sequence of word representations $\{\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n\}$ are taken as the input of BiLSTM network. We concatenate the forward and backward LSTM hidden states of $\mathbf{x}_i$ as the contextual word representation:

$$\mathbf{h}_i = [\overrightarrow{\mathrm{LSTM}}(\mathbf{x}_i); \overleftarrow{\mathrm{LSTM}}(\mathbf{x}_i)], i \in [1, n]$$

where $\mathbf{h}_i \in \mathbb{R}^{2\times d_{he}}$, and d_{he} indicates the dimension of the BiLSTM hidden state. Then we use $S_c = \{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n\}$ to represent the context-level sentence features.

Relation-Based Attention Mechanism

Based on our assumption, the words in the sentence play different roles under different relations. To this end, we propose a relation-based attention mechanism for assigning different weights to the context words under each relation. The attention score is obtained as follows:

$$\mathbf{s}_g = \mathrm{avg}\{\mathbf{h}_1, \mathbf{h}_2, ..., \mathbf{h}_n\}, \quad (1)$$

$$\mathbf{e}_{ik} = \mathbf{v}^T \tanh(\mathbf{W}_r \mathbf{r}_k + \mathbf{W}_g \mathbf{s}_g + \mathbf{W}_h \mathbf{h}_i), \quad (2)$$

$$\alpha_{ik} = \frac{\exp(\mathbf{e}_{ik})}{\sum_{j=1}^{n} \exp(\mathbf{e}_{jk})}, \quad (3)$$

where $\mathbf{r}_k \in \mathbb{R}^{d_r}$ is the trainable embedding of the k-th relation, and $\mathbf{v} \in \mathbb{R}^{d_{att}}$, $\mathbf{W}_r \in \mathbb{R}^{d_{att}\times d_r}$, $\mathbf{W}_g$, $\mathbf{W}_h \in \mathbb{R}^{d_{att}\times 2d_{he}}$ are trainable parameters. Here $\mathbf{s}_g$ indicates the

global representation of the sentence. In this way, the attention score can not only measure the importance of each word to the relational expression, but also its contribution to the entire sentence. The specific sentence representation $\mathbf{s}_k$ under relation type r_k is then generated by weighted sum of the sentence words,

$$\mathbf{s}_k = \sum_{i=1}^{n} \alpha_{ik} \mathbf{h}_i. \tag{4}$$

Relational Gated Mechanism

So far we have obtained sentence representations fused with relation information. As we argued before, the relation-oriented representations make sense to the followed entity extraction only when the relation is positive to the sentence, while that of the unrelated relations will only confuse the subsequent decoding process. In order to adaptively control the relation information provided by the previous attention layer, we propose a gated mechanism as the bridge. Still taking the k-th relation as an example, the gated operations are defined as follows:

$$g_k = \sigma((\mathbf{W}_1 \mathbf{s}_g + b_1) \oplus (\mathbf{W}_2 \mathbf{s}_k + b_2)), \tag{5}$$

$$\mathbf{u_k} = g_k \odot \tanh(\mathbf{W}_3 \mathbf{s}_k + b_3), \tag{6}$$

where $\mathbf{W}_1$, $\mathbf{W}_2$, $\mathbf{W}_3 \in \mathbb{R}^{d_g \times 2d_{he}}$, b_1, b_2, $b_3 \in \mathbb{R}^{d_g}$ are parameters, $\oplus$ is concatenating operation, and $\odot$ is dot product. σ indicates the element-wise sigmoid activation function, which returns values from 0 to 1, therefore the results can be viewed as percentage of information to keep. The purpose of Eq. 5 is to measure whether the inherent sentence representation $\mathbf{s}_g$ or the relation-based representation $\mathbf{s}_k$ is more useful for the entity extraction. $\mathbf{u_k}$ is the reserved relational features. We concatenate $\mathbf{h}_i$ and $\mathbf{u_k}$ to obtain the final representation of the i-th word.

$$\mathbf{h}_i^k = \mathbf{h}_i \oplus \mathbf{u_k}, \tag{7}$$

here $\mathbf{h}_i^k \in \mathbb{R}^{2d_{he}+d_g}$. Sentence S is thus represented as $S^k = \{\mathbf{h}_1^k, \mathbf{h}_2^k, ..., \mathbf{h}_n^k\}$, and will be used for the entity extraction process.

Relation-Specific Entity Decoder

We perform a relation-specific sequence labeling process as the entity decoder. Here we run another BiLSTM network on the word sequence S^k, and map each of the word to the tag space:

$$\mathbf{o}_i^k = [\overrightarrow{\mathrm{LSTM}}(\mathbf{h}_i^k); \overleftarrow{\mathrm{LSTM}}(\mathbf{h}_i^k)], \tag{8}$$

$$P(y_i^k) = \mathrm{Softmax}(\mathbf{W}_\mathrm{o} \cdot \mathbf{o}_\mathrm{i}^\mathrm{k} + \mathbf{b}_\mathrm{o}), \tag{9}$$

where $i \in [1, n]$. $\mathbf{W}_o \in \mathbb{R}^{2d_{hd} \times n_l}$, $\mathbf{b}_o \in \mathbb{R}^{d_{n_l}}$ are parameters, and d_{hd} is the dimension of hidden state of BiLSTM, n_l is the total number of tags. $P(y_i^k)$ indicates the probability of i-th word's predicted tag under relation r_k.

Training

Notice that the number of relations present in a sentence is much smaller compared to the size of R. If we perform entity decoding for all given relations during training, there will be a large amount of negative samples, which makes it difficult for convergency. Therefore, we apply a relational negative sampling strategy, i.e., randomly select n_{neg} relations from the negative set of the current sentence. Here n_{neg} is a hyper-parameter. All of the words will be labeled with tag `O` since there are no triplets based on those negative relations. Then for a sentence S with n_{sp} positive relations, our model will totally generate $n_s = n_{sp} + n_{neg}$ tag sequences while decoding. We use negative log-likelihood (NLL) loss function to train our model. We denote the ground truth labels under relation r_k as $\{\hat{y}_1^k, \hat{y}_2^k, ..., \hat{y}_n^k\}$, then the NLL loss can be defined as:

$$\mathcal{L} = \frac{1}{n_s \times n} \sum_{k=1}^{n_s} \sum_{i=1}^{n} -\log P(y_i^k = \hat{y}_i^k).$$

DataSet	NYT	WebNLG
Relation types	24	246
Tain sentences	56195	5019
Dev sentences	5000	500
Test sentences	5000	703

Table 1: Statistics of the datasets.

5 Experiments

5.1 DataSets

Following [Zeng *et al.*, 2018], we evaluate our model on two widely used datasets: New York Times (NYT) and WebNLG. NYT is first constructed using distant supervised method, which automatically aligns knowledge base and plain texts to generate large-scale training data. WebNLG is created by [Gardent *et al.*, 2017] for Natural Language Generation (NLG) task, and all of the standard sentences are written by annotators. To be consistent with all other baselines, we only select the first standard sentence in each instance to reconstruct the corpus. Statistics of the two datasets are shown in Table 1.

5.2 Implementation Details

We set the dimension of word embedding $d_w = 100$, POS embedding $d_{pos} = 15$, character embedding $d_c = 50$, and relation embedding $d_r = 300$. All of those embeddings are randomly initialized. The window size of CNN for character-based word feature vector is set to 3, the maximum length of words is set to 10, and the number of filters is 50 ($d_f = 50$). Hidden State of the encoder BiLSTM (d_{he}), attention (d_{att}), gate (d_g) and the decoder BiLSTM (d_{hd}) are all set to 300 dimensions. The sentence-level relational negative sampled number n_{neg} is set to 4. The model is trained using Adam [Kingma and Ba, 2014] with learning rate of 0.001 and batch size of 16. We apply dropout mechanism to the embedding layer with a rate of 0.5 to avoid overfitting.

5.3 Baselines and Evaluation Metrics

We compare our model with the following baselines:

- **NovelTagging** [Zheng *et al.*, 2017] applies a novel tagging strategy that incorporates both entity types and relation roles, and converts the joint extraction task to a sequence labeling problem. This model cannot extract triplets with overlapping entities.

Model	NYT			WebNLG		
	Prec	Rec	F1	Prec	Rec	F1
Novel Tagging [Zheng *et al.*, 2017]	0.624	0.371	0.420	0.525	0.193	0.283
CopyRE [Zeng *et al.*, 2018]	0.610	0.566	0.587	0.377	0.364	0.371
GraphRel [Fu *et al.*, 2019]	0.639	0.60	0.619	0.447	0.411	0.429
CopyMTL [Zeng *et al.*, 2019a]	0.757	0.687	0.720	0.580	0.549	0.564
OrderRL [Zeng *et al.*, 2019b]	0.779	0.672	0.721	0.663	0.599	0.616
HRL [Takanobu *et al.*, 2019]	0.781	0.771	0.776	-	-	-
ETL-Span [Yu *et al.*, 2020]	0.841	0.746	0.791	0.691	0.695	0.693
WDec [Nayak and Ng, 2020]	**0.881**	0.761	0.817	**0.848**	0.649	0.735
RSAN	0.857	**0.836**	**0.846**	0.805	**0.838**	**0.821**

Table 2: Main results of the compared models on NYT and WebNLG.

- **CopyRE** [Zeng *et al.*, 2018] first explores Seq2Seq model for the joint entity and relation extraction task, and generates the triplets in the sentence sequentially using copy mechanism. This model can only copy the last word of an entity.
- **GraphRel** [Fu *et al.*, 2019] constructs a complete word graph for each sentence, and employs GCN to predict relations between all word pairs.
- **CopyMTL** [Zeng *et al.*, 2019a] improves the copy strategy of CopyRE, and applies a multi-task learning framework to solve the problem of generating multi-token entities.
- **OrderRL** [Zeng *et al.*, 2019b] incorporates reinforcement learning into Seq2Seq model to learn the extraction order of triplets.
- **HRL** [Takanobu *et al.*, 2019] applies a hierarchical paradigm which performs relation detection first as a high-level reinforcement learning process, then identifies entities as a low-level one.
- **ETL-Span** [Yu *et al.*, 2020] applies a novel decomposition strategy, which first distinguishes all head entities, and then identifies corresponding tail entities and relations.
- **WDec** [Nayak and Ng, 2020] proposes a novel triplets representation scheme and employs Seq2Seq to generate the word sequences.

We use standard Precision (Prec), Recall (Rec) and F1 score as our evaluation metrics. A triplet is considered to be correctly extracted if and only if its relation type and two entities are exactly matched.

5.4 Results

Table 2 shows all of the comparison results. Overall, our RSAN outperforms all other baselines. We attribute the gains of RSAN to its two advantages: (1) RSAN focuses more on the relation-related entities, which excludes the error caused by predictions on the redundant entity pairs; (2) The relation-attentive entity tagging process has the ability to capture the dependencies between the extraction of entities and relations.

In addition, our RSAN also achieves higher performance among the relation-guided methods, like CopyRE [Zeng *et al.*, 2018], OrderRL [Zeng *et al.*, 2019b] and HRL [Takanobu *et al.*, 2019]. We consider that is because our attention mechanism incorporates fine-grained relation information, which enables more explicit guidance on the entity extraction process.

Model	Precision	Recall	F1
RSAN	**0.857**	**0.836**	**0.846**
–POS embedding	0.846	0.821	0.833
–Character embedding	0.850	0.827	0.838
–Relation-based Attention	0.794	0.835	0.813
–Relational Gate	0.825	0.832	0.828

Table 3: Ablation study of RSAN on NYT dataset.

6 Analysis

6.1 Ablation Study

We conduct ablation experiments to demonstrate the effectiveness of POS embedding, character-level word embedding, relation-based attention mechanism and the relational gate in our model. We remove one component at a time to observe its impact on the experimental results, which is summarized in Table 3. (1) POS embeddings in the input layer effectively provide additional syntactic information to the sentence. (2) The character-level embeddings are helpful to provide prior knowledge for OOV words. (3) In order to verify the usage of the relation-based attention mechanism, we no longer construct the relation-attentive sentence representation $\mathbf{s}_k$ (Eq. 4), and replace the $\mathbf{s}_k$ in Eq. 5 and 6 with relation embedding $\mathbf{r}_k$. That is to say, we try to directly use the relation embedding as the guidance of entity extraction. As shown in the results, the model's precision drops significantly. We consider that using relation embedding simply learns the shallow co-occurrence of the triplets, resulting in more triplet predictions but lower precision of the model. On the contrary, our attention mechanism can capture fine-grained semantic relation features in the sentence, which lead to a more significant distinction between positive and negative relations. (4) For the relational gate component, we omit the operations of Eq. 5 and 6. As an alternative to $\mathbf{u}_k$ in Eq. 7, we explicitly use the sentence representation $\mathbf{s}_k$, ignoring the possible impact of negative relations. We found a decrease in the result, which indicates that our relational gated mechanism has contributed to reducing the noise brought by unrelated relations.

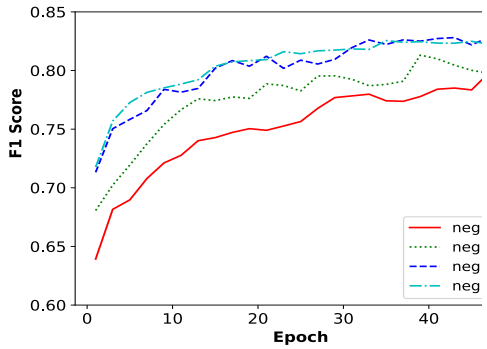

Figure 4: Training curves of F1 score in NYT dataset with n_{neg}.

6.2 Parameter Analysis

In each training iteration, we randomly sample n_{neg} n relations for the sentence, which aims to balance the r convergence speed and generalization performance. I vious that constructing more negative samples can i the robustness of the model, but we don't have to higher value of n_{neg} to achieve this. Actually, with the appropriate setting of random sampling count, almost all negative relations of the sentence will be covered as the increasing iterations. Thus, there should be an upper bound for hyperparameter n_{neg}, leading to no longer improvement on the model's performance when it is larger than the bound value.

There are 24 relation types in the NYT dataset, with an average of 1.44 positive ones per sentence. Therefore, we try to select n_{neg} among $\{1,2,4,6\}$, which is an appropriate range based on the number of positive relations in average. Figure 4 shows the curves of F1 score on validation set varying with training epochs under different values of n_{neg}. It can be observed that when n_{neg} = 4 or 6, there is almost no difference in convergence and prediction performance. Thus for NYT dataset, we consider n_{neg} = 4 as the upper bound, which can ensure the effectiveness of the model and speed up training process at the same time.

6.3 Analysis on Overlapping Cases

To verify the capability of our RSAN in extracting multiple triplets, we follow [Zeng *et al.*, 2018], and conduct further experiments on NYT dataset. The test sentences are divided into three categories based on different overlapping cases, i.e., *Normal*, *SingleEntityOverlap (SEO)*, and *EntityPairOverlap (EPO)* (See the examples in Figure 1). We then verify several latest models' performance on each of the category. The results are shown in Figure 5. As we can see, RSAN outperforms all other methods in the overlapping situations, especially for the EPO class. We attribute the improvements to the fact that the entity pair overlapped triplets only have different relations, hence our separate prediction on each of the relation can effectively handle such cases. Another observation is that ETL-Span achieves the best performance in *Normal* class. It is because its decomposition strategy is designed more suitable for the *Normal* cases, while our RSAN performs much better in the overlapping classes.

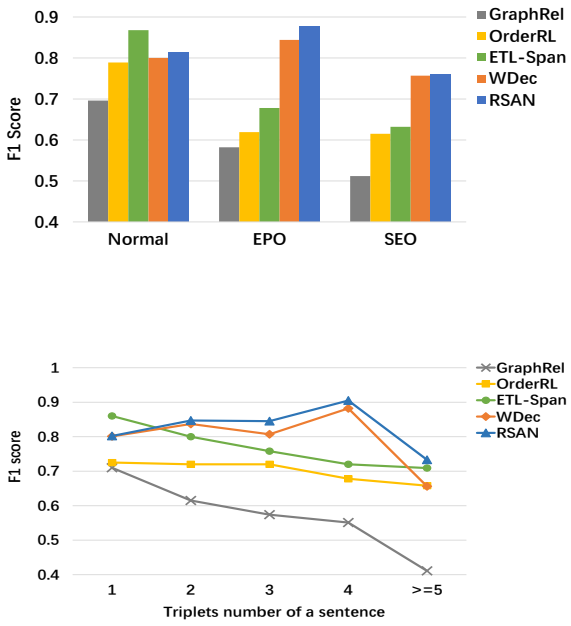

Figure 6: Relation extraction on sentences with different number of triplets.

We also compare the models' ability of extracting multiple triplets in a sentence. We divide the sentences of NYT test set into 5 categories, which respectively indicate its number of triplets is 1,2,3,4 and ≥ 5. The results are shown in Figure 6. It can be observed that our RSAN gains great improvements compared with other models in extracting multiple triplets. Besides, RSAN shows more stable performance with the increasing of triplets numbers in the sentence. These two additional experiments fully demonstrate the advantages of our proposed model in dealing with complex extracting situation.

7 Conclusion

In this paper, we propose a relation-attentive sequence labeling framework named RSAN for joint entity and relation extraction task. It decomposes the overlapping triplets extraction problem into several relation-specific entity tagging processes, and applies attention mechanism to incorporate fine-grained relational information as the guidance of entity extraction. Experiments on the NYT and WebNLG corpus show that our proposed model RSAN has achieved significant improvement. The extended experiments demonstrate the effectiveness of RSAN in handling overlapping and multiple triplets extraction scenarios.

Acknowledgments

This work is supported by National Key R&D Program 2016 (No.2016YFB0801300), and the National Natural Science Foundation of China (No.61202226). We thank all anonymous reviewers for their constructive comments.

References

[Chan and Roth, 2011] Yee Seng Chan and Dan Roth. Exploiting syntactico-semantic structures for relation extrac-

tion. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 551–560. Association for Computational Linguistics, 2011.

[Dai *et al.*, 2019] Dai Dai, Xinyan Xiao, Yajuan Lyu, Shan Dou, Qiaoqiao She, and Haifeng Wang. Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 6300–6308, 2019.

[Fu *et al.*, 2019] Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. Graphrel: Modeling text as relational graphs for joint entity and relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1409–1418, 2019.

[Gardent *et al.*, 2017] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. Creating training corpora for nlg micro-planning. 2017.

[Gupta *et al.*, 2016] Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 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*, pages 2537–2547, 2016.

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

[Kate and Mooney, 2010] Rohit J Kate and Raymond J Mooney. Joint entity and relation extraction using card-pyramid parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 203–212. Association for Computational Linguistics, 2010.

[Katiyar and Cardie, 2017] Arzoo Katiyar and Claire Cardie. 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 (Volume 1: Long Papers)*, pages 917–928, 2017.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[Li and Ji, 2014] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, 2014.

[Liu *et al.*, 2018] Shuman Liu, Hongshen Chen, Zhaochun Ren, Yang Feng, Qun Liu, and Dawei Yin. Knowledge diffusion for neural dialogue generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1498, 2018.

[Luan *et al.*, 2018] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *arXiv preprint arXiv:1808.09602*, 2018.

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

[Miwa and Sasaki, 2014] Makoto Miwa and Yutaka Sasaki. Modeling joint entity and relation extraction with table representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1858–1869, 2014.

[Nayak and Ng, 2020] Tapas Nayak and Hwee Tou Ng. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In *Proceedings of The Thirty-Fourth AAAI Conference on Artificial Intelligence (AAAI)*, 2020.

[Takanobu *et al.*, 2019] Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. A hierarchical framework for relation extraction with reinforcement learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7072–7079, 2019.

[Yu *et al.*, 2020] Bowen Yu, Zhenyu Zhang, Xiaobo Shu, Yubin Wang, Tingwen Liu, Bin Wang, and Sujian Li. Joint extraction of entities and relations based on a novel decomposition strategy. In *Proc. of ECAI*, 2020.

[Zelenko *et al.*, 2003] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106, 2003.

[Zeng *et al.*, 2018] Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, Jun Zhao, et al. Extracting relational facts by an end-to-end neural model with copy mechanism. 2018.

[Zeng *et al.*, 2019a] Daojian Zeng, Haoran Zhang, and Qianying Liu. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. *CoRR*, abs/1911.10438, 2019.

[Zeng *et al.*, 2019b] Xiangrong Zeng, Shizhu He, Daojian Zeng, Kang Liu, Shengping Liu, and Jun Zhao. Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 367–377, 2019.

[Zheng *et al.*, 2017] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. Joint extraction of entities and relations based on a novel tagging scheme. *arXiv preprint arXiv:1706.05075*, 2017.