A Simple but Effective Bidirectional Framework for Relational Triple Extraction

Feiliang Ren[*] [†]
Northeastern University
Shenyang, China

Longhui Zhang[†]
Northeastern University
Shenyang, China

Xiaofeng Zhao
Northeastern University
Shenyang, China

Shujuan Yin
Northeastern University
Shenyang, China

Shilei Liu
Northeastern University
Shenyang, China

Bochao Li
Northeastern University
Shenyang, China

ABSTRACT

Tagging based relational triple extraction methods are attracting growing research attention recently. However, most of these methods take a unidirectional extraction framework that first extracts all subjects and then extracts objects and relations simultaneously based on the subjects extracted. This framework has an obvious deficiency that it is too sensitive to the extraction results of subjects. To overcome this deficiency, we propose a bidirectional extraction framework based method that extracts triples based on the entity pairs extracted from two complementary directions. Concretely, we first extract all possible subject-object pairs from two paralleled directions. These two extraction directions are connected by a shared encoder component, thus the extraction features from one direction can flow to another direction and vice versa. By this way, the extractions of two directions can boost and complement each other. Next, we assign all possible relations for each entity pair by a biaffine model. During training, we observe that the share structure will lead to a convergence rate inconsistency issue which is harmful to performance. So we propose a share-aware learning mechanism to address it. We evaluate the proposed model on multiple benchmark datasets. Extensive experimental results show that the proposed model is very effective and it achieves state-of-the-art results on all of these datasets. Moreover, experiments show that both the proposed bidirectional extraction framework and the share-aware learning mechanism have good adaptability and can be used to improve the performance of other tagging based methods. The source code of our work is available at: https://github.com/neukg/BiRTE.

CCS CONCEPTS

• **Computing methodologies → Information extraction**.

[*]Corresponding author, renfeiliang@cse.neu.edu.cn.
[†]Both authors contribute equally to this research and are listed randomly.

WSDM '22, February 21–25, 2022, Tempe, AZ, USA

ACM ISBN 978-1-4503-9132-0/22/02…$15.00
https://doi.org/10.1145/3488560.3498409

KEYWORDS

relational triple extraction, joint extraction of entities and relations, overlapping triple issue, bidirectional extraction framework, convergence rate inconsistency issue, share-aware learning mechanism

ACM Reference Format:
Feiliang Ren, Longhui Zhang, Xiaofeng Zhao, Shujuan Yin, Shilei Liu, and Bochao Li. 2022. A Simple but Effective Bidirectional Framework for Relational Triple Extraction. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining (WSDM '22), February 21–25, 2022, Tempe, AZ, USA*. ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3488560.3498409

1 INTRODUCTION

The task of relational triple extraction (RTE for short) is to extract triples from unstructured natural language text (often sentences). These relational triples store factual knowledge in the form of (*subject, relation, object*), where both *subject* and *object* are entities and they are connected semantically by *relation*. For example, a triple (*Washington, capital_of, the United States*) expresses the knowledge that "*Washington is the capital of the United States*". Nowadays, RTE are attracting more and more research interest due to its importance for many downstream applications like automatic knowledge graph construction, and many novel RTE methods have been proposed.

Early RTE methods [2, 28, 35] often use a pipeline based extraction framework that recognizes all entities in the input text first, and then predicts the relations for all the combinations of entity pairs. These methods are flexible for they can make full use of existing achievements in the research domains of both name entity recognition and relation classification. But they have following two fatal deficiencies. First, they ignore the correlations between entity recognition and relation prediction. Second, they suffer from the error propagation issue greatly. Thus more and more researchers begin to explore a kind of joint extraction methods that extracts entities and relations simultaneously in an end-to-end way, and lots of novel joint extraction methods have been proposed [1, 7, 8, 15, 23, 24, 26, 27, 29, 34].

Among these joint extraction methods, a kind of tagging based methods [24, 26, 34] show great superiority in both the performance and the ability of extracting triples from following two kinds of complex sentences. The first kind is the sentences that contain overlapping triples (a single entity or an entity pair participates in multiple relational triples of the same sentence [31]). The second kind is the sentences that contains multiple triples. Existing tagging based methods often decompose the whole RTE task into two tagging based subtasks: the first one recognizes all subjects and the second one recognizes all objects and relations simultaneously. For convenience, we call them as unidirectional extraction framework based methods. Despite the great success, they are far from their full potential because they suffer from the following issue greatly: once the extraction of a subject is failed, the extraction of all triples associated with this subject would be failed accordingly. Here we call an entity as a ground entity if it is extracted firstly in a triple, and call the mentioned issue as *ground entity extraction failure*. Obviously, this issue is harmful to the performance of RTE greatly.

To address the mentioned issue, we propose *BiRTE*, a *Bi*directional extraction framework based *R*elational *T*riple *E*xtraction model. It follows the tagging based extraction route but takes both subjects and objects as ground entities. Our method is mainly inspired by following cognition: if we take both subjects and objects as ground entities and extract triples from the directions of "*subject->object->relation*" and "*object->subject->relation*", then even if a ground entity is failed to be extracted in one direction, it is still possible to be extracted from another direction (but not as a ground entity), thus its associated triples are still possible to be extracted from another direction accordingly. Thus, the mentioned *ground entity extraction failure* issue can be well addressed inherently.

Based on above cognition, *BiRTE* is designed as follows. First, it extracts *subject-object* (*s-o* for short) pairs from the directions of *subject-to-object* (*s2o* for short) and *object-to-subject* (*o2s* for short). These two extraction directions work in parallel but are connected by a shared encoder component, which makes the extraction features from one direction can be injected into the extraction features of another direction, and vice versa. This extraction structure brings an obvious advantage: the extractions of two directions complement each other and their extraction results can be validated each other. And such advantage is much helpful for the whole triple extraction since reliable *s-o* pairs are the foundation of extracting accurate triples. With this bidirectional extraction framework, lots of *s-o* pairs are extracted. Among them, there are also many noise pairs that do not possess any relations. Thus a strong relation classification model is required. In *BiRTE*, we use a biaffine model to assigns all possible relations for each *s-o* pair. Given a *s-o* pair, the biaffine model can mine deep interactions between the subject and the object, thus all the relations of this *s-o* pair can be easily extracted. Besides, during training, we observe there would be a harmful *convergence rate inconsistency issue* caused by the share structure. To overcome it, we propose a *share-aware* learning mechanism which assigns different learning rates for different modules.

We evaluate *BiRTE* on multiple benchmark datasets. Extensive experiments show it consistently outperforms existing best RTE methods on all datasets, and achieves new state-of-the-art results.

2 RELATED WORK

At present, the joint extraction methods are becoming dominated in RTE. According to the extraction routes taken, we roughly classify them into following three main kinds.

Tagging Based Methods In this kind of methods, binary tagging sequences are often used to determine the start and end positions of entities, sometimes are also used to determine the relations between two entities too. For example, [34] propose a tagging based framework that converts the joint extraction task into a tagging problem to extract entities and their relations directly. Recently, researchers [24, 26] begin to explore a unidirectional extraction framework based tagging methods: first extract all subjects, and then extract objects and relations simultaneously based on the subjects extracted. Especially, CasRel [24], one of the most latest tagging based methods, provides a fresh perspective for the RTE task: it models relations as functions that map subjects to objects. Experiments show that CasRel not only achieves very competitive results, but also has strong ability for extracting triples from sentences that contain overlapping triples or multiple triples.

Table Filling Based Methods This kind of methods [11, 14, 23, 32] would maintain a $l \times l$ table for each relation (l is the number of tokens in an input sentence), and the items in this table usually denotes the start and end positions of two entities (or even the types of these entities) that possess this specific relation. So the RTE task in this kind of methods is converted into the task of filling these tables accurately and effectively.

Seq2Seq Based Methods This kind of methods often view a triple as a token sequence, and convert the RTE task into a *generation* task that generates a triple in some orders, such as first generates a relation, then generates entities, etc. For example, [16] use an encoder-decoder architecture in their method. [25] propose a contrastive triple extraction method with a generative transformer. Other representative work of this kind includes [29–31].

Researchers also explore other extraction routes for RTE. For example, [3] propose a unified framework to extract explicit and implicit relational triples jointly. [22] provide a revealing insight into RTE from a stereoscopic perspective. [33] propose a joint RTE framework based on potential relation and global correspondence.

3 METHODOLOGY

The architecture of *BiRTE* is shown in Figure 1, from which we can see that it consists of following three main components: a BERT based *Encoder* component, a *Bi*directional *E*ntity *P*air *E*xtraction component (*BiEPE* for short), and a biaffine based *R*elation *E*xtraction component (*RE* for short). During training, the modules in *BiEPE* and *RE* work in a multi-task learning manner. This brings an advantage that each module can be trained with the ground-truth input, thus a more reliable model can be obtained. But in the inference phase, *BiEPE* and *RE* work in a sequential manner.

3.1 Encoder

Here we first use a pre-trained BERT (Cased) [4] model to generate an initial representation for each token (denoted as $\mathrm{h}^{(\cdot)} \in \mathcal{R}^{d_h}$) in an input sentence. Then we generate three distinct token representation sequences as a kind of context features for subjects, objects, and relations respectively. This is much different from most of existing state-of-the-art methods like CasRel [24], TPLinker [23], or PMEI [20]: all of them use an unified feature for subjects, objects, and relations. But we think that different kinds of items in triples have their own characteristics. Thus, it would be more reasonable

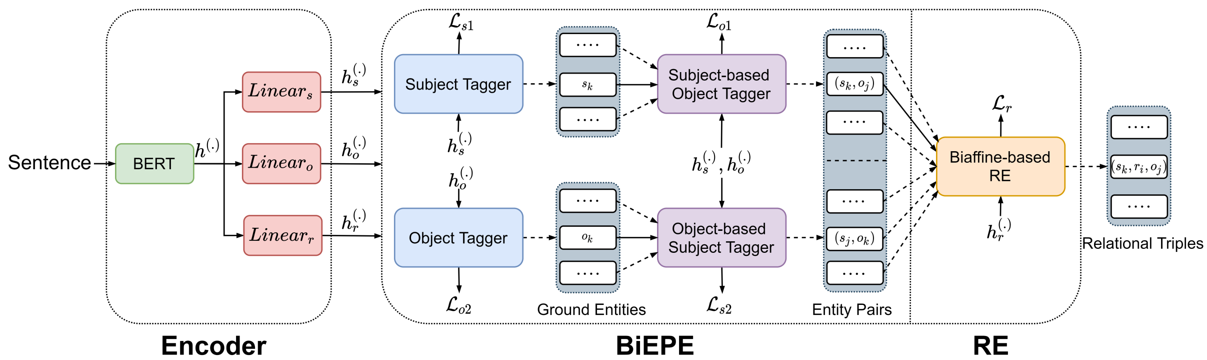

Figure 1: The Architecture of BiRTE. The modules with the same color have the similar inner structures. The solid lines represent the training process, and the dashed lines together with the solid lines represent the inference phase.

to use different features for them. Concretely, we denote the i-th token representations of these three contextual features as h_s^i , h_o^i , and h_r^i respectively, and they are computed with Eq.(1).

$$\begin{aligned} \mathrm{h}_s^i &= \mathrm{W}_s \mathrm{h}^i + \mathrm{b}_s \\ \mathrm{h}_o^i &= \mathrm{W}_o \mathrm{h}^i + \mathrm{b}_o \\ \mathrm{h}_r^i &= \mathrm{W}_r \mathrm{h}^i + \mathrm{b}_r \end{aligned} \tag{1}$$

where $\mathrm{W}_{(.)} \in \mathcal{R}^{d_h \times d_h}$ is a trainable matrix, $\mathrm{b}_{(.)} \in \mathcal{R}^{d_h}$ is a bias vector, and d_h is the dimension.

Besides, considering the subject and object in a triple are highly correlated, the features from one entity would be helpful for the extraction of another entity. So we add the object token representation sequence's *CLS* vector (denoted as h_o^{cls}) to h_s^i to enhance the representation ability of the subject's context features. Similar operation is also performed on the object part, as shown in Eq.(2).

$$\begin{aligned} \mathrm{h}_s^i &= \mathrm{h}_s^i + \mathrm{h}_o^{cls} \\ \mathrm{h}_o^i &= \mathrm{h}_o^i + \mathrm{h}_s^{cls} \end{aligned} \tag{2}$$

3.2 BiEPE

BiEPE has a bidirectional framework that extracts *s-o* pairs from following two directions: (i) a *s2o* direction that first extracts subjects and then extracts objects conditioned on the subjects extracted, and (ii) a reversed *o2s* direction that first extracts objects and then extracts subjects. These two directions' extractions share the *Encoder* component. The inner structures of two directions are similar, so here we only introduce the direction of *s2o* for space saving.

Subject Tagger is a binary tagging based module that aims to extract all subjects from an input sentence. For each token in the input sentence, two probabilities are assigned to denote the possibilities of it being the start token and end token of a subject. Specifically, these two probabilities are computed with Eq.(3).

$$\begin{aligned} p_s^{i,start} &= \sigma\left(\mathrm{W}_s^{start} \mathrm{h}_s^i + \mathrm{b}_s^{start}\right) \\ p_s^{i,end} &= \sigma\left(\mathrm{W}_s^{end} \mathrm{h}_s^i + \mathrm{b}_s^{end}\right) \end{aligned} \tag{3}$$

where $p_s^{i,start}$ and $p_s^{i,end}$ represent the probabilities of the i-th token being the start token and end token of a subject respectively. $\mathrm{W}_s^{(.)} \in \mathcal{R}^{1 \times d_h}$ is a trainable matrix, $\mathrm{b}_s^{(.)} \in \mathcal{R}^1$ is a bias vector. In all equations of this paper, σ denotes a sigmoid activation function.

In this study, we use a simple *1/0* tagging scheme, which means a token will be assigned a *1* tag if its probability exceeds a certain threshold or a *0* tag otherwise.

Subject-based Object Tagger is used to extract all objects conditioned on the subjects extracted. To this end, it designs an iterative tagging structure that takes the subjects extracted one by one and extracts all the objects for each selected subject.

Specifically, given a selected subject, each token in the input sentence are assigned two probabilities to denote the possibilities of it being the start token and end token of an object that is related to this selected subject. And these two kinds of probabilities are computed with Eq. (4).

$$\begin{aligned} \mathrm{v}_s^{s\_k} &= \mathrm{maxpool}\left(\mathrm{h}_s^{s\_k\_start}, \ldots, \mathrm{h}_s^{s\_k\_end}\right) \\ p_o^{i,start} &= \sigma\left(\mathrm{W}_o^{start}\left(\mathrm{h}_o^i \circ \mathrm{v}_s^{s\_k}\right) + \mathrm{b}_o^{start}\right) \\ p_o^{i,end} &= \sigma\left(\mathrm{W}_o^{end}\left(\mathrm{h}_o^i \circ \mathrm{v}_s^{s\_k}\right) + \mathrm{b}_o^{end}\right) \end{aligned} \tag{4}$$

where $\mathrm{h}_s^{s\_k\_start}, \ldots, \mathrm{h}_s^{s\_k\_end}$ are the vector representations of the tokens in the k-th subject, so $\mathrm{v}_s^{s\_k}$ can be viewed as a representation for the k-th subject. $\mathrm{maxpool}(.)$ means the *max-pooling* operation. $p_o^{i,start}$ and $p_o^{i,end}$ are the probabilities of the i-th token being the start and end tokens of an object related to the k-th subject respectively. $\circ$ denotes a hadamard product operation. $\mathrm{W}_o^{(.)} \in \mathcal{R}^{1 \times d_h}$ is a trainable matrix, and $\mathrm{b}_o^{(.)} \in \mathcal{R}^1$ is a bias vector.

Cross Entropy based Losses As mentioned above, all the extraction modules in two directions work in a multi-task learning manner. Thus, both extraction modules in each direction have their own loss functions. We denote the losses of above two tagger modules as $\mathcal{L}_{s1}$ and $\mathcal{L}_{o1}$ respectively, and both of them are defined with a binary cross entropy based loss, as shown in Eq. (5).

$$\begin{aligned}
&\mathrm{ce}\,(p,t) = -\left[t log p + (1-t)\, log\,(1-p)\right] \\
&\mathcal{L}_{s1} = \frac{1}{2\times l}\sum_{m\in\{\text{start},\text{end}\}}\sum_{i=1}^{l}\mathrm{ce}\left(p_s^{i,m}, t_s^{i,m}\right) \\
&\mathcal{L}_{o1} = \frac{1}{2\times l}\sum_{m\in\{\text{start},\text{end}\}}\sum_{i=1}^{l}\mathrm{ce}\left(p_o^{i,m}, t_o^{i,m}\right)
\end{aligned} \tag{5}$$

where $\mathrm{ce}(p,t)$ is a binary cross entropy loss, $p \in (0,1)$ is the predicted probability and t is the true tag, l is the number of tokens in an input sentence.

Similarly, there are two tagger losses in the *o2s* direction. We denote them as $\mathcal{L}_{s2}$ and $\mathcal{L}_{o2}$ respectively and they are computed with the similar method as shown in Eq. (5).

3.3 RE

The proposed framework makes *BiEPE* output more *s-o* pairs, where there are many noise pairs. This is harmful to the precision of *BiRTE*. Thus, RE should have a strong classification ability. Here we use a biaffine model [6, 10] for the RE module. It maintains a parameter matrix for each relation, and an entity pair will be computed with each relation-specific matrix to determine whether it possesses the corresponding relation or not. Specifically, for an entity pair (s_k, o_j), we first obtain the representation vectors $\mathrm{v}_r^{s\_k}$ and $\mathrm{v}_r^{o\_j}$ for its two entities. Then the possibility (denoted as p_r^i) of (s_k, o_j) possessing the i-th relation is computed. The process is shown in Eq. (6), where $\mathrm{W}_r^i \in \mathcal{R}^{(d_h+1)\times(d_h+1)}$ is the parameter matrix of the i-th relation.

$$\begin{aligned}
&\mathrm{v}_r^{s_k} = \mathrm{maxpool}\left(\mathrm{h}_r^{s_k_start}, \dots, \mathrm{h}_r^{s_k_end}\right) \\
&\mathrm{v}_r^{o_j} = \mathrm{maxpool}\left(\mathrm{h}_r^{o_j_start}, \dots, \mathrm{h}_r^{o_j_end}\right) \\
&p_r^i = \sigma\left(\begin{bmatrix} \mathrm{v}_r^{s_k} \\ 1 \end{bmatrix}^{\top} \mathrm{W}_r^i \begin{bmatrix} \mathrm{v}_r^{o_j} \\ 1 \end{bmatrix}\right)
\end{aligned} \tag{6}$$

Here we select the biaffine model mainly due to its following two advantages. First, it maintains a matrix for each relation, which can model the characteristics of a relation accurately. Second, its probability computation mechanism makes it can accurately mine the interactions between a subject and an object. Both advantages are much helpful for improving the extraction precision.

RE Loss To train the *RE* component, we also define a cross entropy based loss, as shown in Eq. (7), where R is the predefined relation set and $|R|$ is the number of total relations.

$$\mathcal{L}_r = \frac{1}{|R|}\sum_{i=1}^{|R|}\mathrm{ce}\left(p_r^i, t_r^i\right) \tag{7}$$

3.4 Share-aware Learning Mechanism

Totally, there are five extraction modules in *BiRTE*. During the multi-task learning based training, each of them will form a relative independent extraction *task* with the *Encoder* module. We use the popular *teacher forcing* mode to train all the *tasks* except the ones that *ONLY* take original sentence as input. Under this mode, each *task* randomly selects some correct samples as input for training. Besides, to alleviate the *exposure bias* issue [24], we merge some randomly generated negative samples into the correct samples and use them together to train these *tasks* where the *teacher forcing* mode used. The negative samples can simulate the real scenario in the inference phase, which is helpful for training a robust model. Accordingly, the mentioned *exposure bias* issue is alleviated greatly. Finally, the overall loss of *BiRTE* is defined with Eq.(8).

$$\mathcal{L} = \mathcal{L}_{s1} + \mathcal{L}_{o1} + \mathcal{L}_{s2} + \mathcal{L}_{o2} + \mathcal{L}_r \tag{8}$$

However, we observe that the parameters in the shared *Encoder* module will receive back propagated gradients from the parameters of each extraction module. Consequently, the convergence rate of the *Encoder* module will be much different from those in other extraction modules. This will result in a *convergence rate inconsistency issue*, which means if we set a unified learning rate for these five extraction modules and the *Encoder* module, it would be difficult for them to converge to their optimal points simultaneously. In other words, some modules will be over-trained while others will be under-trained under a unified learning rate.

So we propose a *share-aware* learning mechanism that assigns different learning rates for different modules. The basic idea of this mechanism is that the more *tasks* a module is shared by, the smaller learning rate it should be assigned. For example, the *Encoder* module should be assigned a smaller learning rate than other extraction modules since it is shared by more *tasks*. Specifically, the proposed learning mechanism assigns learning rates with Eq.(9).

$$\xi_i = \begin{cases} \xi, & k_i = 1 \\ \frac{(1+\delta)}{f(k_i)} * \xi, & k_i > 1 \end{cases} \tag{9}$$

where ξ is a base learning rate, ξ_i is the learning rate for the i-th module and k_i is the number of *tasks* that the i-th module is shared by. For example, in *BiRTE*, the corresponding k of the *Encoder* module would be 5 since this module is shared by all the five *tasks*, while the corresponding k of the *subject tagger* module in the *s2o* direction would be 1 since this module is only used by its own *task*. $\delta \in [0,1]$ is a regulatory factor that is used to finely adjust the learning rate, and $f(.)$ is a mapping function that transforms the input k_i to a reasonable real value (often larger than 1) so as to determine the major magnitude of the learning rate.

Category	NYT		WebNLG		NYT10		NYT11	
	Train	Test	Train	Test	Train	Test	Train	Test
Normal	37013	3266	1596	246	59396	2963	53395	368
EPO	9782	978	227	26	5376	715	2100	0
SEO	14735	1297	3406	457	8772	742	7365	1
ALL	56195	5000	5019	703	70339	4006	62648	369

Table 1: Statistics of datasets. *EPO* and *SEO* refer to the *entity pair overlapping* and *single entity overlapping* respectively [31]. Note a sentence can belong to both *EPO* and *SEO*.

4 EXPERIMENTS

4.1 Experiment Settings

Datasets We evaluate *BiRTE* on following benchmark datasets: NYT [18], WebNLG [9], NYT10 [18], and NYT11 [12]. To be fair, we follow some latest work [20, 23, 24], which uses the preprocessed

NYT and WebNLG datasets released by [31], and uses the preprocessed NYT10 and NYT11 datasets released by [21]. Some statistics of these datasets are shown in Table 1.

Note that both NYT and WebNLG have two different versions according to following two annotation standards: 1) annotating the last token of the entities, and 2) annotating the whole entity span. Different work chooses different versions of these datasets. For convenience, we denote the datasets based on the first standard as NYT* and WebNLG*, and the datasets based on the second standard as NYT and WebNLG. Obviously, the full annotated datasets can reveal the true performance of a model better.

Besides, [24] point out that both NYT10 and NYT11 are far less popular than either NYT or WebNLG, and they are usually used to show the generalization capability of a model because most test sentences in them belong to the *Normal* class. Thus, for space saving, we adopt them only in the main experiment part.

Evaluation Metrics The standard micro precision, recall, and *F1* score are used to evaluate the results. There are two match standards for the RTE task: one is *Partial Match* that an extracted triple is regarded as correct if the predicted relation and the head of both subject entity and object entity are correct; and the other is *Exact Match* that a triple is regarded as correct only when its entities and relation are completely matched with a correct triple. To be fair, we follow previous work [20, 23, 24] and use *Partial Match* on NYT* and WebNLG*, use *Exact Match* on NYT and WebNLG.

Implementation Details AdamW [13] is used to train *BiRTE*. The threshold for judging whether there a subject, an object, or a relation is set to 0.5. In Eq.(9), ξ is set to $1.5e^{-4}$, the regulatory factor δ is set to 0, and the mapping function $f(.)$ is defined as an identity function. The batch size is set to 18 on NYT, NYT*, NYT10 and NYT11, and is set to 6 on WebNLG and WebNLG*. All involved hyperparameters are determined based on the results on the development sets. Other parameters are randomly initialized. In experiments, all the involved *BERT* model refers to *BERT (base)*. On all datasets, we run our model 5 times and the averaged results are taken as the final reported results.

Baselines Following strong state-of-the-art models are taken as baselines, including: *ETL-Span* [26], *WDec* [16], *RSAN* [27], *RIN* [19], *CasRel* [24], *TPLinker* [23], *StereoRel* [22], *PRGC* [33], *R-BPtrNet* [3], *PMEI* [20], and *CGT* [25]. Most results of these baselines are copied from their original papers directly. Moreover, following previous work [3, 20, 23, 24], we also implement a *BiLSTM*-encoder version of *BiRTE* where 300-dimensional GloVe embeddings [17] and 2-layer stacked BiLSTM are used. Some baselines did not report their results on some datasets. In such case, we report the best results we obtained (marked by *) by running the source code (if available). But if a baseline did not report the results of its *BiLSTM*-encoder version, we would not obtain these results even if the source code is available: because it needs to modify the provided source code in such case, which will increase the concern of whether such modification is correct and whether the obtained results are objective.

4.2 Experimental Results

Main Results The main results are shown in Table 2. On all datasets, *BiRTE* achieves almost all the best results in term of *F1* when compared with the models that use the same kind of encoder (*BERT* or *BiLSTM*). When considering the complete version of each model where *BERT* used, *BiRTE* works much better than all the compared baselines: it achieves the best results on almost all datasets in term of recall and *F1*. *BiRTE* achieves slightly poor but still much competitive precision results. This is in line with our previous analyses that some noise pairs are extracted by the bidirectional framework, which is harmful to precision. However, the proposed framework brings much more benefits on recall, which makes *BiRTE* achieves much higher *F1* scores. Another interesting observation is that *BiRTE* achieves far better results than *CasRel*, which proves the correctness of our motivation.

Besides, *BiRTE* achieves better *F1* results on all the full annotated datasets. This is very meaningful because it indicates that *BiRTE* will perform well when deployed in real scenarios where both the *full annotation* standard and the *exact match* standard are usually required. *BiRTE* also achieves much better results than all the compared baselines on both NYT10 and NYT11, which indicates that *BiRTE* has a good generalization capability.

In subsequent sections, we evaluate *BiRTE* from diverse aspects, and all the results are obtained when the BERT-based encoder used.

Evaluations on Complex Sentences Here we evaluate *BiRTE*'s ability for extracting triples from sentences that contain overlapping triples and multiple triples. This ability is widely discussed in existing models, and is an important metric to evaluate the robustness of a model. For fair comparison, we follow the settings of some previous best models [3, 23, 24, 33], which are: (i) classifying sentences according to the degree of overlapping and the number of triples contained in a sentence, and (ii) conducting experiments on different subsets of NYT* and WebNLG*.

The results are shown in Table 3. We can see that *BiRTE* has great superiority for handling complex sentences. On both datasets, it achieves much better results than the compared baselines on most cases. Moreover, *BiRTE* achieves more performance improvement when handling the sentences of *SEO* class. This is mainly because that a single entity in a *SEO* sentence may associate with multiple triples, thus the existing models (even including the non-tagging based models like *TPLinker*) are more likely to suffer from the *ground entity extraction failure* issue on the *SEO* sentences than on other types of sentences: once the extraction of an entity in some *SEO* triples is failed, all the associated triples of this entity would not be extracted either. But the bidirectional framework in *BiRTE* can effectively overcome such deficiency and the mentioned issue almost has no effect on it when handling the *SEO* sentences. This is also the reason why *BiRTE* performs well on sentences that contain multiple triples. Note *R-BPtrNet* [3] also achieves very competitive results, which is partly because it uses extra entity type knowledge.

Detailed Evaluations Here we make three kinds of detailed evaluations on *BiRTE*, and the results are shown in Table 4.

First, we evaluate the contributions of the proposed bidirectional extraction framework from following four aspects.

(1) We evaluate whether the proposed bidirectional extraction framework is superior to the unidirectional extraction frameworks. To this end, we implement following two variants of *BiRTE*: (i) BiRTE$_{s2o}$, a variant that only uses the *s2o* direction to extract entity pairs, based on which the triples are extracted; (ii) BiRTE$_{o2s}$, a variant that only uses the *o2s* direction to extract entity pairs, based on which the triples are extracted. Results show that the performance

Model	Partial Match						Exact Match					
	NYT*			WebNLG*			NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
ETL-Span [26]	84.9	72.3	78.1	84.0	91.5	87.6	85.5	71.7	78.0	84.3	82.0	83.1
WDec [16]	–	–	–	–	–	–	88.1	76.1	81.7	–	–	–
RSAN [27]	–	–	–	–	–	–	85.7	83.6	84.6	80.5	83.8	82.1
RIN [19]	87.2	87.3	87.3	87.6	87.0	87.3	83.9	85.5	84.7	77.3	76.8	77.0
CasRel$_{LSTM}$ [24]	84.2	83.0	83.6	86.9	80.6	83.7	–	–	–	–	–	–
PMEI$_{LSTM}$ [20]	88.7	86.8	87.8	88.7	87.6	88.1	84.5	84.0	84.2	78.8	77.7	78.2
TPLinker$_{LSTM}$ [23]	83.8	83.4	83.6	90.8	90.3	90.5	86.0	82.0	84.0	**91.9**	81.6	86.4
R-BPtrNet$_{LSTM}$‡ [3]	90.9	91.3	91.1	90.7	**94.6**	92.6	–	–	–	–	–	–
CGT$_{UniLM}$ [25]	**94.7**	84.2	89.1	92.9	75.6	83.4	–	–	–	–	–	–
CasRel$_{BERT}$ [24]	89.7	89.5	89.6	93.4	90.1	91.8	89.8*	88.2*	89.0*	88.3*	84.6*	86.4*
PMEI$_{BERT}$ [20]	90.5	89.8	90.1	91.0	92.9	92.0	88.4	88.9	88.7	80.8	82.8	81.8
TPLinker$_{BERT}$ [23]	91.3	92.5	91.9	91.8	92.0	91.9	91.4	92.6	92.0	88.9	84.5	86.7
StereoRel$_{BERT}$ [22]	92.0	92.3	92.2	91.6	92.6	92.1	92.0	92.3	92.2	–	–	–
PRGC$_{BERT}$ [33]	93.3	91.9	92.6	94.0	92.1	93.0	**93.5**	91.9	92.7	89.9	87.2	88.5
R-BPtrNet$_{BERT}$‡ [3]	92.7	92.5	92.6	**93.7**	92.8	93.3	–	–	–	–	–	–
BiRTE$_{LSTM}$	86.5	89.0	87.7	90.5	91.6	91.0	86.4	87.1	86.7	85.2	87.3	86.2
BiRTE$_{BERT}$	92.2	**93.8**	**93.0**	93.2	94.0	**93.6**	91.9	**93.7**	**92.8**	89.0	**89.5**	**89.3**

Model	Partial Match						Exact Match					
	NYT10			NYT11			NYT10			NYT11		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
PMEI$_{LSTM}$ [20]	79.1	67.2	72.6	56.0	58.6	57.2	75.4	65.8	70.2	**55.3**	57.8	56.5
CasRel$_{BERT}$ [24]	77.7	68.8	73.0	50.1	58.4	53.9	76.8*	68.0*	72.1*	49.1*	56.4*	52.5*
StereoRel$_{BERT}$ [22]	80.0	67.4	73.2	53.8	55.4	54.6	–	–	–	–	–	–
PMEI$_{BERT}$ [20]	79.1	70.4	74.5	55.8	59.7	57.7	77.3	69.7	73.3	54.9	58.9	56.8
TPLinker$_{BERT}$ [23]	78.9*	71.1*	74.8*	55.9*	60.2*	58.0*	78.5*	68.8*	73.4*	54.8*	59.3*	57.0*
BiRTE$_{LSTM}$	79.0	68.8	73.5	55.1	60.4	57.6	76.1	67.4	71.5	54.1	60.5	57.1
BiRTE$_{BERT}$	**80.6**	**71.8**	**76.0**	**56.4**	**62.0**	**59.1**	**80.1**	**71.4**	**75.5**	55.0	**61.2**	**57.9**

Table 2: Main experiments. Note *CGT* uses *UniLM* [5]. ‡: R-BPtrNet uses extra entity type features while all other models not.

Model	NYT*								WebNLG*							
	Normal	SEO	EPO	T = 1	T = 2	T = 3	T = 4	T ≥ 5	Normal	SEO	EPO	T = 1	T = 2	T = 3	T = 4	T ≥ 5
CasRel$_{BERT}$ [24]	87.3	91.4	92.0	88.2	90.3	91.9	94.2	83.7	89.4	92.2	94.7	89.3	90.8	94.2	92.4	90.9
TPLinker$_{BERT}$ [23]	90.1	93.4	94.0	90.0	92.8	93.1	96.1	90.0	87.9	92.5	95.3	88.0	90.1	94.6	93.3	91.6
PRGC$_{BERT}$ [33]	91.0	94.0	94.5	91.1	93.0	93.5	95.5	**93.0**	90.4	93.6	95.9	89.9	91.6	95.0	94.8	92.8
R-BPtrNet$_{BERT}$ [3]	90.4	94.4	**95.2**	89.5	93.1	93.5	**96.7**	91.3	89.5	93.9	**96.1**	88.5	91.4	**96.2**	**94.9**	**94.2**
BiRTE$_{BERT}$	**91.4**	**94.7**	94.2	**91.5**	**93.7**	**93.9**	95.8	92.1	**90.1**	**95.9**	94.3	**90.2**	**92.9**	95.7	94.6	92.0

Table 3: F1 scores on sentences with different overlapping pattern and different triplet number. Results of *CasRel* are copied from *TPLinker* directly. "T" is the number of triples contained in a sentence.

of both variants drops on all datasets, which shows the superiority of the proposed bidirectional framework. Especially, both variants achieve lower recalls, which indicates again that the unidirectional extraction framework based models are sensitive to the *ground entity extraction failure* issue. While in *BiRTE*, the two directions' *s-o* pair extractions can boost each other, so the mentioned issue is alleviated greatly, which is much helpful for recall.

(2) We evaluate whether the proposed bidirectional extraction framework does be helpful for extracting better ground entities than the unidirectional frameworks. To this end, we compare the ground entities' extraction results between *BiRTE*, BiRTE$_{s2o}$, and BiRTE$_{o2s}$. The results are shown in Table 5. We can see that in each direction, *BiRTE* achieves much better extraction results than its variant of the same direction. This is mainly because that with the help of the multi-task learning mechanism, the ground entity extractions of two directions can boost each other by the explicitly injected context features through the shared *Encoder* component, which is much helpful for the extraction results of each direction.

Model	Partial Match						Exact Match					
	NYT*			WebNLG*			NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
BiRTE$_{BERT}$	**92.2**	**93.8**	**93.0**	**93.2**	**94.0**	**93.6**	**91.9**	93.7	**92.8**	**89.0**	**89.5**	**89.3**
BiRTE$_{s2o}$	91.5	91.3	91.4	92.0	90.4	91.2	91.5	91.0	91.2	88.3	87.0	87.6
BiRTE$_{o2s}$	91.4	91.0	91.2	91.8	90.5	91.1	91.5	90.8	91.1	88.5	87.5	88.0
BiRTE$_{FinePipeline}$	90.4	91.2	90.8	91.0	91.6	91.3	89.7	90.1	89.9	84.0	85.6	84.8
BiRTE$_{CoarsePipeline}$	90.9	92.3	91.6	91.9	92.1	92.0	90.5	91.0	90.7	85.7	87.3	86.5
BiRTE$_{OneLr}$	91.0	92.4	91.7	92.5	93.6	93.0	91.2	91.8	91.5	88.1	89.0	88.5
BiRTE$_{uif}$	91.6	92.9	92.2	92.7	93.8	93.2	91.3	92.5	91.9	88.8	88.6	88.7
BiRTE$_{tru}$	92.1	93.4	92.7	93.2	93.8	93.5	91.5	93.2	92.3	88.9	89.3	89.1
BiRTE$_{BIO}$	92.1	93.7	92.9	93.0	93.9	93.4	**91.9**	93.8	**92.8**	88.8	**89.5**	89.1
BiRTE$_{2step}$	89.5	92.3	90.9	89.9	91.9	90.9	89.0	91.5	90.2	84.7	87.6	86.1
BiRTE$_{Li}$	91.0	93.6	92.3	91.6	92.9	92.2	90.5	**93.9**	92.2	87.2	89.3	88.2

Table 4: Results of detailed evaluations.

Models	Direction	NYT*	WebNLG*	NYT	WebNLG
BiRTE	s2o	95.0	95.3	94.2	91.0
	o2s	94.8	95.6	93.9	91.1
BiRTE$_{s2o}$	s2o	93.6	92.6	93.1	89.3
BiRTE$_{o2s}$	o2s	93.2	92.8	92.8	89.5

Table 5: F1 results of the ground entity extraction.

Models	NYT*	WebNLG*	NYT	WebNLG
ETL-Span	54.3	56.1	56.8	60.2
CasRel	49.7	48.5	55.7	51.8
BiRTE$_{s2o}$	55.2	39.6	56.0	42.8
BiRTE$_{o2s}$	53.5	51.2	54.8	52.2
BiRTE	9.7	5.4	11.0	9.3

Table 6: Proportions (%) of triples that are not extracted due to the *ground entity extraction failure* issue.

(3) We compare the proportion of the triples that are not extracted due to the *ground entity extraction failure* issue between *BiRTE* and other tagging based methods that take an unidirectional extraction framework. This proportion can quantitatively demonstrate both the severity caused by the mentioned issue and the practical effect of the proposed bidirectional extraction framework. The results are shown in Table 6.

We can see that for all the unidirectional extraction framework based models, almost half of the failed extracted triples are caused by the mentioned *ground entity extraction failure* issue. While for *BiRTE*, this proportion drops sharply. These results show that the harm of the mentioned issue is eliminated greatly by the proposed bidirectional framework.

(4) We evaluate whether a simple combination of two paralleled extraction components can also performs well like the proposed framework. To this end, we implement following two variants of *BiRTE*, both of which are *pipeline*-based models. (i) BiRTE$_{FinePipeline}$, a model that splits *Subject Tagger*, *Object Tagger*, *Subject-based Object Tagger*, *Object-based Subject Tagger*, and *RE* into five separated models that do not share the *Encoder* component; and (ii) BiRTE$_{CoarsePipeline}$, a model that splits *BiEPE* and *RE* into two separated models that do not share the *Encoder* component.

Results show that the performance of both variants drops sharply on all datasets, which indicates that the proposed extraction framework should NOT be viewed as a simple combination of two individual extraction components. In fact, under the multi-task learning mechanism, the *Encoder*-share structure in our framework enables different modules complement and boost each other, which is much helpful for the performance of the whole RTE task. For example, in each direction, either *Subject Tagger* or *Object Tagger* will push parameters in *Encoder* to be updated in the way that is beneficial for its own extraction. As these two taggers are performed alternately in the multi-task learning manner, features that are beneficial for the *subject extraction* are injected into the parameters of *Encoder* by the back propagated gradients, based on which the *object extraction* is performed, and vice versa. Thus, the *subject-related* features are implicitly used for *object extraction*, which makes two taggers complement and boost each other.. Besides, both variants have a greater *F1* degradation on WebNLG* and WebNLG than that on other two datasets. This is mainly because WebNLG is a sparse dataset for it contains a smaller number of training samples but a larger number of relations. Thus on WebNLG, the scarcity of training samples can be effectively compensated by the proposed framework by making the correlated modules boost each other.

Second, we evaluate the contributions of the proposed share-aware learning mechanism from following two aspects.

(1) We evaluate the performance difference between using and without using the proposed learning mechanism. To this end, we implement BiRTE$_{OneLr}$, a variant of *BiRTE* that uses an identical learning rate. From the comparison results we can see that the performance of BiRTE$_{OneLr}$ drops obviously on all datasets, which indicate: (i) the *convergence rate inconsistency issue* does exist in the models where contain some shared modules; and (ii) the proposed learning mechanism is effective for addressing this issue.

Model	Partial Match						Exact Match					
	NYT*			WebNLG*			NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
ETL-Span$_{BiDir}$	84.6	73.5(↑)	78.7(↑)	83.3	92.0(↑)	87.4	85.2	73.0(↑)	78.6(↑)	83.5	83.1(↑)	83.3(↑)
CasRel$_{BiDir}$	89.0	91.1(↑)	90.0(↑)	92.6	91.2(↑)	91.9(↑)	89.0	90.1(↑)	89.5(↑)	87.1	85.1(↑)	86.1
ETL-Span$_{SaLr}$	85.3(↑)	73.0(↑)	78.7(↑)	84.3(↑)	91.7(↑)	87.8(↑)	86.2(↑)	72.3(↑)	78.6(↑)	83.0	84.6(↑)	83.8(↑)
CasRel$_{SaLr}$	90.1(↑)	89.9(↑)	90.0(↑)	93.5(↑)	90.5(↑)	92.0(↑)	90.1(↑)	89.1(↑)	89.6(↑)	87.9	87.1(↑)	87.5(↑)

Table 7: Adaptability evaluations. "↑" denotes the performance is increased.

(2) We evaluate the influence of the mapping function in the proposed learning mechanism. To this end, we explore following two kinds of mapping functions. (i) An uniform increasing function $f(k_i) = 1 + 2(n_i - 1)k_i/(n - 1) \in [1, 1 + 2k_i]$, where n is the total number of epochs, and n_i is the current epoch number. (ii) A truncated function $f(k_i) = min(k_i, 1+2(n_i-1)k_i/(n-1)) \in [1, k_i]$. We denote the variants of *BiRTE* that use these two mapping functions as BiRTE$_{uif}$ and BiRTE$_{tru}$ respectively. Results show that the mapping function has an obvious influence on the performance. But all the models that use the proposed learning mechanism achieve significant better results than BiRTE$_{OneLr}$, which confirms again the proposed learning mechanism is effective. Note the mapping function selection is still an open issue and calls for further research.

Third, we conduct experiments to answer following two issues to further demonstrate the effectiveness of *BiRTE*.

(1) *BiRTE* uses the *1/0* tagging scheme in *BiEPE*. However, there are other widely used schemes like *BIO*, which can provide more richful label information than the *1/0* scheme. Thus, there is an issue: whether it would be better when the *BIO* scheme used?

To answer this issue, we implement BiRTE$_{BIO}$, a variant that uses the *BIO* scheme. We can see that the performance of BiRTE$_{BIO}$ drops slightly on most of cases except on NYT where it achieves close results with *BiRTE*. In fact, there are two advantages in the *1/0* scheme. First, its labels can realize the roles of most labels in the *BIO* scheme. Second, it reduces the complexity of a tagging model because with this simpler scheme, for each token, a model only needs to distinguish whether it is an entity token or not, other than to distinguish whether this token is a *beginning* or *inside* token of an entity, or not an entity token. Obviously, this simplification reduces the risk of introducing tagging errors.

(2) The *ground entity extraction failure* issue can also be solved by a simple strategy that firstly extracting all entities without distinguishing subject and object, and then using the *RE* module to classify all entity pairs. Accordingly, there would be an issue: whether a simpler *2-step extraction strategy* would work better?

To answer this issue, we implement BiRTE$_{2step}$, a *2-step extraction* based model. Results show that the performance of BiRTE$_{2step}$ drops sharply compared with *BiRTE*. Especially, the degradation of its precision is far larger than that of its recall on all datasets. This indicates that by considering all combinations of entity pairs, the *ground entity extraction failure* issue is alleviated to some extent. However, among these combinations, there are lots of noise pairs that have no any relation, which results in a more significant degradation in precision. Consequently, its F1 score drops. These results indicate BiRTE$_{2step}$ is not a good choice to address the mentioned issue because it often results in far larger degradation in precision, which neutralizes the benefits from the improvement of recall.

Adaptability Evaluations In fact, both the proposed bidirectional extraction framework and the proposed share-aware learning mechanism are adaptive and can be easily transplanted to other models. Here we evaluate their adaptabilities by transplanting them to *CasRel* and *ETL-Span*. Both these selected two models are state-of-the-art tagging based methods and have a shared *Encoder*.

Specifically, we denote the new models that use the proposed bidirectional extraction framework as CasRel$_{BiDir}$ and ETL-Span$_{BiDir}$ respectively. Both *CasRel* and *ETL-Span* first extract subjects, then extract objects and relations simultaneously. Here in their new variants, we simply merge the triples extracted from two directions as final outputted triples. We denote the new models that use the proposed share-aware learning mechanism as ETL-Span$_{SaLr}$ and CasRel$_{SaLr}$ respectively. The results are shown in Table 7.

We can see that on almost all datasets, both CasRel$_{BiDir}$ and ETL-Span$_{BiDir}$ achieve better performance than their original versions in term of *F1* and *recall*. These results further confirm that the bidirectional extraction framework can well address the *ground entity extraction failure* issue, which is much helpful for recall. These two new models' *precision* scores are lower than their original versions, this is because that there are more noise introduced by the bidirectional extraction framework, thus a stronger relation classification model is required. For example, when replacing the biaffine model with a common linear classification model that takes the concatenation of two entities' representations as input, the performance of *BiRTE* (BiRTE$_{Li}$ in Table 4) drops accordingly. We can also see that when the proposed share-aware learning mechanism used, both CasRel$_{SaLr}$ and ETL-Span$_{SaLr}$ achieve better results than their original versions on both datasets under almost all evaluation metrics, even slightly better than CasRel$_{BiDir}$ and ETL-Span$_{BiDir}$.

5 CONCLUSIONS

In this paper, we propose a simple but effective RTE model. There are two main contributions in our work. First, we observe the *ground entity extraction failure* issue existed in existing tagging based RTE methods, and propose a bidirectional extraction framework to address it. Second, we observe the *convergence rate inconsistency* issue existed in the share structures, and propose a *share-aware* learning mechanism to address it. We conduct extensive experiments on multiple benchmark datasets to evaluate the proposed model from diverse aspects. Experimental results show that the two proposed mechanisms are effective and adaptive, and they help our model achieve state-of-the-art results on all of these benchmark datasets.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No.61572120 and No.U1708261), the Fundamental Research Funds for the Central Universities (No.N181602013 and No.N2016006), Shenyang Medical Imaging Processing Engineering Technology Research Center (17-134-8-00), Ten Thousand Talent Program (No.ZX20200035), and Liaoning Distinguished Professor (No.XLYC1902057).

REFERENCES

[1] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and Chris Develder. 2018. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems With Applications* 114 (2018), 34–45.

[2] Yee Seng Chan and Dan Roth. 2011. Exploiting Syntactico-Semantic Structures for Relation Extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.* 551–560.

[3] Yubo Chen, Yunqi Zhang, Changran Hu, and Yongfeng Huang. 2021. Jointly Extracting Explicit and Implicit Relational Triples with Reasoning Pattern Enhanced Binary Pointer Network. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (Eds.). Association for Computational Linguistics, 5694–5703.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina N. Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers).* 4171–4186.

[5] Li Dong, Nan Yang, Wenhui Wang, Furu Wei, Xiaodong Liu, Yu Wang, Jianfeng Gao, Ming Zhou, and Hsiao-Wuen Hon. 2019. Unified Language Model Pre-training for Natural Language Understanding and Generation. In *Advances in Neural Information Processing Systems*, Vol. 32. 13042–13054.

[6] Timothy Dozat and Christopher D. Manning. 2016. Deep Biaffine Attention for Neural Dependency Parsing. In *ICLR (Poster).*

[7] Markus Eberts and Adrian Ulges. 2019. Span-Based Joint Entity and Relation Extraction with Transformer Pre-Training.. In *ECAI.* 2006–2013.

[8] Tsu-Jui Fu, Peng-Hsuan Li, and Wei-Yun Ma. 2019. 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.* 1409–1418.

[9] Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating Training Corpora for NLG Micro-Planners. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers).* 179–188.

[10] K.C. Goh, L. Turan, M.G. Safonov, G.P. Papavassilopoulos, and J.H. Ly. 1994. Biaffine matrix inequality properties and computational methods. In *Proceedings of 1994 American Control Conference - ACC '94*, Vol. 1. 850–855.

[11] Pankaj Gupta, Hinrich Schütze, and Bernt Andrassy. 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.* The COLING 2016 Organizing Committee, Osaka, Japan, 2537–2547.

[12] Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-Based Weak Supervision for Information Extraction of Overlapping Relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies.* 541–550.

[13] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun (Eds.).

[14] Makoto Miwa and Mohit Bansal. 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 (Volume 1: Long Papers).* Association for Computational Linguistics, Berlin, Germany, 1105–1116.

[15] Tapas Nayak and Hwee Tou Ng. 2020. Effective Modeling of Encoder-Decoder Architecture for Joint Entity and Relation Extraction. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 5 (2020), 8528–8535.

[16] Tapas Nayak and Hwee Tou Ng. 2020. Effective Modeling of Encoder-Decoder Architecture for Joint Entity and Relation Extraction. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020.* AAAI Press, 8528–8535.

[17] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP).* 1532–1543.

[18] Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases.* 148–163.

[19] Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2020. Recurrent Interaction Network for Jointly Extracting Entities and Classifying Relations. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (Eds.). Association for Computational Linguistics, 3722–3732.

[20] Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2021. Progressive Multitask Learning with Controlled Information Flow for Joint Entity and Relation Extraction. In *Association for the Advancement of Artificial Intelligence (AAAI).*

[21] Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. 2019. A Hierarchical Framework for Relation Extraction with Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 33, 1 (2019), 7072–7079.

[22] Xuetao Tian, Liping Jing, Lu He, and Feng Liu. 2021. StereoRel: Relational Triple Extraction from a Stereoscopic Perspective. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 4851–4861.

[23] Yucheng Wang, Bowen Yu, Yueyang Zhang, Tingwen Liu, Hongsong Zhu, and Limin Sun. 2020. TPLinker: Single-stage Joint Extraction of Entities and Relations Through Token Pair Linking. In *Proceedings of the 28th International Conference on Computational Linguistics.* Barcelona, Spain (Online), 1572–1582.

[24] Zhepei Wei, Jianlin Su, Yue Wang, Yuan Tian, and Yi Chang. 2020. A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics.* Association for Computational Linguistics, Online, 1476–1488.

[25] Hongbin Ye, Ningyu Zhang, Shumin Deng, Mosha Chen, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Contrastive Triple Extraction with Generative Transformer. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021.* AAAI Press, 14257–14265.

[26] Bowen Yu, Zhenyu Zhang, Xiaobo Shu, Tingwen Liu, Yubin Wang, Bin Wang, and Sujian Li. 2019. Joint Extraction of Entities and Relations Based on a Novel Decomposition Strategy.. In *ECAI.* 2282–2289.

[27] Yue Yuan, Xiaofei Zhou, Shirui Pan, Qiannan Zhu, Zeliang Song, and Li Guo. 2020. A relation-specific attention network for joint entity and relation extraction. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*, Vol. 4. 4054–4060.

[28] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of Machine Learning Research* 3, 6 (2003), 1083–1106.

[29] Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. CopyMTL: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning. *Proceedings of the AAAI Conference on Artificial Intelligence* 34, 5 (2020), 9507–9514.

[30] Xiangrong Zeng, Shizhu He, Daojian Zeng, Kang Liu, Shengping Liu, and Jun Zhao. 2019. 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).* 367–377.

[31] Xiangrong Zeng, Daojian Zeng, Shizhu He, Kang Liu, and Jun Zhao. 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 (Volume 1: Long Papers)*, Vol. 1. 506–514.

[32] Meishan Zhang, Yue Zhang, and Guohong Fu. 2017. End-to-End Neural Relation Extraction with Global Optimization. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing.* Association for Computational Linguistics, Copenhagen, Denmark, 1730–1740.

[33] Hengyi Zheng, Rui Wen, Xi Chen, Yifan Yang, Yunyan Zhang, Ziheng Zhang, Ningyu Zhang, Bin Qin, Xu Ming, and Yefeng Zheng. 2021. PRGC: Potential Relation and Global Correspondence Based Joint Relational Triple Extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (Eds.). Association for Computational Linguistics, 6225–6235.

[34] Suncong Zheng, Feng Wang, Hongyun Bao, Yuexing Hao, Peng Zhou, and Bo Xu. 2017. Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational*

Linguistics (Volume 1: Long Papers), Vol. 1. 1227–1236.
[35] GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring Various Knowledge in Relation Extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*. 427–434.