OD-RTE: A One-Stage Object Detection Framework for Relational Triple Extraction

Jinzhong Ning, Zhihao Yang*, Yuanyuan Sun, ZhiZheng Wang, Hongfei Lin
School of Computer Science and Technology, Dalian University of Technology, China
`(Jinzhong_Ning, wzz_dllg)@mail.dlut.edu.cn`
`(yangzh, syuan, hflin)@dlut.edu.cn`

Abstract

The Relational Triple Extraction (RTE) task is a fundamental and essential information extraction task. Recently, the table-filling RTE methods have received lots of attention. Despite their success, they suffer from some inherent problems such as underutilizing regional information of triple. In this work, we treat the RTE task based on table-filling method as an Object Detection task and propose a one-stage **O**bject **D**etection framework for **R**elational **T**riple **E**xtraction (**OD-RTE**). In this framework, the vertices-based bounding box detection, coupled with auxiliary global relational triple region detection, ensuring that regional information of triple could be fully utilized. Besides, our proposed decoding scheme could extract all types of triples. In addition, the negative sampling strategy of relations in the training stage improves the training efficiency while alleviating the imbalance of positive and negative relations. The experimental results show that 1) OD-RTE achieves the state-of-the-art performance on two widely used datasets (i.e., NYT and WebNLG). 2) Compared with the best performing table-filling method, OD-RTE achieves faster training and inference speed with lower GPU memory usage. To facilitate future research in this area, the codes are publicly available at `https://github.com/NingJinzhong/ODRTE`.

1 Introduction

The Relational Triple Extraction (RTE for short) aims to extract triples of the form (head, relation, tail) consisting of entity pairs and their relations from unstructured text, which is an important task of information extraction. In the early stage, traditional pipeline methods (Zelenko et al., 2003; Zhou et al., 2005; Chan and Roth, 2011) usually decompose the RTE task into two independent steps of named entity recognition and relation extraction.

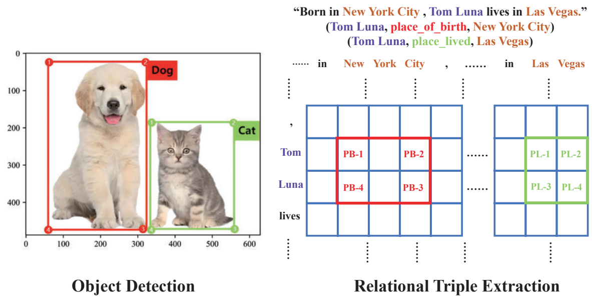


Figure 1: A Comparison of Object Detection and Relational Triple Extraction based on table-filling method. PB denotes relation "place_of_birth" and PL denotes relation "place_lived". Note that the table cells (representations of token pairs), the table, and the table regions occupied by the triples are aligned to the pixels, the image, and the objects on the visual side, respectively.

Although the pipelined approach is flexible, it ignores the correlation between the two tasks and suffers from error propagation (Ren et al., 2022).

To overcome this problem, some researchers try to use end-to-end joint entity and relation extraction models to solve the RTE task. These joint extraction models can be divided into four categories: tagging-based methods (Zheng et al., 2017; Wei et al., 2020; Zheng et al., 2021; Ren et al., 2022), table-filling methods (Wang et al., 2020; Ren et al., 2021; Shang et al., 2022a), text generation methods (Zeng et al., 2018, 2020; Ye et al., 2021) and graph-linking methods (Shang et al., 2022b). And the recently proposed table-filling method OneRel (Shang et al., 2022a) and the graph-linking method DirectRel (Shang et al., 2022b) achieve state-of-the-art performance and enable one-module and one-step extraction of relational triples.

Despite the promising success of existing joint methods, they suffer from the several problems[1]: (1) For table-filling methods, such as TPLinker (Wang et al., 2020), GRTE(Ren et al., 2021) and OneRel (Shang et al., 2022a), the triple's regional

* Corresponding author

[1]The specific explanations are shown in the Appendix.B

Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 11120–11135

information is insufficiently used during the extraction process. (2) Referring to OneRel, the current state-of-the-art method for table-filling, it fails to extract all types of triples. (3) Most existing methods, including OneRel and DirectRel, involve all relations in the training stage. The redundant relations involved in training will not only dominate the negative relation samples and make the model results more biased towards the negative relation samples, but also reduce the training efficiency.

Different from the above studies, we present a novel perspective on the task of relational triple extraction based on the table-filling method. We observed a high similarity between the table-filling-based RTE task and the object detection (OD) task in computer vision (CV). As shown in Figure 1, they all need to locate Regions of Interest (ROIs) in a two-dimensional array of pixels or token pairs. Further, inspired by the keypoint-based one-stage object detection methods (Duan et al., 2019; Law and Deng, 2018; Zhou et al., 2019), we propose a one-stage **O**bject **D**etection framework for **R**elational **T**riple **E**xtraction (short for **OD-RTE**) to address the relational triple detection problem.

Specifically, for the three problems in the existing methods mentioned above, the point-by-point solution of our proposed method is described as follows: (1) OD-RTE directly predicts the bounding boxes through identifying and grouping four vertices of each Region of Interest (shown in Figure 1). Vertices-based bounding box detection, coupled with global relational triple region detection, allows triple regional information to be better exploited compared to existing table-filling methods. (2) We propose the vertices-based relational triple encoding scheme and the Bidirectional Diagonal Walk (Bi-DW) decoding algorithm to ensure that OD-RTE has the ability to extract all types of triples. (3) During the training stage, we introduce a relation negative sampling strategy, which improves the training efficiency while alleviating the problem of imbalanced positive and negative relations in the full-relation training strategy.

The main contributions of this work can be summarized as follows:

- Treating the relational triple extraction task based on the table-filling method as an object detection task, we propose a one-stage triple extractor called OD-RTE. To the best of our knowledge, this is the first end-to-end RTE model based on the object detection framework.

- Following our perspective, we propose the vertices-based relational triple encoding method and an auxiliary global relational triple region detection task to make fuller use of the triple regional information. And we further propose the Bidirectional Diagonal Walk decoding algorithm, which enables the model to extract all types of triples.

- We introduce a relation negative sampling strategy in the training stage to improve the training efficiency while alleviating the problem of imbalanced positive and negative relations.

- We evaluate our model on two widely used public datasets and the results show that our model not only outperforms state-of-the-art baselines, but also achieves an improvement in computational efficiency.

2 Related Work

2.1 Relational triple extraction (RTE)

The RTE methods can be roughly divided into following four categories based on the idea of relational triple extraction. The first category is the tagging-based method, which utilizes several correlated sequence labeling modules to annotate head entities, tail entities and even relations. For example, the Novel Tagging Scheme proposed by Zheng et al. (2017) firstly converts the RTE task into a tagging problem. Then the CASREL proposed by Wei et al. (2020) models relations as functions that map subjects to objects in a sentence, which naturally handles the overlapping problem. And Zheng et al. (2021) proposed an extractor based on Potential Relation and Global Correspondence to alleviate the redundancy of relation prediction. BiRTE (Ren et al., 2022) proposes a bidirectional extraction framework based method that extracts triples based on the entity pairs extracted from two complementary directions. The second category is the table-filling method, which determines the head and tail entities by classifying the relationships of token pairs. The typical representation of the table-filling methods is TPLinker (Wang et al., 2020), which introduces a novel handshaking tagging scheme that aligns the boundary tokens of

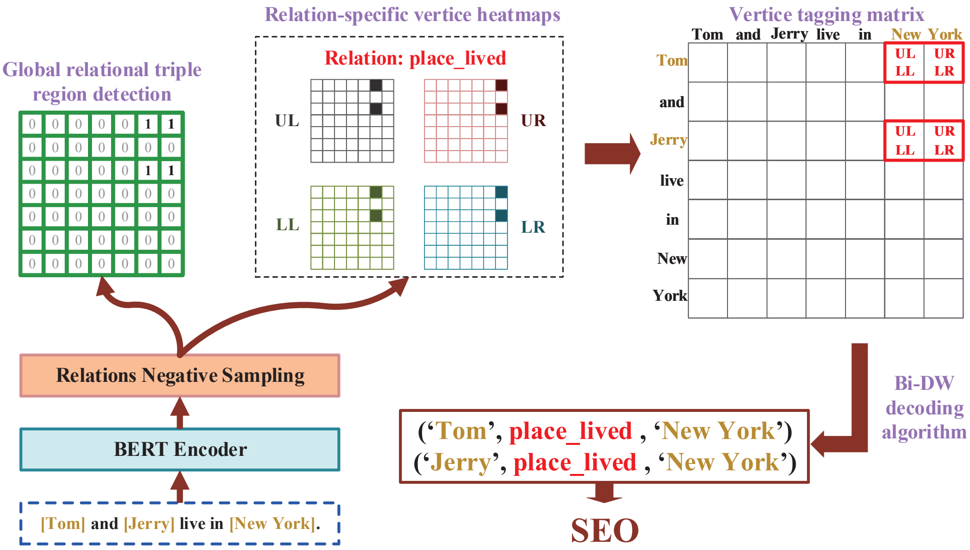


Figure 2: The overall architecture of OD-RTE. In this figure, the given input contains two SEO (Single Entity Overlap) triples. The UL, UR, LL and LR denote the upper left vertex, upper right vertex, lower left vertex and lower right vertex, respectively.

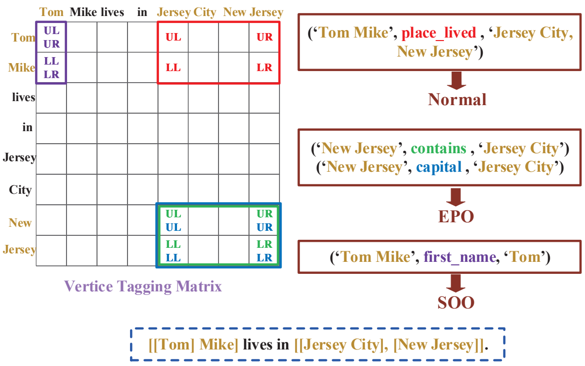


Figure 3: Examples of how OD-RTE handles normal, EPO (Entity Pair Overlap) and SOO (Subject Object Overlap) triples.

entity pairs under each relation type. And the recently proposed OneRel (Shang et al., 2022a) casts joint extraction as a fine-grained triple classification problem. The third category is the text generation methods (Zeng et al., 2018, 2020; Ye et al., 2021), which employ the seq2seq structure to generate the triples. And the fourth category is the graph-linking method (Shang et al., 2022b), which models the triple extraction problem as a bipartite graph linking problem of enumerated candidate entities.

2.2 Object Detection (OD)

The Object Detection aims to locate and identify objects of interest from natural images and is a fundamental but challenging task in Computer Vision. Two-stage object detectors such as R-CNN (Girshick et al., 2014), Faster-RCNN (Ren et al., 2015), Mask-RCNN (He et al., 2017) have achieved great success. Recently, one-stage OD models, such as YOLO (Redmon et al., 2016), SSD (Liu et al., 2016) and FCOS (Tian et al., 2019), have received much attention due to their excellent real-time performance. And our approach is also inspired by the keypoint-based one-stage object detection method (Duan et al., 2019; Law and Deng, 2018; Zhou et al., 2019). Shen et al. (2021) also proposed a two-stage detector which treats the nested named entity recognition task as an OD task. Different from their two-stage nested NER detector, our proposed OD-RTE is a one-stage detector, which achieves both performance and computational efficiency improvements on the RTE task.

3 Methodology

In this section, we first introduce the task definition. Then we detail the implementation of OD-RTE, whose overall structure of OD-RTE is shown in Figure 2.

3.1 Task Definition

Given a sentence $S = \{w_1, w_2, \cdots, w_L\}$, where L is the length of the sentence. The RTE task aims to extract the set of all potential relational triples $\Gamma = \{\Gamma_i \,|\Gamma_i = (h_i, r_i, t_i)\,, i = 1, \cdots, N\}$ from S, where $h_i, t_i \in E$, E is the set of all entities in S, $r_i \in R$ and $R = \{r_1, \cdots, r_K\}$ is the K predefined relations.

3.2 Regarding RTE as OD

3.2.1 OD-style relational triple tagging scheme

As can be seen from Figure 2, the head and tail entities belonging to the same triple enclose a rectangular area in the table composed of representations of token pairs. The existing work illustrates that entities in a triple can be determined by their bounding token (Wei et al., 2020) and that one-stage object detection can be achieved by identifying and grouping the key points of the bounding box (Duan et al., 2019; Law and Deng, 2018; Zhou et al., 2019). Inspired by these ideas, we propose to use the four vertices of the rectangular region enclosed by the head and tail entity of a triple in the relation-specific table to determine the 'object' region of the triple. Four vertices are used to determine its object region in the vertices tagging matrix: **(1) UL** is the upper left vertex of the object region, and also indicates the start position of both the head entity and the tail entity in the triple. **(2) UR** is the upper right vertex of the object region, and also indicates the start position of the head entity and the end position of the tail entity in the triple. **(3) LR** is the lower right vertex of the object region and also indicates the end position of both the head entity and the tail entity in the triple. **(4) LL** is the lower left vertex of the object region, which also indicates the end position of the head entity and the start position of the tail entity in the triple.

It is noted that when an entity in a triple contains only one token, a table cell may serve as multiple object area vertices at the same time. Here we take the triple ('Tom', place_lived , 'New York') in Figure 2 as an example, and the token pair 'Tom'-'New' is located at both the UL and LL vertices of the object region. Meanwhile, the token pair 'Tom'-'York' is located at both the UR and LR vertices of the object region. Moreover, it can be seen from Figure 2 and Figure 3 that our proposed object detection style tagging scheme can naturally cope with different entity overlapping patterns, such as EPO (Entity Pair Overlap), SEO (Single Entity Overlap) and SOO (Subject Object Overlap).

3.2.2 Relational triple region regressor

For a given sentence $S = \{w_1, w_2, \cdots, w_L\}$, we first use the pre-trained language model BERT (Devlin et al., 2019) to obtain the 768-dimensional token representations of the sentence:

$$\{h_1, h_2, \cdots, h_L\} = BERT\left(\{w_1, w_2, \cdots, w_L\}\right) \tag{1}$$

Relation negative sampling strategy: For the predefined relation set $R = \{r_1, \cdots, r_K\}$, we obtain the sampled relation set $\widetilde{R}$ by negative sampling of relations to alleviate the imbalance of positive and negative relations:

$$\widetilde{R} = \{\widetilde{r}_1, \widetilde{r}_2, \cdots, \widetilde{r}_{NS}\} = NegSample(R, NS) \tag{2}$$

where the operation $NegSample(R, NS)$ denotes retaining all positive relations in R while randomly sampling the negative relations and ensuring that the total number of positive and negative relations is NS.

Token pair representation: The token pair representation we used is similar to the existing table-filling method (Wang et al., 2020; Shang et al., 2022a). For the token pair (w_i, w_j), the representation of the token pair $h_{(w_i,w_j)}$ is computed as follows:

$$h_{(w_i,w_j)} = ReLU\left(W_{tp}\left[h_i; h_j\right] + b_{tp}\right) \tag{3}$$

where $1 \leqslant i, j \leqslant L$, $\mathrm{Re}LU(\cdot)$ is the ReLU (Agarap, 2018) activation function, $[\cdot;\cdot]$ is the concatenation operators, $W_{tp} \in \mathbb{R}^{d_e \times 1536}$ and $b_{tp} \in \mathbb{R}^{d_e}$ are learnable parameters.

Relation-specific vertice heatmaps: Then the probability scores of each token pair (w_i, w_j) for different vertices under the specific relation r_m are calculated as follows:

$$Score_{ijm}^{(UL)} = \sigma\left(W_{r_m} W_{UL} h_{(w_i,w_j)} + b_{r_m}^{UL}\right) \tag{4}$$

$$Score_{ijm}^{(UR)} = \sigma\left(W_{r_m} W_{UR} h_{(w_i,w_j)} + b_{r_m}^{UR}\right) \tag{5}$$

$$Score_{ijm}^{(LL)} = \sigma\left(W_{r_m} W_{LL} h_{(w_i,w_j)} + b_{r_m}^{LL}\right) \tag{6}$$

$$Score_{ijm}^{(LR)} = \sigma\left(W_{r_m} W_{LR} h_{(w_i,w_j)} + b_{r_m}^{LR}\right) \tag{7}$$

where $1 \leqslant m \leqslant NS$, $r_m \in \widetilde{R}$, $W_{UL}, W_{UR}, W_{LL}, W_{LR} \in \mathbb{R}^{d_e \times d_e}$, $W_{r_m} \in \mathbb{R}^{1 \times d_e}$ and $b_{r_m}^{UL}, b_{r_m}^{UR}, b_{r_m}^{LL}, b_{r_m}^{LR} \in \mathbb{R}$ are learnable parameters, σ denotes the sigmoid function, $Score_{ijm}^{(UL)}, Score_{ijm}^{(UR)}, Score_{ijm}^{(LL)}, Score_{ijm}^{(LR)} \in \mathbb{R}$ is the probability score indicating the probability that the token pair (w_i, w_j) is located at UL, UR, LL and LR, respectively. When the probability score about a vertex exceeds a threshold γ, the token pair is tagged as that vertex.

Global relational triple region detection: To more fully utilize the information of the triple region in the table, we introduce the global relational triple region detection as an auxiliary task. Each element of the global dependency matrix is calculated as follows:

$$P_{ij}^{(global)} = \sigma\left(W_{global} h_{(w_i,w_j)} + b_{global}\right) \tag{8}$$

where σ denotes the sigmoid function, $W_{global} \in \mathbb{R}^{1\times d_e}$ and $b_{global} \in \mathbb{R}$.

As shown in the green matrix in Figure 2, the ground truth of $P_{ij}^{(global)}$ is as follows:

$$GT_{ij}^{(global)} = \begin{cases} 1 & \text{if} \quad w_i \in span\left(e_i\right), w_j \in span\left(e_j\right) \\ 0 & \text{if} \quad else \end{cases} \tag{9}$$

where both e_i and e_j are entities in the entity set E, the symbol $\exists$ means that the token is within the span of the entity.

Loss Function: Based on the BCE (Binary Cross Entropy) loss, we design the objective function considering two subtasks: i.e., vertice tagging of token pairs and global triple region tagging. Correspondingly, the objective function of OD-RTE is defined as follows:

$$L_{vertice} = \frac{\sum_{V\in\Upsilon}\sum_{m=1}^{NS}\sum_{i=1}^{L}\sum_{j=1}^{L} BCE_{ijm}^{(V)}}{4 \times NS \times L \times L} \tag{10}$$

$$\begin{aligned} BCE_{ijm}^{(V)} &= GT_{ijm}^{(V)} \log\left(Score_{ijm}^{(V)}\right) \\ &+ \left(1 - GT_{ijm}^{(V)}\right)\log\left(1 - Score_{ijm}^{(V)}\right) \end{aligned} \tag{11}$$

$$L_{global} = \frac{\sum_{i=1}^{L}\sum_{j=1}^{L} BCE_{ij}^{(global)}}{L \times L} \tag{12}$$

$$\begin{aligned} BCE_{ij}^{(global)} &= GT_{ij}^{(global)} \log\left(P_{ij}^{(global)}\right) \\ &+ \left(1 - GT_{ij}^{(global)}\right)\log\left(1 - P_{ij}^{(global)}\right) \end{aligned} \tag{13}$$

$$L_{total} = L_{vertice} + \lambda L_{global} \tag{14}$$

where $\lambda \in \mathbb{R}$ is the loss function tuning factor which is set manually, $\Upsilon = \{UL, UR, LL, LR\}$ is a set containing the names of all vertices, $GT_{ijm}^{(V)}$ is the ground truth of $Score_{ijm}^{(V)}$. For example, if the token pair (w_i, w_j) is located as both 'UL' and 'LL' under relation r_m, then $GT_{ijm}^{(UL)} = 1$, $GT_{ijm}^{(LL)} = 1$, $GT_{ijm}^{(UR)} = 0$ and $GT_{ijm}^{(LR)} = 0$.

3.2.3 Decoding Algorithm

For each sentence, the tagging results of all token pairs for different vertices under the relation $r_n \in R(1 \leqslant n \leqslant K)$ are stored into the vertice tagging matrix $VT_n \in \mathbb{R}^{L\times L\times 4}$(as shown in figure 2). We propose the **B**idirectional **D**iagonal **W**alk (short for **Bi-DW**) Decoding Algorithm to easily decode the relational triples contained in each sentence from two diagonal directions of the object region. And the triples are decoded along two decoding directions: *decoding direction 1* (UL→UR→LR) and *decoding direction 2* (LR→LL→UL). Specifically, for the *decoding direction 1*, we first enumerate all token pairs located at the UL vertex, and then for each UL token pair search for the following nearest token pair located at the UR vertex. Next, for each UR token pair, we search for the following nearest token pair located at the LR vertex. As a result, the tokens between vertices UL and UR form the tail entity, and the tokens between vertices UR and LR form the head entity. Similarly, the meaning of *decoding direction 2* (LR→LL→UL) is similar to that of *decoding direction 1* (UL→UR→LR). Finally, the relational triples which are decoded by *decoding direction 1* and *decoding direction 2* will both be added to the final decoding results, which ensures all types of nested triples can be decoded. The Figure 6 in the Appendix illustrates the implementation of the Bi-DW Decoding Algorithm more specifically.

3.2.4 OD-RTE versus Existing tabling-filling methods

The similarities and differences between OD-RTE and existing tabling-filling methods are summarized as follows:

Similarity: The only similarity between OD-RTE and existing table-filling methods is that they both adopt the token pair representation as shown in Equation 3, which is also used in TPLinker (Wang et al., 2020) and OneRel (Shang et al., 2022a).

Differences: (1) In OD-RTE, we propose a new encoding scheme and decoding algorithm of relational triples based on the object detection framework. (2) Unlike existing table-filling methods (Wang et al., 2020; Ren et al., 2021; Shang et al., 2022a) that utilize only the head and tail information of entities, OD-RTE is, to our knowledge, the first table-filling method that introduces the information of the whole triple region. (3) As shown in Figure 2, OD-RTE uses different vertex heatmaps to tag the vertex labels of each token pair separately,

Model	NYT*			NYT			WebNLG*			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
CasRel$_{BERT}$ (Wei et al., 2020)	89.7	89.5	89.6	-	-	-	93.4	90.1	91.8	-	-	-
CasRel$_{random}$(Wei et al., 2020)	81.5	75.7	78.5	-	-	-	84.7	79.5	82.0	-	-	-
TPLinker (Wang et al., 2020)	91.3	92.5	91.9	91.4	92.6	92.0	91.8	92.0	91.9	88.9	84.5	86.7
PRGC$_{BERT}$ (Zheng et al., 2021)	93.3	91.9	92.6	93.5	91.9	92.7	94.0	92.1	93.0	89.9	87.2	88.5
PRGC$_{random}$(Zheng et al., 2021)	89.6	82.3	85.8	87.8	83.8	85.8	90.6	88.5	89.5	82.5	79.2	80.8
BiRTE (Ren et al., 2022)	92.2	93.8	93.0	91.9	**93.7**	92.8	93.2	94.0	93.6	89.0	89.5	89.3
GRTR$_{BERT}$ (Ren et al., 2021)	92.9	93.1	93.0	93.4	93.5	93.4	93.7	94.2	93.9	92.3	87.9	90.0
DirectRel (Shang et al., 2022b)	**93.7**	92.8	93.2	93.6	92.2	92.9	94.1	94.1	94.1	91.0	89.0	90.0
OneRel (Shang et al., 2022a)	92.8	92.9	92.8	93.2	92.6	92.9	94.1	94.4	94.3	91.8	90.3	91.0
OD-RTE$_{random}$	89.8	85.1	87.4	88.3	86.4	87.3	91.2	89.4	90.3	83.5	80.5	82.0
OD-RTE	93.5	**93.9**	**93.7**	**94.2**	93.6	**93.9**	**94.6**	**95.1**	**94.9**	**92.8**	**92.1**	**92.5**
-GRD	93.1	93.6	93.4	93.5	93.3	93.4	94.3	94.9	94.6	92.6	91.6	92.1
-RNS	93.7	93.5	93.6	94.0	93.1	93.5	94.4	93.9	94.1	92.5	90.7	91.6

Table 1: Precision (%), Recall (%) and F1-score (%) of the proposed OD-RTE method and baselines. The subscripted $BERT$ denotes using the pre-trained BERT encoder parameters and the subscripted $random$ denotes using the randomly initialized BERT encoder parameters. '-GRD' denotes OD-RTE without the global relational triple region detection module. '-RNS' denotes OD-RTE without the relation negative sampling strategy.

Model	NYT*									WebNLG*								
	Normal	EPO	SEO	SOO	N=1	N=2	N=3	N=4	N≥5	Normal	EPO	SEO	SOO	N=1	N=2	N=3	N=4	N≥5
TPLinker	90.1	94.0	93.4	90.1‡	90.0	92.8	93.1	96.1	90.0	87.9	95.3	92.5	86.0	88.0	90.1	94.6	93.3	91.6
PRGC	91.0	94.5	94.0	81.8	91.1	93.0	93.5	95.5	93.0	90.4	95.9	93.6	94.6	89.9	91.6	95.0	94.8	92.8
OneRel	90.6	95.1	94.8	90.8	90.5	93.4	93.9	96.5	94.2	91.9	95.4	94.7	94.9	91.4	93.0	95.9	95.7	94.5
DirectRel	**91.7**	94.8	94.6	90.0	**91.7**	**94.1**	93.5	96.3	92.7	92.0	**97.1**	94.5	94.6	**91.6**	92.2	**96.0**	95.0	94.9
OD-RTE	91.3	**95.9**	**95.7**	**91.4**	91.3	93.4	**94.6**	**96.9**	**95.3**	**92.1**	95.9	**95.4**	**95.4**	91.1	**93.5**	95.9	**96.1**	**95.1**

Table 2: F1-score (%) on sentences with different overlapping patterns and different triple numbers N. The symbol ‡ marks the results reported by PRGC (Zheng et al., 2021).

which makes it possible to assign multiple vertex labels to the same token pair. This allows OD-RTE to be more flexible in handling various types of nested triples. However, in the existing tabling-filling methods, a token pair will be assigned only one label under a relation. (4) To the best of our knowledge, OD-RTE is the first table-filling-based RTE method that introduces the relation negative sampling strategy instead of full relation training.

4 Experiments

4.1 Datasets

To provide a fair comparison with existing works (Wang et al., 2020; Shang et al., 2022a), we evaluate OD-RTE on two widely used benchmark datasets, i.e., NYT (Riedel et al., 2010) and WebNLG (Gardent et al., 2017). Both datasets contain two different versions: version 1 annotates the whole entity span and version 2 annotates only the last word of the entities. In this paper, the two datasets are denoted as NYT and WebNLG for version 1 and NYT* and WebNLG* for version 2. To further evaluate the performance of OD-RTE when facing different complex scenarios, we split the test set based on the number of triples and entity overlapping patterns in the sentence.

4.2 Experimental Settings

Consistent with all baseline models described in Appendix.A, we used three standard evaluation metrics, i.e., micro Precision (Prec.), Recall (Rec.) and F1 score (F1). Following the baseline models, for NYT* and WebNLG* , we use *Partial Matching*: a predicted triple is considered correct only if the relation and the last word of the head and tail entities are correct. And for NYT and WebNLG, *Exact Matching* is employed, i.e., the whole span of the head and tail entities in the extracted triples needs to be extracted completely.

All experiments are performed on a workstation equipped with i7-11700@2.50GHz, 32G memory and an RTX 3090 GPU. For pre-trained BERT, we use the cased base version of the English BERT published by Huggingface [2] and fine-tune it during training. The hyper-parameters are determined manually on the valid set using the grid search. And we used the Adam optimizer to train our model with a cosine annealing learning rate schedule for all the datasets and a learning rate of 5e-5. In particular, the batch size is set to 6 and 16 on datasets WebNLG/WebNLG* and NYT/NYT*, respectively. The representation dimension d_e of the token pairs

[2] https://huggingface.co/bert-base-cased

is 768×3 and the maximum sequence length is set to 100. The negative sampling number NS of the relations is set to 20 for all datasets. The loss function tuning factor λ is set to 0.01.

4.3 Results and Analysis

4.3.1 Main Results

We compare our proposed OD-RTE model with seven strong baseline models and the experimental results on all datasets are reported in Table 1. It can be seen that OD-RTE outperforms all baselines and achieves the state-of-the-art performance in terms of F1 scores on all datasets. From the experimental results, we can further observe that:

(1) Compared with the tagging-based methods (i.e., CasRel, PRGC and BiRTE), OD-RTE achieves a significant performance improvement. This indicates that the one-stage triple extraction method adopted in our proposed method can effectively alleviate the error propagation and exposure bias in the training stage compared with the tagging-based methods with multi-module cascade (Wang et al., 2020). In addition, the triple encoding and decoding strategy applied in OD-RTE can simultaneously decode triples that belong to the same relation and the head entities or tail entities are nested with each other, which cannot be handled by the tagging-based methods.

(2) When compared with the table-filling methods (i.e., TPLinker, GRTR and OneRel), OD-RTE still has superior performance. This verifies our claim that the triple determination manner in OD-RTE based on the four vertices of the object region can make better use of the regional information for triples.

(3) To verify the effectiveness and robustness of OD-RTE decoder, we abandon the pre-trained weights to randomly initialize the BERT. The experimental results show that OD-RTE$_{random}$ still outperforms CasRel$_{random}$ and PRGC$_{random}$ even without the beneficiation from pre-trained BERT. This indicates that the performance improvement brought by OD-RTE comes not only from the pre-trained BERT but also from its decoder itself.

(4) We can observe that without the global relational triple region detection module, the performance of OD-RTE decreases noticeably. This shows that except for the vertices' information, the whole region information of the triple also has a positive impact on the performance of OD-RTE.

(5) The results also show that the performance of the model obviously degrades without the relation negative sampling strategy and the detailed analysis is described in Section 4.3.4.

4.3.2 Analysis on Different Sentence Types

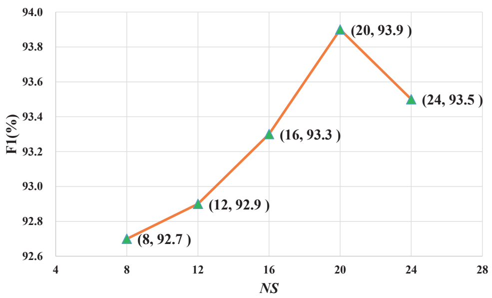
To verify the ability of OD-RTE to handle different overlapping patterns and multiple triples in a sentence, we conduct the corresponding experiments on NYT* and WebNLG*. The four state-of-the-art models are selected as the baselines and the detailed experimental results are shown in Tabel 2. It can be seen that our proposed OD-RTE model achieves the state-of-the-art performance on 12 out of 18 subsets of the two test sets, especially in multiple triples ($N > 3$) and two complex overlapping patterns (SEO and SOO) scenarios. In the table-filling method, the table consisting of token pairs under each relation is severely sparse, i.e., the positive and negative examples of token pairs are grossly imbalanced. The table composed of token pairs in OD-RTE has more non-zero elements in the scenarios of SEO patterns, SOO patterns and multiple triples, which alleviates the imbalance of token pairs of positive and negative examples to a certain extent. As a result, OD-RTE learns more adequate positive sample information in several complex scenarios mentioned above and thus achieves the competitive performance.

4.3.3 Analysis on Model Efficiency

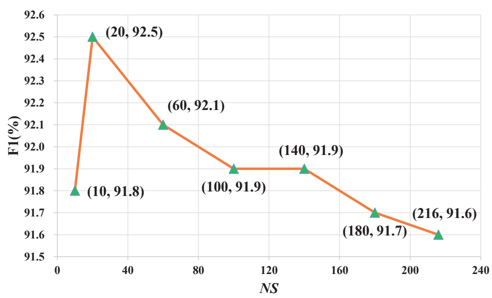
Compared with other table-filling methods, we evaluate the efficiency of OD-RTE from three aspects, i.e., Training Time, Inference Time and GPU Memory. The experimental results are shown in Table 3. For fair comparison, the experimental results of the above three efficiency evaluation metrics are all obtained with the same parameter settings. From the experimental results, we can observe that OD-RTE outperforms TPLinker, GRTE and OneRel while using the least single-epoch training time which illustrates the efficiency and strong learning ability of OD-RTE. Additionally, OD-RTE achieves the fastest inference speed, which illustrates the high efficiency of our proposed Bi-DW decoding algorithm. Compared with the state-of-the-art table-filling method OneRel, OD-RTE still achieves a performance improvement while reducing the GPU memory usage by about 2/3 during the training phase. OD-RTE does not rely on large memory GPUs, which illustrates the economics of training it. Although OD-RTE occupies slightly more GPU memory than TPLinker, it brings signifi-

Dataset	Model	Training Time(S)	Inference Time(S)	GPU Memory(MB)	F1(%)
NYT	TPLinker	$1601^{\#\ddagger}$	$45^{\#\ddagger}$	**6014**$^{\#\ddagger}$	92.0
	GRTE	$931^{\#\ddagger}$	$43^{\#\ddagger}$	$18771^{\#\ddagger}$	93.4
	OneRel	$1203^{\#\ddagger}$	$42^{\#\ddagger}$	$23703^{\#\ddagger}$	92.9
	OD-RTE$_{wns}$	$825^{\#}$	**38**$^{\#}$	$8389^{\#}$	93.5
	OD-RTE	**798**$^{\#}$	**38**$^{\#}$	$8372^{\#}$	**93.9**
WebNLG	TPLinker	$602^{\#\ddagger}$	$13^{\#\ddagger}$	**5951**$^{\#\ddagger}$	86.7
	GRTE	$118^{\#\ddagger}$	$13^{\#\ddagger}$	$15345^{\#\ddagger}$	90.0
	OneRel	$89^{\#\ddagger}$	$11^{\#\ddagger}$	$21338^{\#\ddagger}$	91.0
	OD-RTE$_{wns}$	$78^{\#}$	**9**$^{\#}$	$8781^{\#}$	91.6
	OD-RTE	**70**$^{\#}$	**9**$^{\#}$	$7515^{\#}$	**92.5**

Table 3: Comparison of the efficiency with other table-filling methods. Training Time (S) means the time taken to train an epoch. The Inference Time (S) denotes the time required to predict the triples of sentences in the whole test set. GPU Memory (MB) represents the maximum amount of GPU memory that the model occupies during the training phase. The superscript $\#$ indicates that the result is obtained on the NYT/WebNLG dataset with batch size 8/6 and the superscript $\ddagger$ marks results obtained by official implementations. OD-RTE$_{wns}$ is a version of OD-RTE without relation negative sampling strategy.



(a) NYT



(b) WebNLG

Figure 4: The effect of the relation negative sampling number NS on datasets NYT and WebNLG.

cant performance improvement, therefore choosing OD-RTE is still cost-effective. In addition, OD-RTE surpasses OD-RTE$_{wns}$ in performance and efficiency metrics other than inference time, indicating that relation negative sampling strategy can not only improve the model efficiency, but also help improve the model performance. Compared with NYT, because of the large number of relations, the improvement brought by the negative sampling strategy of relations is more significant in WebNLG.

4.3.4 Analysis on Relation Negative Sampling

We also conduct experiments to verify the effect of the relation negative sampling number NS on the OD-RTE performance. Since both NYT and WebNLG versions of the dataset contain the complete span of entities, the performance of the model on these two datasets can more intuitively reflect the RTE performance in real scenarios. Therefore, NYT and WebNLG are selected as the experimental datasets, and the experimental results are shown in Figure 4. An interesting observation from the experimental results is that OD-RTE achieves the best performance on both NYT and WebNLG datasets when $NS = 20$. When NS is large than 20, the F1 score performance of OD-RTE shows a decreasing trend as NS increased. This validates our motivation that the positive and negative relations in the training data become imbalanced with the increase of NS. At this time, the training of the model will be dominated by negative relations and the inference results of the model will be more biased towards negative relations. Meanwhile, when NS is less than 20, the performance of the model decreases as NS decreases. This suggests that when the number of negative relations in the training data is too small, the model will be under-trained causing performance degradation. In addition, it can be seen that OD-RTE still outperforms the state-of-the-art table-filling method OneRel when no relational negative sampling is employed during the training stage, i.e., when $NS = 24$ on NYT and $NS = 216$ on WebNLG. This illustrates that our proposed model can utilize the regional information of triple more effectively than the existing table-filling methods do.

5 Conclusions

In this work, we treat the RTE task as an object detection task from a novel perspective and propose a one-stage relational triple extraction model OD-RTE. The vertices-based relational triple encoding method and Bi-DW decoding algorithm used in OD-RTE enable it to handle various complex entity scenarios. In addition, the negative sampling strategy of relations in the training stage improves the training efficiency while alleviating the imbalance of positive and negative relations. Compared with existing table-filling methods, our proposed OD-RTE can more effectively utilize the regional information for triple. Experimental results on public datasets demonstrate that OD-RTE not only outperforms other state-of-the-art models in multiple complex scenarios, but also has high computational efficiency.

Limitations

In this section, we would like to discuss two limitations of OD-RTE as follows:

(1) In the current table-filling based RTE methods including OD-RTE, the issue of sparse labels in the tables still exists. As cells of the table, the number of positive and negative token pairs is grossly unbalanced. In this work, although we alleviate the problem of unbalanced positive and negative relations by introducing the relation negative sampling strategy, the problem of unbalanced positive and negative token pairs still exists and needs to be addressed. We will try to mitigate the problem in our future work.

(2) Currently, OD-RTE can only be applied to the relational triple extraction task. In recent years, the table-filling-based approaches have been widely used for many information extraction tasks besides the RTE task, such as opinion mining (Wu et al., 2020) and named entity recognition (Li et al., 2022). Therefore, in future work, we will try to extend the object detection framework to other information extraction tasks to let the model make full use of the information of entity boundaries.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No.62276043 and No.62076046) and the Fundamental Research Funds for the Central Universities (No. DUT22ZD205).

References

Abien Fred Agarap. 2018. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*.

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*, pages 551–560.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL*, pages 4171–4186.

Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. 2019. Centernet: Keypoint triplets for object detection. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6568–6577. IEEE Computer Society.

Claire Gardent, Anastasia Shimorina, Shashi Narayan, and Laura Perez-Beltrachini. 2017. Creating training corpora for nlg micro-planning. In *55th annual meeting of the Association for Computational Linguistics (ACL)*.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. 2014. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587.

Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969.

Hei Law and Jia Deng. 2018. Cornernet: Detecting objects as paired keypoints. In *Proceedings of the European conference on computer vision (ECCV)*, pages 734–750.

Jingye Li, Hao Fei, Jiang Liu, Shengqiong Wu, Meishan Zhang, Chong Teng, Donghong Ji, and Fei Li. 2022. Unified named entity recognition as word-word relation classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 10965–10973.

Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer.

Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788. IEEE.

Feiliang Ren, Longhui Zhang, Shujuan Yin, Xiaofeng Zhao, Shilei Liu, Bochao Li, and Yaduo Liu. 2021.

A novel global feature-oriented relational triple extraction model based on table filling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2646–2656.

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 *WSDM '22: The Fifteenth ACM International Conference on Web Search and Data Mining, Virtual Event / Tempe, AZ, USA, February 21 - 25, 2022*, pages 824–832.

Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28.

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*, pages 148–163. Springer.

Yu-Ming Shang, Heyan Huang, and Xianling Mao. 2022a. Onerel: Joint entity and relation extraction with one module in one step. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 11285–11293.

Yu-Ming Shang, Heyan Huang, Xin Sun, Wei Wei, and Xian-Ling Mao. 2022b. Relational triple extraction: One step is enough. *arXiv preprint arXiv:2205.05270*.

Yongliang Shen, Xinyin Ma, Zeqi Tan, Shuai Zhang, Wen Wang, and Weiming Lu. 2021. Locate and label: A two-stage identifier for nested named entity recognition. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2782–2794.

Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. 2019. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636.

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*, pages 1572–1582.

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*, pages 1476–1488.

Zhen Wu, Chengcan Ying, Fei Zhao, Zhifang Fan, Xinyu Dai, and Rui Xia. 2020. Grid tagging scheme for aspect-oriented fine-grained opinion extraction. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2576–2585.

Hongbin Ye, Ningyu Zhang, Shumin Deng, Mosha Chen, Chuanqi Tan, Fei Huang, and Huajun Chen. 2021. Contrastive triple extraction with generative transformer. In *Proceedings of the AAAI conference on artificial intelligence*, pages 14257–14265.

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

Daojian Zeng, Haoran Zhang, and Qianying Liu. 2020. Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. In *Proceedings of the AAAI conference on artificial intelligence*, pages 9507–9514.

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)*, pages 506–514.

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 (Volume 1: Long Papers)*, pages 6225–6235.

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)*, pages 1227–1236.

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)*, pages 427–434.

Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. 2019. Objects as points. *arXiv preprint arXiv:1904.07850*.

A Other experimental settings

The other experimental settings we used are described in detail as follows:

Statistical information of datasets: The statistical information of the two datasets is shown in Table 4.

Baselines: We compare OD-RTE with seven state-of-the-art baseline models, including CasRel

Dataset	Relations	Sentences			Details of test set									
		Train	Valid	Test	Normal	SEO	EPO	SOO	N=1	N=2	N=3	N=4	N>5	Triples
NYT	24	56195	5000	5000	3222	1273	969	117	3240	1047	314	290	109	8120
WebNLG	216	5019	500	703	239	448	6	85	256	175	138	93	41	1607
NYT*	24	56195	4999	5000	3266	1297	978	45	3244	1045	312	291	108	8110
WebNLG*	171	5019	500	703	245	457	26	84	266	171	131	90	45	1591

Table 4: Statistics of datasets used in our experiments. N is the number of triples in a sentence.

Hyper-parameter	**Range**
Batch size	[4,6,8,16,32]
Init learning rate	[5e-6,e-5,3e-5,5e-5]
Representation dimension d_e	[768,2*768,3*768]
NS for NYT/NYT*	[8,12,16,20,24]
NS for WebNLG/WebNLG*	[10,20,60,100,140,180,216]
λ	[1,0.1,0.01,0.0001]

Table 5: Hyperparameters and its search range.

Model	Sub Task	NYT*			WebNLG*		
		Prec.	Rec.	F1	Prec.	Rec.	F1
PRGC	(h,t)	94.0	92.3	93.1	96.0	93.4	94.7
	r	95.3	96.3	95.8	92.8	96.2	94.5
	(h,r,t)	93.3	91.9	92.6	94.0	92.1	93.0
OneRel	(h,t)	93.3	93.4	93.3	**96.2**	96.5	96.3
	r	96.7	96.9	96.8	96.7	97.0	96.8
	(h,r,t)	92.8	92.9	92.8	94.1	94.4	94.3
DirectRel	(h,t)	**94.1**	93.2	93.7	95.8	95.9	95.8
	r	**97.3**	96.4	96.9	**96.8**	96.7	96.7
	(h,r,t)	**93.7**	92.8	93.2	94.1	94.1	94.1
OD-RTE	(h,t)	93.7	**94.2**	**94.0**	95.9	**97.3**	**96.6**
	r	96.7	**97.2**	**97.0**	96.5	**97.2**	**96.8**
	(h,r,t)	93.5	**93.9**	**93.7**	**94.6**	**95.1**	**94.9**

Table 6: Experimental results of different subtasks on the NYT* and WebNLG* datasets. (h,t) denotes entity extraction, r denotes relation classification, and (h,r,t) denotes relational triple extraction.

(Wei et al., 2020), TPLinker (Wang et al., 2020), PRGC (Zheng et al., 2021), BiRTE (Ren et al., 2022), GRTE (Ren et al., 2021), DirectRel (Shang et al., 2022b) and OneRel (Shang et al., 2022a).

Determination of hyperparameters: We determine the hyperparameters manually based on the performance of the model on the valid set. The search range of hyperparameters is shown in Table 5.

B Supplementary description of the Introduction section

As for the issues with the existing methods discussed in the introduction of this paper, we supplement the first two issues with a more detailed description as follows:

(1) Regarding the problem that the existing table-filling-based RTE methods make insufficient use of the regional information of the triple, we illustrate it specifically using Figure 5. As can be seen in Figure 5(a), a triple occupies a rectangular region in the table. However, existing table-filling methods, such as TPLinker in Figure 5(b), GRTE in Figure 5(c) and OneRel in Figure 5(d), utilize only part of the table cells associated with the entity head and tail information to tag the triples. This leaves the triple region information in the table underexploited. As shown in Figures 5(e) and 5(f), OD-RTE can exploit all the table cells in the region occupied by the triple.

(2) With respect to the problem that the state-of-the-art table-filling method, OneRel, cannot extract all types of triples, we illustrate it in detail with two cases. The first case is shown in Figure 6. It can be seen that OneRel cannot simultaneously decode both of the two triples that are nested in either the head entity or the tail entity under the same relation. The relational triple encoding scheme in OD-RTE with the Bi-DW decoding algorithm can handle this situation. The second case is shown in Figure 7. It can be seen that if there is a single token entity in the relational triple, the triple encoding scheme of original OneRel cannot encode it validly. In triple encoding scheme of improved

Decoding Methods	NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1
S-DW	94.2	92.5	93.3	92.8	91.3	92.0
RVW	**94.5**	92.4	93.4	93.0	90.5	91.7
Bi-DW$_\cap$	94.4	92.6	93.5	**93.0**	90.6	91.8
Bi-DW	94.2	**93.6**	**93.9**	92.8	**92.1**	**92.5**

Table 7: Precision (%), Recall (%) and F1-score (%) of different decoding algorithms. S-DW denotes for single direction diagonal walk method, which only takes the decoding result of decoding *decoding direction 1* (UL→UR→LR). The decoding strategy applied in the S-DW decoding method is also adopted in OneRel (Shang et al., 2022a). RVW denotes the rectangle vertex walk method, which collects relational triples along the new decoding direction (UL→UR→LR→LL). In the Bi-DW$_\cap$, only the relational triples that appear in the decoding results of both *decoding direction 1* and *decoding direction 2* will be inserted into the final decoding results.

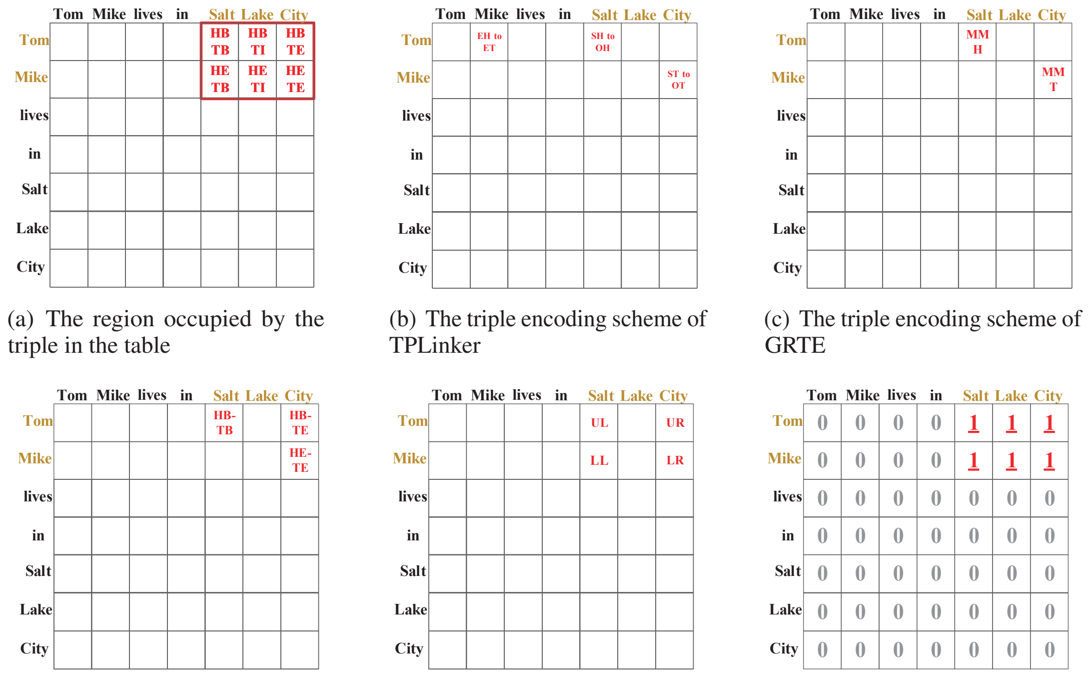
OneRel from the official implementation, the [unu] character is inserted after each token to achieve a valid encoding of the single token entity in the triple. However, this not only doubles the sentence length leading to a serious reduction in the computational efficiency of the model, but also makes the text sequences no longer natural language sequences affecting the fine-tuning performance of the pre-trained BERT. But our proposed triple encoding scheme of OD-RTE can efficiently encode relational triples containing single token entities.

C Results on Different Sub-tasks

To analyze the advantages of OD-RTE in relational triple extraction process in detail, we conduct experiments on two sub-tasks i.e., entity extraction and relation classification. The experimental results are shown in Table 6, OD-RTE outperforms all baseline methods in F1 score on each subtask. The two subtasks of entity extraction and relation classification in OD-RTE are jointly performed in the same module in one stage, which allows the two subtasks to interact while avoiding the accumulation of errors caused by the cascade structure. At the same time, compared with the state-of-the-art table-filling method OneRel, OD-RTE can still achieve a performance improvement, indicating that it can more fully utilize the regional information for triple.

D Analysis on Different Decoding Methods

We conduct experiments to verify the performance of different decoding methods. The results are reported in Table 7. It can be seen that our proposed Bi-DW decoding algorithm achieves the highest recall rate while maintaining a high precision rate. In addition, it should be noted that the Bi-DW algorithm can simultaneously decode the triples of nested head entities or tail entities under the same relation (shown in Figure 6). However, other decoding algorithms shown in Table 7, as well as some state-of-the-art baseline models, such as PRGC, OneRel, do not have this capability. This suggests that the Bi-DW decoding algorithm is not only efficient but also improves the performance.



(a) The region occupied by the triple in the table

(b) The triple encoding scheme of TPLinker

(c) The triple encoding scheme of GRTE

(d) The triple encoding scheme of OneRel

(e) The triple encoding scheme of OD-RTE

(f) Dependency matrix of the global triple region of OD-RTE

Figure 5: Comparison of different triple encoding schemes for table-filling-based RTE methods. For clarity of expression, we employ the same tagging notation as in the original paper. The example sentence "Tom Mike lives in Salt Lake City. " in this figure contains a relational triple ('Tom Mike', place_lived, 'Salt Lake City').

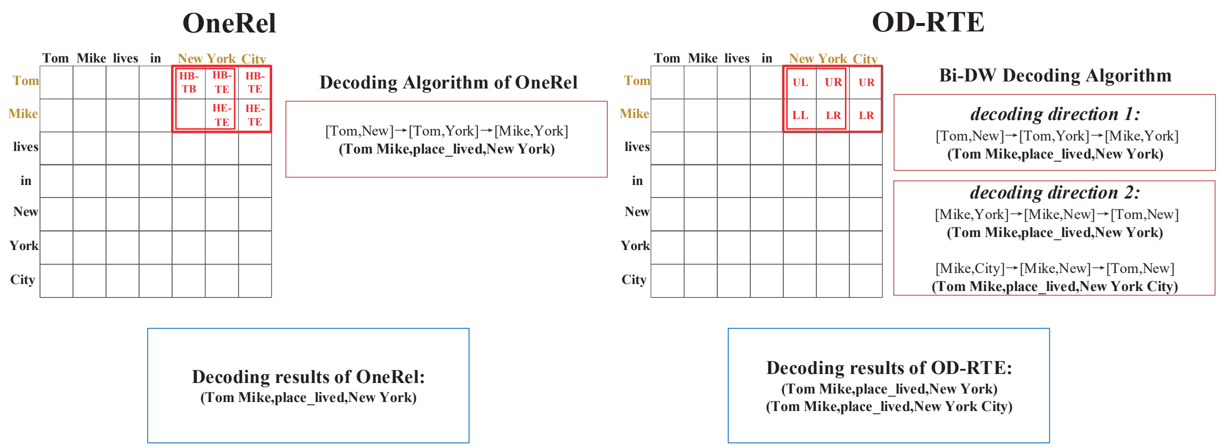


Figure 6: Comparison of OneRel and OD-RTE dealing with nested triples under the same relation. The example sentence "Tom Mike lives in New York City. " in this figure contains two relational triples ('Tom Mike', place_lived, 'New York') and ('Tom Mike', place_lived, 'New York City'). The two relational triples share the same head entity 'Tom Mike'. And the tail entities 'New York' and 'New York City' of the two relational triples are nested. The two tokens in square brackets represent a token pair.

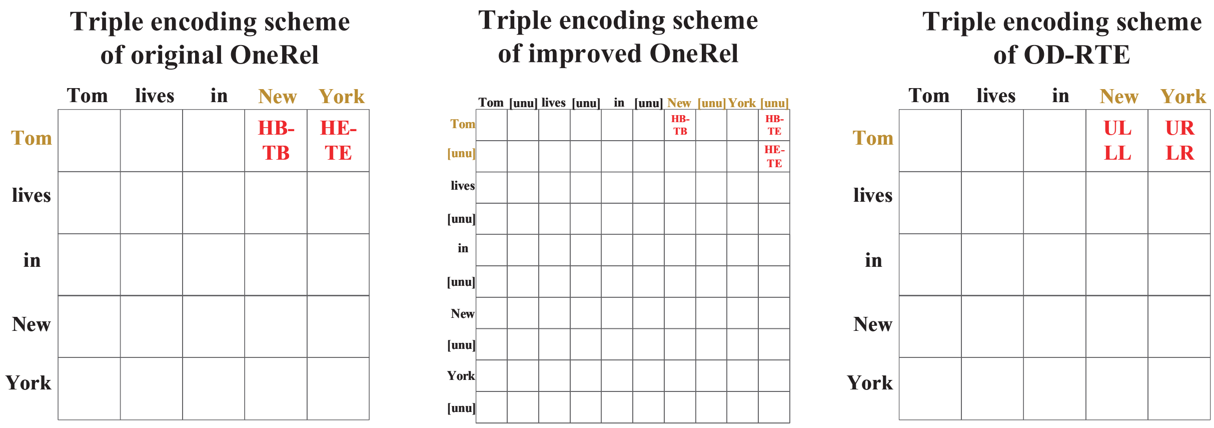


Figure 7: Comparison of OneRel and OD-RTE dealing with triples containing single token entities. The example sentence "Tom lives in New York. " in this figure contains a relational triple ('Tom', place_lived, 'New York'). The symbol [usu] indicates the abbreviation of the unused token [unused1] in the vocabulary list of the BERT tokenizer. 'Triple encoding scheme of original OneRel' is described in the original paper for OneRel (Shang et al., 2022a). 'Triple encoding scheme of improved OneRel' comes from the official implementation of OneRel.

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
section before Acknowledgements

☐ A2. Did you discuss any potential risks of your work?
Not applicable. We used two publicly available datasets. And numerous scholars have worked on this dataset before us.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and 1 Introduction

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

3 Methodology

☑ B1. Did you cite the creators of artifacts you used?
3 Methodology

☒ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
All the artifacts we use are open source and publicly available.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
3 Methodology

☒ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
The datasets we use are produced and published in a process that meets these requirements.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
4.1 Datasets

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
4.1 Datasets

C ☑ Did you run computational experiments?

4.1 Datasets

☐ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
No response.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
4.1 Datasets

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
4.1 Datasets

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
4 Experiments and Appendix.A

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.