A Novel Global Feature-Oriented Relational Triple Extraction Model based on Table Filling

Feiliang Ren[†,*], Longhui Zhang[†], Shujuan Yin, Xiaofeng Zhao, Shilei Liu
Bochao Li, Yaduo Liu
School of Computer Science and Engineering
Key Laboratory of Medical Image Computing of Ministry of Education
Northeastern University, Shenyang, 110169, China
renfeiliang@cse.neu.edu.cn

EMNLP 2021
Code: https://github.com/neukg/GRTE

Reported by Nengqiang Xiang

Introduction

Existing methods fill relation tables mainly based on local features that are extracted from either a single token pair or the filled history of some limited token pairs, but ignore following two kinds of valuable global features: the global associations of token pairs and of relations.

To overcome this deficiency, we propose a global feature-oriented triple extraction model that makes full use of the mentioned two kinds of global associations.

sentence	*Edward Thomas and John are from New York City,USA*
Triplets	1. *Edward Thomas, live_in, New York* 2. *John, live_in, USA* 3. *New York, located_in, USA*

Introduction

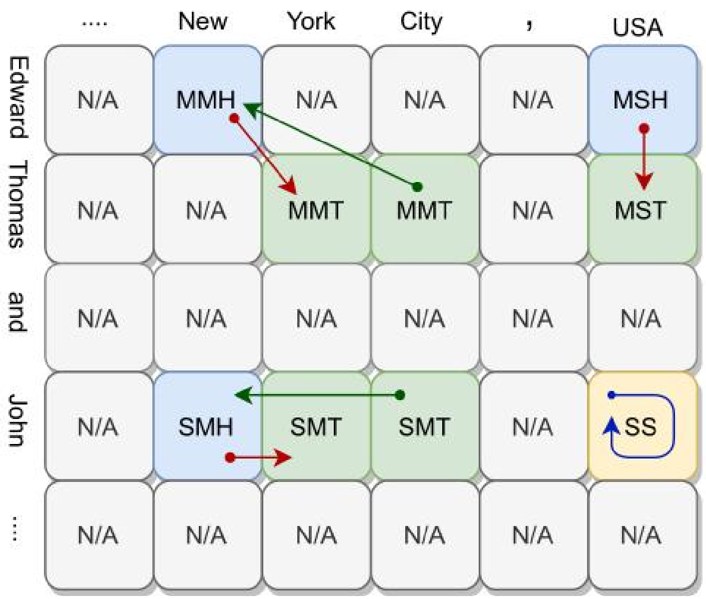


Figure 1: Examples of table filling and decoding strategy. Arrows with different colors correspond to different search routes defined in *Algorithm* 1.

Given a sentence $S = w_1 w_2 \dots w_n$, we will maintain a table $table_r$ (the size is $n \times n$) for each relation r ($r \in R$, and R is the relation set).

For a token pair indexed by the i-th row and the j-th column, we denote it as (w_i, w_j) and denote its label as l.

L = {"N/A", "MMH", "MMT", "MSH", "MST", "SMH", "SMT", "SS"}

(Edward Thomas, New York City), (Edward Thomas, New York), (Edward Thomas, USA), (John, New York City), (John, New York) , (John, USA)

Method

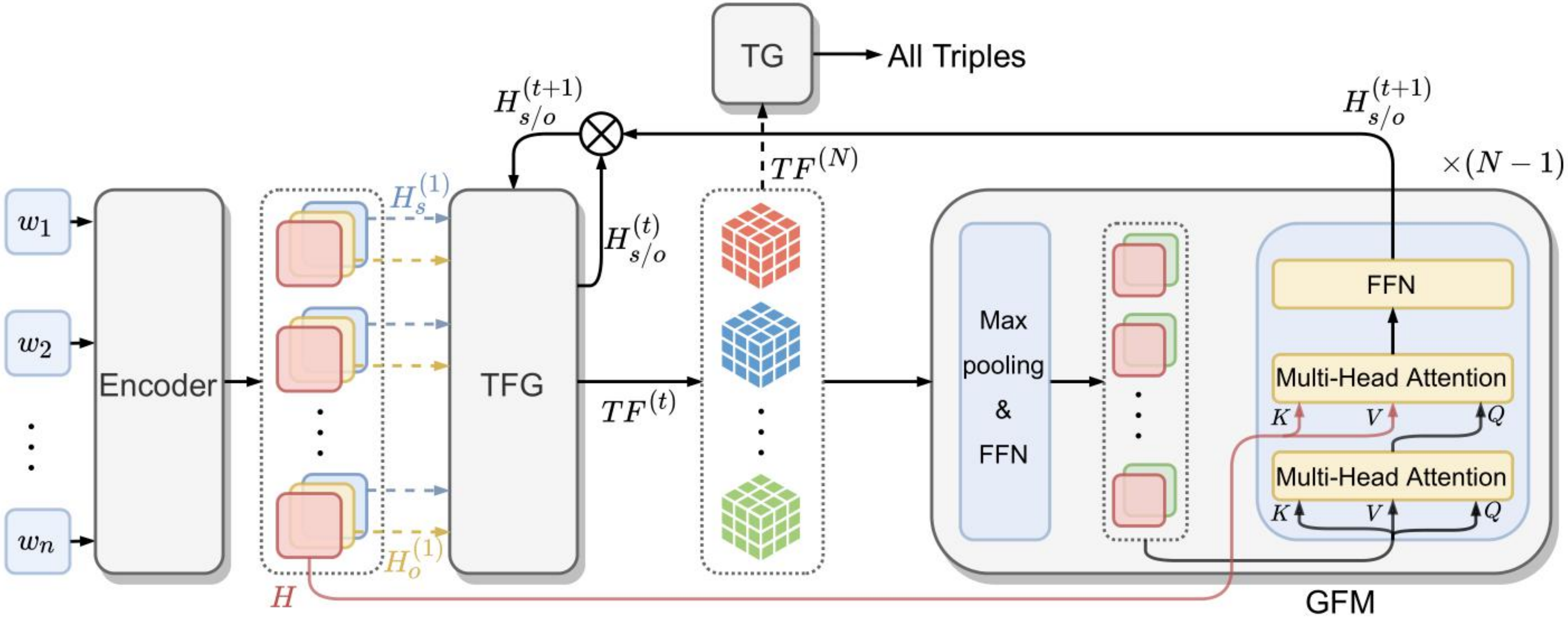
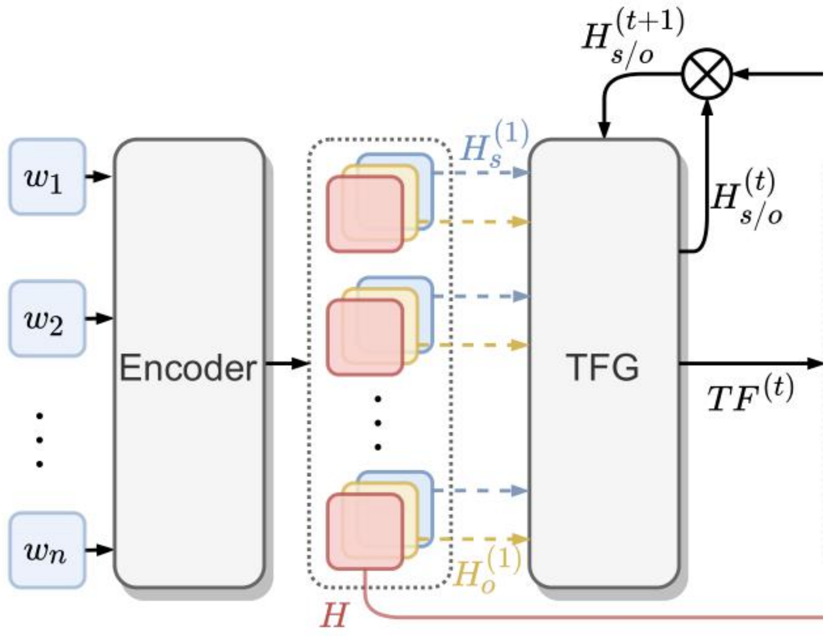


Figure 2: Model Architecture. The dotted arrows to *TFG* means that $H_s^{(1)}$ and $H_o^{(1)}$ will be inputted to *TFG* only at the first iteration. The dotted arrow to *TG* means that $TF^{(N)}$ will be inputted into *TG* only at the last iteration.

Method



Encoder Module：

$$H = BERT(sentence)$$

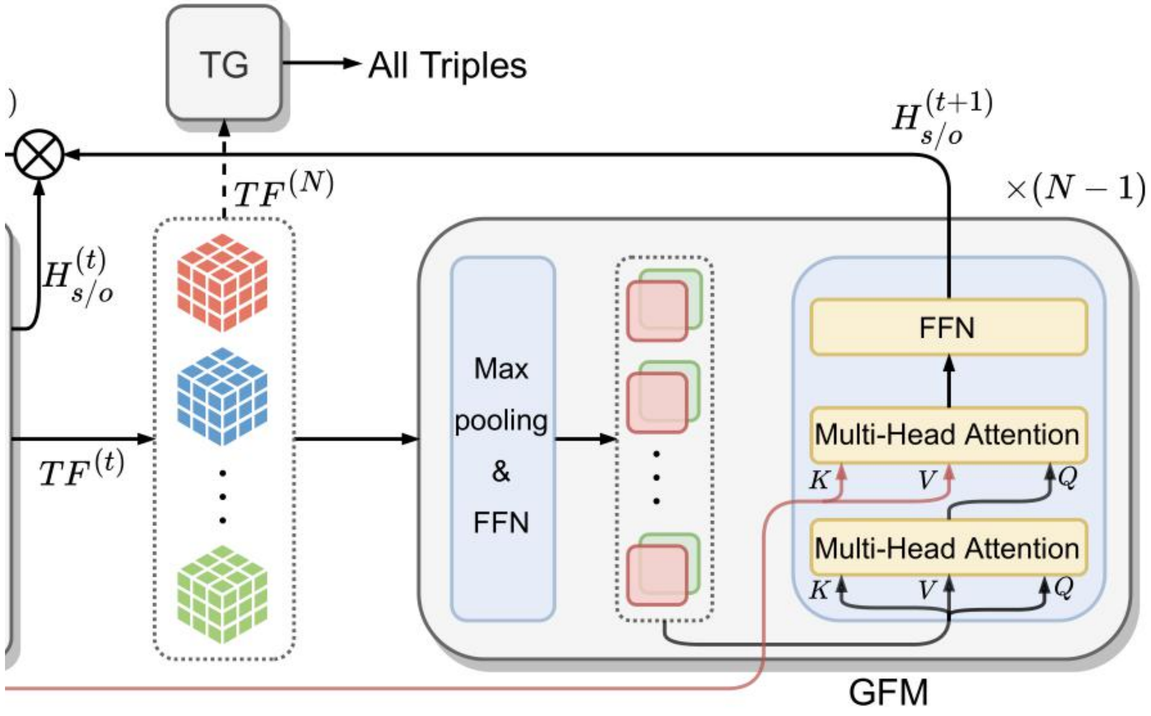
$$H_s^{(1)} = W_1 H + b_1 \qquad (1)$$

$$H_o^{(1)} = W_2 H + b_2$$

TFG Module：

$$TF_r^{(t)}(i,j) = W_r \, \mathrm{ReLU}(H_{s,i}^{(t)} \circ H_{o,j}^{(t)}) + b_r \qquad (2)$$

Method



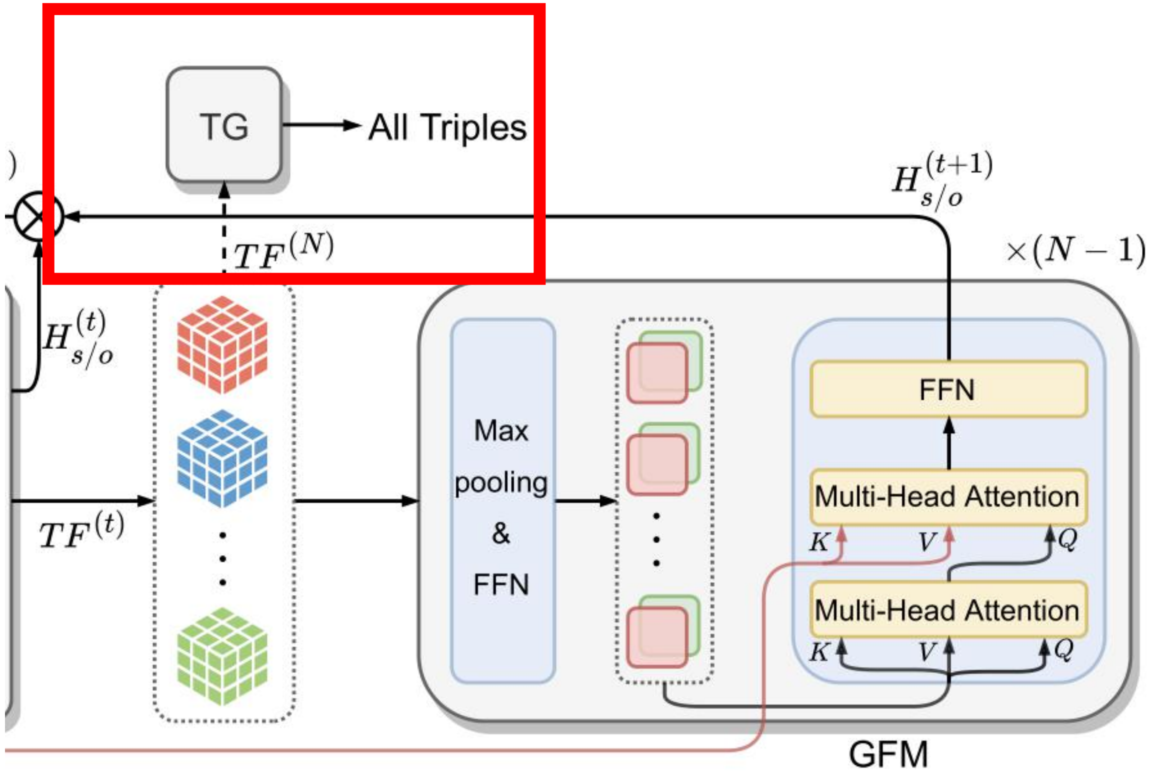
GFM Module：

$$TF_s^{(t)} = W_s \underset{s}{\text{maxpool}} \left(TF^{(t)}\right) + b_s$$
$$TF_o^{(t)} = W_o \underset{o}{\text{maxpool}} \left(TF^{(t)}\right) + b_o \qquad (3)$$

$$\hat{TF}_{s/o}^{(t)} = \text{MultiHeadSelfAtt}(TF_{s/o}^{(t)})$$
$$\hat{H}_{(s/o)}^{(t+1)} = \text{MultiHeadAtt}(\hat{TF}_{s/o}^{(t)}, H, H) \qquad (4)$$
$$H_{(s/o)}^{(t+1)} = \text{ReLU}(\hat{H}_{(s/o)}^{(t+1)} W + b)$$

$$H_{(s/o)}^{(t+1)} = \text{LayerNorm}\left(H_{(s/o)}^{(t)} + H_{(s/o)}^{(t+1)}\right) \qquad (5)$$

Method



TG Module：

$$\hat{table}_r(i,j) = \operatorname{softmax}\left(TF_r^{(N)}(i,j)\right)$$
$$table_r(i,j) = \operatorname*{argmax}_{l \in L}\left(\hat{table}_r(i,j)[l]\right) \tag{6}$$

Loss Function：

$$\mathcal{L} = \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{|R|} -\log p\left(y_{r,(i,j)} = table_r(i,j)\right)$$
$$= \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{r=1}^{|R|} -\log \hat{table}_r(i,j)[y_{r,(i,j)}] \tag{7}$$

Experiments

Category	NYT29		NYT24		WebNLG	
	Train	Test	Train	Test	Train	Test
Normal	53444	2963	37013	3266	1596	246
EPO	8379	898	9782	978	227	26
SEO	9862	1043	14735	1297	3406	457
ALL	63306	4006	56195	5000	5019	703
Relation	29		24		216 / 171*	

Table 1: Statistics of datasets. *EPO* and *SEO* refer to *entity pair overlapping* and *single entity overlapping* respectively (Zeng et al., 2018). Note that a sentence can belong to both *EPO* and *SEO*. And 216 / 171* means that there are 216 / 171 relations in WebNLG and WebNLG* respectively.

Experiments

Model	NYT29			NYT24*			NYT24			WebNLG*			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
CopyRE	–	–	–	61.0	56.6	58.7	–	–	–	37.7	36.4	37.1	–	–	–
GraphRel	–	–	–	63.9	60.0	61.9	–	–	–	44.7	41.1	42.9	–	–	–
OrderCopyRE	–	–	–	77.9	67.2	72.1	–	–	–	63.3	59.9	61.6	–	–	–
ETL-Span	74.5*	57.9*	65.2*	84.9	72.3	78.1	85.5	71.7	78.0	84.0	91.5	87.6	84.3	82.0	83.1
WDec	77.7	60.8	68.2	–	–	–	88.1	76.1	81.7	–	–	–	–	–	–
RSAN	–	–	–	–	–	–	85.7	83.6	84.6	–	–	–	80.5	83.8	82.1
RIN	–	–	–	87.2	87.3	87.3	83.9	85.5	84.7	87.6	87.0	87.3	77.3	76.8	77.0
CasRel$_{LSTM}$	–	–	–	84.2	83.0	83.6	–	–	–	86.9	80.6	83.7	–	–	–
PMEI$_{LSTM}$	–	–	–	88.7	86.8	87.8	84.5	84.0	84.2	88.7	87.6	88.1	78.8	77.7	78.2
TPLinker$_{LSTM}$	–	–	–	83.8	83.4	83.6	86.0	82.0	84.0	90.8	90.3	90.5	91.9	81.6	86.4
CasRel$_{BERT}$	77.0*	68.0*	72.2*	89.7	89.5	89.6	89.8*	88.2*	89.0*	93.4	90.1	91.8	88.3*	84.6*	86.4*
PMEI$_{BERT}$	–	–	–	90.5	89.8	90.1	88.4	88.9	88.7	91.0	92.9	92.0	80.8	82.8	81.8
TPLinker$_{BERT}$	78.0*	68.1*	72.7*	91.3	92.5	91.9	91.4	92.6	92.0	91.8	92.0	91.9	88.9	84.5	86.7
SPN$_{BERT}$	76.0*	71.0*	73.4*	**93.3**	91.7	92.5	92.5	92.2	92.3	93.1	93.6	93.4	85.7*	82.9*	84.3*
GRTE$_{LSTM}$	74.3	67.9	71.0	87.5	86.1	86.8	86.2	87.1	86.6	90.1	91.6	90.8	88.0	86.3	87.1
GRTE$_{BERT}$	**80.1**	**71.0**	**75.3**	92.9	**93.1**	**93.0**	**93.4**	**93.5**	**93.4**	**93.7**	**94.2**	**93.9**	**92.3**	**87.9**	**90.0**
GRTE$_{w/o\ GFM}$	77.9	68.9	73.1	90.6	92.5	91.5	91.8	92.6	92.2	92.4	91.1	91.7	88.4	86.7	87.5
GRTE$_{GRU\ GFM}$	78.2	**71.7**	74.8	92.5	92.9	92.7	93.4	92.2	92.8	93.4	92.6	93.0	90.1	**88.0**	89.0
GRTE$_{w/o\ m-h}$	77.8	70.9	74.2	91.9	92.9	92.4	93.2	92.9	93.0	92.9	92.1	92.5	90.5	87.6	89.0
GRTE$_{w/o\ shared}$	79.5	71.5	**75.3**	92.7	93.0	92.8	**93.6**	92.7	93.1	93.4	94.0	93.7	91.5	87.4	89.4

Table 2: Main results. A model with a subscript *LSTM* refers to replace its *BERT* based encoder with the *BiLSTM* based encoder. * means the results are produced by us with the available source code.

Experiments

Model	NYT24*								WebNLG*							
	Normal	SEO	EPO	T = 1	T = 2	T = 3	T = 4	T $\geq$ 5	Normal	SEO	EPO	T = 1	T = 2	T = 3	T = 4	T $\geq$ 5
CasRel$_{BERT}$	87.3	91.4	92	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$_{BERT}$	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
SPN$_{BERT}$	90.8	94.0	94.1	**90.9**	93.4	94.2	95.5	90.6	89.5*	94.1*	90.8*	89.5	91.3	96.4	94.7	93.8
GRTE$_{BERT}$	**91.1**	**94.4**	**95**	90.8	**93.7**	**94.4**	**96.2**	**93.4**	**90.6**	**94.5**	**96**	**90.6**	**92.5**	**96.5**	**95.5**	**94.4**

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. * means the results are produced by us with the provided source codes.

Experiments

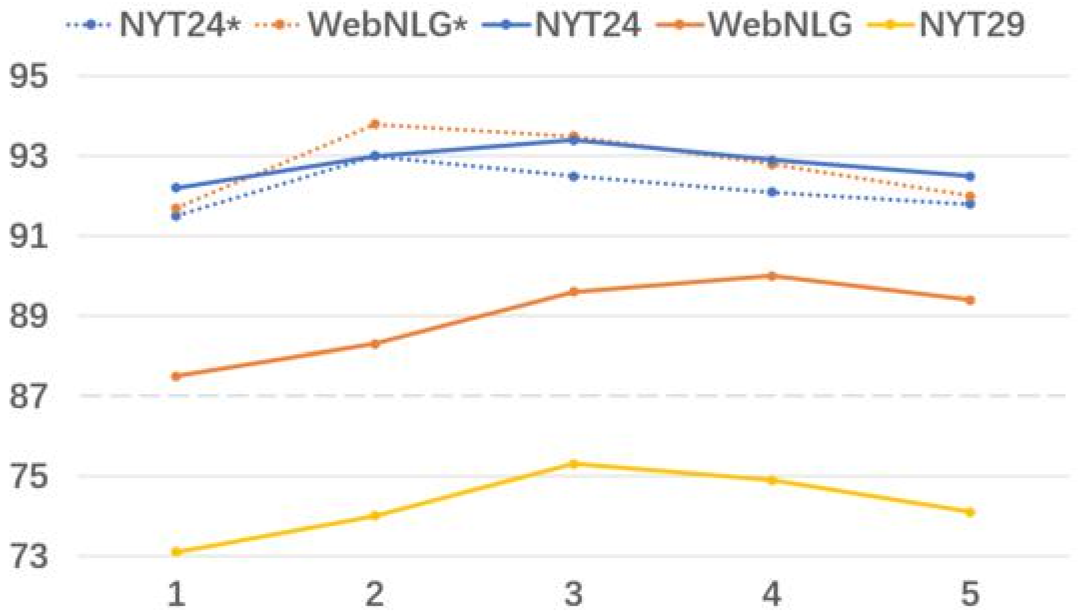


Figure 3: F1 results under different N.

Experiments

Model	NYT24*			WebNLG*		
	Params_{all}	$\text{Prop}_{encoder}$	Inference Time	Params_{all}	$\text{Prop}_{encoder}$	Inference Time
CasRel_{BERT}	107,719,680	99.96%	53.9	107,984,216	99.76%	77.5
TPLinker_{BERT}	109,602,962	98.82%	18.1 / 83.5†	110,281,220	98.21%	26.9 / 120.4†
SPN_{BERT}	141,428,765	76.58%	26.4 / 107.9†	150,989,744	71.73%	22.6 / 105.7†
GRTE_{BERT}	119,387,328	90.72%	21.3 / 109.6†	122,098,008	88.70%	28.7 / 124.1†

Table 4: Computational efficiency. Params_{all} is the number of parameters for the entire model. $\text{Prop}_{encoder}$ refers to the proportion of encoder parameters in the total model parameters. Inference Time represents the average time (millisecond) the model takes to process a sample. † marks the inference time when the batch size is set to 1.

Thanks