A Conditional Cascade Model for Relational Triple Extraction

Feiliang Ren*
renfeiliang@cse.neu.edu.cn
Northeastern University
Shenyang, China

Longhui Zhang*
Northeastern University
Shenyang, China

Shujuan Yin
Northeastern University
Shenyang, China

Xiaofeng Zhao
Northeastern University
Shenyang, China

Shilei Liu
Northeastern University
Shenyang, China

Bochao Li
Northeastern University
Shenyang, China

ABSTRACT

Tagging based methods are one of the mainstream methods in relational triple extraction. However, most of them suffer from the class imbalance issue greatly. Here we propose a novel tagging based model that addresses this issue from following two aspects. First, at the model level, we propose a three-step extraction framework that can reduce the total number of samples greatly, which implicitly decreases the severity of the mentioned issue. Second, at the intra-model level, we propose a *confidence threshold* based cross entropy loss that can directly neglect some samples in the major classes. We evaluate the proposed model on NYT and WebNLG. Extensive experiments show that it can address the mentioned issue effectively and achieves state-of-the-art results on both datasets. The source code of our model is available at: https://github.com/neukg/ConCasRTE.

CCS CONCEPTS

• **Computing methodologies → Information extraction**.

KEYWORDS

relational triple extraction, class imbalance issue

ACM Reference Format:

Feiliang Ren, Longhui Zhang, Shujuan Yin, Xiaofeng Zhao, Shilei Liu, and Bochao Li. 2021. A Conditional Cascade Model for Relational Triple Extraction. In *Proceedings of the 30th ACM International Conference on Information and Knowledge Management (CIKM '21), November 1–5, 2021, Virtual Event, QLD, Australia.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3459637.3482045

1 INTRODUCTION

Taking unstructured text (often sentences) as input, relational triple extraction (RTE for short) aims to extract triples that are in the form of (*subject, relation, object*), where both *subject* and *object* are entities and they are connected semantically by *relation.* RTE is important for some tasks like automatic knowledge graph construction.

*Both authors contribute equally to this research and are listed randomly.

CIKM '21, November 1–5, 2021, Virtual Event, QLD, Australia

ACM ISBN 978-1-4503-8446-9/21/11…$15.00
https://doi.org/10.1145/3459637.3482045

Nowadays, the methods that jointly extract entities and relations are dominant in RTE. Lots of novel joint extraction methods have been proposed [1, 5, 6, 14, 21–23, 25, 29], and they achieve much better results than the pipeline based methods. According to the extraction routes taken, most of existing joint extraction methods can be roughly classified into following three kinds. (i) *Tagging based methods* [21, 22, 29] that often use binary (positive and negative) tag sequences to determine: (1) the start and end tokens of entities, and (2) all the relations for each entity pair. (ii) *Table-filling based methods* [8, 13, 20, 28] that maintain a table for each relation and the items in a table usually denotes the start and end positions of two entities (or even the types of these entities) that possess this relation. (iii) *Seq2Seq based methods* [15, 25–27] that view a triple as a token sequence and *generate* a triple in some orders, such as first *generate* a relation, then *generate* entities, etc.

Recently, tagging based methods are attracting more and more research interests due to their superiority in both the performance and the ability of extracting triples from complex sentences that contain overlapping triples [27] or multiple triples. However, in these methods, the negative class usually contains far more samples than the positive class since there are always much more non-entity tokens in a sentence and most entity pairs possess only a very small number of relations. Therefore, these methods suffer from the *class imbalance issue* greatly: the major classes (here is the negative class) have far more samples than the minor classes (here is the positive class). This issue is very harmful to performance because it makes the training inefficient and the trained model biased towards the major classes [3, 9]. Most recent methods for addressing this issue can be divided into following two kinds [3]: (i) re-sampling based methods[31] that adjust the number of samples directly by adding repetitive data for the minor classes or removing data for the major classes; and (ii) cost-sensitive re-weighting based methods [3, 11, 12] that influence the loss function by assigning relatively higher costs to samples from minor classes. However, as [3] point out that the first ones are error-prone, and the second ones often make some assumptions on the sample difficulty and data distribution, but these assumptions do not always hold.

Obviously, the key of addressing the *class imbalance issue* is to narrow the number gap between samples in the classes of major and minor. Following this line, we propose *ConCasRTE*, a *Con*ditional *Cas*cade *RTE* model that can address this issue existed in the tagging based RTE methods from following two aspects. First, we propose a three-step extraction framework. Compared with existing two-step extraction framework [21, 22] that first extracts subjects then extracts objects and relations simultaneously based on the subjects

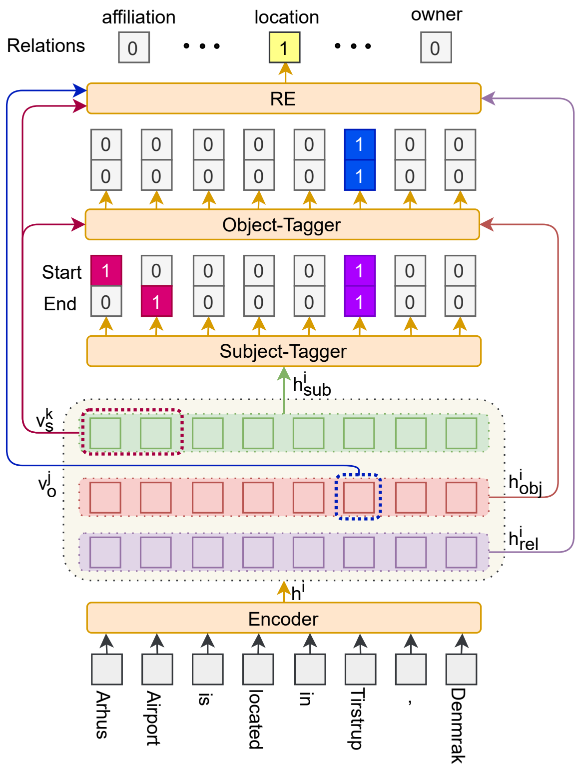

Figure 1: Model Architecture

extracted, this new framework generates far less samples. Thus it narrows the mentioned number gap implicitly due to the fact that the less samples there are, the less possibility there would be a large mentioned number gap. Second, we propose a *confidence threshold* based cross entropy loss function that can directly neglect lots of samples in the major classes, which narrows the mentioned number gap explicitly. We evaluate *ConCasRTE* on two benchmark datasets, namely NYT and WebNLG. Extensive experiments show it is effective and achieves the state-of-the-art results on both datasets.

2 METHODOLOGY

The architecture of *ConCasRTE* is shown in Figure 1. There are four main modules in it: an *Encoder* module, a *Subject-Tagger* module, an *Object-Tagger* module, and a *Relation Extraction* module (*RE* for short). These modules work in a cascade manner. And the latter three modules form a three-step extraction framework: first extracts subjects, then extracts objects, and finally extracts relations.

Encoder Firstly, a pre-trained *BERT-Base (Cased)* model [4] is used to generate an initial representation (denoted as $\mathbf{h}^i \in \mathcal{R}^{d_h}$) for each token in an input sentence. Then the context features for subjects, objects, and relations are generated with Eq.(1), where $\mathbf{W}_{(.)} \in \mathcal{R}^{d_h \times d_h}$ are trainable weights, and $\mathbf{b}_{(.)} \in \mathcal{R}^{d_h}$ are biases.

$$\begin{aligned} \mathbf{h}^i_{sub} &= \mathbf{W}_{sub}\mathbf{h}^i + \mathbf{b}_{sub} \\ \mathbf{h}^i_{obj} &= \mathbf{W}_{obj}\mathbf{h}^i + \mathbf{b}_{obj} \\ \mathbf{h}^i_{rel} &= \mathbf{W}_{rel}\mathbf{h}^i + \mathbf{b}_{rel} \end{aligned} \tag{1}$$

Subject/Object Taggers Taking each token in a sentence as input, *Subject-Tagger* uses two binary tag sequences to determine whether it is the start and end tokens of a subject, as shown in Eq.(2).

$$\begin{aligned} p^{s,i}_{start} &= \sigma(\mathbf{W}^s_{start}\mathbf{h}^i_{sub} + \mathbf{b}^s_{start}) \\ p^{s,i}_{end} &= \sigma(\mathbf{W}^s_{end}\mathbf{h}^i_{sub} + \mathbf{b}^s_{end}) \end{aligned} \tag{2}$$

where $p^{s,i}_{start}$ and $p^{s,i}_{end}$ denote the probabilities of the i-th input token being the start and end tokens of a subject respectively.

Subsequently, taking each extracted subject as an input prior condition, *Object-Tagger* extracts all objects of this subject. It also uses two binary tag sequences to determine whether a token in the input sentence is the start and end tokens of an object that can form a (*subject, object*) pair with the input subject, as shown in Eq.(3).

$$\begin{aligned} p^{o,i,k}_{start} &= \sigma(\mathbf{W}^o_{start}(\mathbf{h}^i_{obj} \circ \mathbf{v}^k_s) + \mathbf{b}^o_{start}) \\ p^{o,i,k}_{end} &= \sigma(\mathbf{W}^o_{end}(\mathbf{h}^i_{obj} \circ \mathbf{v}^k_s) + \mathbf{b}^o_{end}) \end{aligned} \tag{3}$$

where $\mathbf{v}^k_s$ is the vector representation of the k-th input subject and is obtained by simply averaging all its tokens' vector representations; $p^{o,i,k}_{start}$ and $p^{o,i,k}_{end}$ denote the probabilities of the i-th input token being the start and end tokens of an object that can form an entity pair with the k-th input subject; $\circ$ denotes the hadamard product operation.

RE Taking each (*subject, object*) pair as input, *RE* extracts all relations for this input entity pair, as shown in Eq. (4).

$$\begin{aligned} \mathbf{p}^{k,j}_r &= \frac{1}{|LOC|}\sum_{i \in LOC} \sigma\left(\mathbf{W}_r\left(\mathbf{h}^i_{rel} \circ \mathbf{v}^k_s \circ \mathbf{v}^j_o\right) + \mathbf{b}^{k,j}_r\right) \\ LOC &= [loc^{k,start}_s, loc^{k,end}_s] \cup [loc^{j,start}_o, loc^{j,end}_o] \end{aligned} \tag{4}$$

where $\mathbf{v}^k_s$ and $\mathbf{v}^j_o$ are vector representations of the k-th subject and j-th object, and $\mathbf{v}^j_o$ is obtained by the same way as $\mathbf{v}^k_s$; $\mathbf{p}^{k,j}_r \in \mathcal{R}^{|R|}$ is a probability sequence, $|R|$ is the size of relation set R, and each item in $\mathbf{p}^{k,j}_r$ corresponds to a specific relation and is used to determine whether this relation should be assigned to the input entity pair; $loc^{k,start}_s$, $loc^{k,end}_s$, $loc^{j,start}_o$ and $loc^{j,end}_o$ denote the start and end positions of the two input entities; LOC is the position range of the input entity pair, and $|LOC|$ is the number of tokens in this pair.

In Eq.(2)-(4), $\mathbf{W}^s_{(.)}, \mathbf{W}^o_{(.)} \in \mathcal{R}^{1 \times d_h}, \mathbf{W}_r \in \mathcal{R}^{|R| \times d_h}$ are weights, $\mathbf{b}^s_{(.)}, \mathbf{b}^o_{(.)} \in \mathcal{R}^1, \mathbf{b}^{(.)}_r \in \mathcal{R}^{|R|}$ are biases, and σ is a sigmoid function.

***Confidence Threshold* based Loss** Traditional loss functions like cross entropy usually assign lower costs to samples whose predictions are correct and the model is *confident* for these predictions (here we say a model is *confident* for a prediction if it assigns a very high or a very low probability for this prediction). The major classes usually account for the majority of these low cost samples due to the overwhelming number of samples in them. So it would bring following two benefits if we directly neglect these low cost samples. First, most of the neglected samples would be in the major classes. Second, neglecting these samples wouldn't have much impact on the model training since the predictions of these samples are correct and *confident*. Accordingly, the *class imbalance issue* would be alleviated greatly by such a neglect operation. Inspired by this, we propose a *confidence threshold* based cross entropy loss which makes a model only be trained by the samples whose predictions are not *confident* or incorrect, as shown in Eq. (5)- (7).

$$ce'(p,t) = \xi * ce(p,t) \tag{5}$$

$$ce(p,t) = -[tlogp + (1-t)log(1-p)] \tag{6}$$

$$\xi = \begin{cases} 0, (t-T)(p-T) > 0 \& |p-0.5| > C \\ 1, otherwise \end{cases} \tag{7}$$

where ce' is the proposed loss; ce is a basic binary cross entropy loss; $p \in (0,1)$ is a prediction probability and $t \in \{0,1\}$ is its true tag; ξ is a switch coefficient to determine whether the model be trained by an input sample; T is a hyperparameter used to determine whether a prediction is assigned 1 or 0; $C \in [0, 0.5]$ is a hyperparameter and we call it as *confidence threshold*; $|p-0.5| > C$ means the model is *confident* for the prediction: the larger the *confidence threshold* is set, the higher confident degree of the model for its predictions is required; and $(t-T)(p-T) > 0$ means the prediction is correct.

Finally, the proposed loss is used for training the modules of *Subject-Tagger*, *Object-Tagger*, and *RE*. The overall loss of *ConCasRTE* is defined as the sum of these separated losses. During training, we take the popular *teacher forcing* strategy where the ground truth samples are used as input. To alleviate the *exposure bias* issue [20] caused by this strategy, we add some randomly generated noise samples into the ground truth samples and use them together.

3 EXPERIMENTS

3.1 Experiment Settings

Datasets Here following two benchmark datasets are used: NYT [17] and WebNLG [7]. Both of them have two different versions according to following two annotation standards: 1) annotating the last token of each entity, and 2) annotating the whole entity span. Following *TPLinker* [20], 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. Some statistics of these datasets are shown in Table 1: *EPO*, *SEO*, and *Normal* refer to *entity pair overlapping, single entity overlapping*, and *no overlapped triples* respectively [27]. Note a sentence can belong to both *EPO* and *SEO*.

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: (i) *Partial Match*: an extracted triplet is regarded as correct if the predicted relation and the head of both subject entity and object entity are correct; (ii) *Exact Match*: a triple is regarded as correct only when its entities and relation are completely matched with a correct triple. Here we follow [19–21]: use *Partial Match* on NYT* and WebNLG*, and use *Exact Match* on NYT and WebNLG.

Implementation Details AdamW [10] is used to train *ConCasRTE*. All the hyperparameters are determined based on the results on the development set. Finally, they are set as follows. On NYT and NYT*, the batch size is set to 18 and epoch is set to 100. On WebNLG and WebNLG*, the batch size is set to 6 and epoch is set to 50. On all datasets, the learning rate is set to $1e^{-5}$, the *confidence threshold* (C in Eq. (7)) is set to 0.1, and all other thresholds are set to 0.5.

Baselines Following strong state-of-the-art models are taken as baselines: *ETL-Span* [22], *WDec* [14], *RSAN* [23], *RIN* [18], *PMEI* [19], *CasRel* [21], and *TPLinker* [20]. We also implement a *LSTM*-encoder version of *ConCasRTE* where 300-dimensional GloVe embeddings [16] and 2-layer stacked BiLSTM are used.

Table 1: Statistics of datasets.

Category	NYT		WebNLG	
	Train	Test	Train	Test
Normal	37013	3266	1596	246
EPO	9782	978	227	26
SEO	14735	1297	3406	457

3.2 Experimental Results

Main Results The main experimental results are shown in Table 2. We can see that *ConCasRTE* is very effective. On all datasets and under both match standards, it consistently outperforms all the compared state-of-the-art baselines in term of *F1*. As for other metrics, *ConCasRTE* achieves the best results on most of cases, and even the exceptions are very close to the best results.

Evaluations on Complex Sentences Here we evaluate *ConCasRTE*'s ability for extracting triples from complex sentences that contain overlapping triples or multiple triples. This ability is widely discussed by existing work, and can be viewed as a metric to evaluate the robustness of a model. To be fair, we conduct experiments on the same subsets as some previous best models [20, 21].

The results are in Table 3, which demonstrate the great superiority of *ConCasRTE* for handling both kinds of complex sentences. On both datasets, it achieves much better results than the compared baselines. In fact, *ConCasRTE* inherits the main strengths of existing tagging based methods for extracting triples from complex sentences, but focuses more on well addresses the *class imbalance issue* existed in these methods, thus it achieves much better results.

Detailed Analyses Table 4 shows some detailed experimental results about the proposed extraction framework and loss function. All these results are obtained when the *BERT*-based encoder used.

First, we evaluate the effectiveness of the proposed extraction framework. To this end, we implement ConCasRTE$_{Ce}$, a variant that uses the basic binary cross entropy loss. Then we compare it with *CasRel* (the current best tagging based RTE model) since the main difference between them is the extraction framework. We can see ConCasRTE$_{Ce}$ achieves much better results on all datasets. In fact, the proposed framework can reduce the total number of samples greatly, which is much helpful for alleviating the *class imbalance issue.* Taking a l-token sentence as example, the number of samples in *ConCasRTE* is $2l + 2sl + n|R|$ (s is the number of subjects extracted, and n is the number of all (*subject, object*) pairs). In this number, $2l$, $2sl$, and $n|R|$ are generated by the modules of *Subject-Tagger*, *Object-Tagger*, and *RE* respectively. In *CasRel*, the number of samples is $2l + 2sl|R|$, where $2l$ and $2sl|R|$ are generated by its modules of subject extraction and object-relation extraction respectively. Usually $n \ll l$, thus $2l + 2sl + n|R| \ll 2l + 2sl + l|R| < 2l + 2sl|R|$. And there are $2s + 2n + t$ and $2s + 2t$ samples in the positive classes of *ConCasRTE* and *CasRel* respectively (t is the number of triples), and the difference between these two numbers can be negligible since both are very small. So the number gap between samples in classes of positive and negative in *ConCasRTE* is much smaller than that in *CasRel*, which makes the *class imbalance issue* alleviated greatly.

Second, we evaluate the proposed loss function from the aspects of ability for addressing the *class imbalance issue* and adaptability.

Table 2: Main experiments. ⋆ means the results are produced by us by running the available source code.

Model	Partial Match						Exact Match					
	NYT*			WebNLG*			NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
ETL-Span	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	–	–	–	–	–	–	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	87.6	87.0	87.3	83.9	85.5	84.7	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	88.7	87.6	88.1	84.5	84.0	84.2	78.8	77.7	78.2
TPLinker$_{LSTM}$	83.8	83.4	83.6	90.8	90.3	90.5	86.0	82.0	84.0	**91.9**	81.6	86.4
CasRel$_{BERT}$	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$_{BERT}$	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$_{BERT}$	91.3	**92.5**	91.9	91.8	92.0	91.9	91.4	**92.6**	92.0	88.9	84.5	86.7
ConCasRTE$_{LSTM}$	88.1	86.6	87.3	91.2	90.8	91.0	86.6	82.3	84.4	88.3	83.9	86.0
ConCasRTE$_{BERT}$	**92.9**	92.3	**92.6**	**93.8**	**92.5**	**93.1**	**92.9**	92.1	**92.5**	90.6	**88.1**	**89.3**

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.

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$_{BERT}$	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$_{BERT}$	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
ConCasRTE$_{BERT}$	**90.6**	**94.0**	**94.1**	**90.5**	**93.8**	**93.4**	95.2	**91.7**	**91.1**	**93.3**	93.8	**90.7**	**91.9**	**95.5**	**93.4**	**91.9**

Table 4: Detailed Results (F1). ↑ means increased scores.

Models	NYT*	WebNLG*	NYT	WebNLG
ConCasRTE$_{Ce}$	91.8	91.9	91.6	87.9
ConCasRTE$_{DifW}$	92.1	92.4	91.8	88.4
ConCasRTE$_{ReS}$	92.5	92.7	91.9	88.7
ConCasRTE$_{FLos}$	92.5	92.9	92.2	89.0
ETL-Span$_{CLos}$	78.9(↑0.8)	88.8(↑1.2)	78.8(↑0.8)	84.1(↑1.0)
CasRel$_{CLos}$	90.0(↑0.4)	92.3(↑0.5)	89.6(↑0.6)	87.9(↑1.5)
TPLinker$_{CLos}$	92.2(↑0.3)	92.5(↑0.6)	92.3(↑0.3)	88.1(↑1.4)

(i) *Ability Evaluation.* To evaluate the proposed loss function's ability for addressing the *class imbalance issue*, we implement following variants that use different methods for addressing the mentioned issue. (1) ConCasRTE$_{DifW}$: a variant that assigns different weights for the losses of positive and negative classes (here 0.75 for the positive and 0.25 for the negative). (2) ConCasRTE$_{ReS}$: a re-sampling based variant that randomly selects some samples from the negative class so that makes the proportion between samples in the classes of positive and negative be a predefined threshold (here is 1:5). (3) ConCasRTE$_{FLoss}$, a variant that uses *Focal Loss* [11] (its hyperparameter γ is set to 2). We can see that the proposed loss brings the greatest performance improvement over ConCasRTE$_{Ce}$ than all the compared methods, which demonstrates the proposed loss is more effective. Different from existing state-of-the-art methods like *Focal Loss*, the proposed loss function does not try to increase the importance of samples in the minor classes. Instead, it directly removes some samples in the major classes so as to narrow the number gap between samples in the major and minor classes. These comparison results show this strategy is more effective.

(ii) *Adaptability Evaluation.* The proposed loss is applicable to a wide range of models since we don't make any assumptions about the data distribution. For example, it can be used not only in the tagging based methods, but also in other kinds of methods. To evaluate this, we transplant it to following diverse models including *ETL-Span*, *CasRel* , and *TPLinker*. These new models are marked by a subscript "*CLos*". Results show that all these new models achieve significant improvement over their original ones on all datasets.

4 CONCLUSIONS

In this paper, we propose two novelties to address the *class imbalance issue* existed in the tagging based RTE methods, which are a three-step extraction framework and a *confidence threshold* based cross entropy loss function. To the best of our knowledge, this is the first work to explore this issue in RTE. We evaluate the proposed model on two benchmark datasets. Experiments show that both novelties can alleviate the *class imbalance issue* effectively, and they help our model achieve state-of-the-art results on both datasets.

ACKNOWLEDGMENTS

This work is supported by the National Key R&D Program of China (No.2018YFC0830701), the National Natural Science Foundation of China (No.61572120), the Fundamental Research Funds for the Central Universities (No.N181602013 and No.N171602003).

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] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. 2019. Class-Balanced Loss Based on Effective Number of Samples. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 9268–9277.

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

[6] 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.

[7] 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.

[8] 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.

[9] Justin M. Johnson and Taghi M. Khoshgoftaar. 2019. Survey on deep learning with class imbalance. *Journal of Big Data* 6, 1 (2019), 1–54.

[10] 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.).

[11] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. 2020. Focal Loss for Dense Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 42, 2 (2020), 318–327.

[12] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. 2020. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314* (2020).

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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)*.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] 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.

[30] 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.

[31] Yang Zou, Zhiding Yu, B. V. K. Vijaya Kumar, and Jinsong Wang. 2018. Unsupervised Domain Adaptation for Semantic Segmentation via Class-Balanced Self-training. In *Proceedings of the European Conference on Computer Vision (ECCV)*. 297–313.