Hybrid Deep Relation Matrix Bidirectional Approach for Relational Triple Extraction

Yushuai Hu[1,2,3], Jing Zhao[1,2,3,*] Ming Li[4,*] and Lin Yuan[1,2,3]

Abstract— **Relation extraction is a crucial task within information extraction, and numerous models have demonstrated impressive results. However, most of the tagging-based relation triple extraction methods employ unidirectional approaches to extract subjects, objects, and relations, which may overlook crucial information. In this paper, we introduce a novel deep matrix-based bidirectional relation extraction model. Firstly, we extract forward and backward entity pairs. During the bidirectional extraction process,there may be some redundant relationships,so we use a shared encoder to connect and enhance the extraction process. Secondly, we design a low-complexity relation extraction matrix to allocate all possible relations. We assess our model using diverse benchmark datasets, and comprehensive experiments show that our approach effectively addresses subsequent triple extraction issues stemming from entity extraction failures.**

I. INTRODUCTION

Since the inception of relation extraction, deriving triples from unstructured natural language text has persisted as a vital task.Relation triples store factual information in the (subject, relation, object) format. Entities encompass subjects and objects, while relations semantically connect them. For instance, the triple (Elizabeth II, is the queen of, the United Kingdom) conveys the fact that "Elizabeth II is the queen of the United Kingdom." With the increasing importance of knowledge graph construction, link prediction, and many other downstream applications, the significance of relation extraction has grown, leading to the emergence of numerous relation extraction methods.

Initially, research efforts [2], [20], [25] commonly utilized a sequential model, typically comprising two stages: entity identification [11] and relationship prediction [22]. These methods were simple and straightforward, but they had two fatal flaws. Firstly, these models neglected the link between entity identification and relation forecasting. Secondly, they encountered issues stemming from error propagation inherent in their design.As a result, many subsequent studies started addressing these issues by proposing joint extraction models, which simultaneously extracted relations and entities while ensuring an end-to-end architecture [1], [5], [6], [9], [16]–[19], [21], [24].

Among these joint extraction models, one approach that uses tagging for relation extraction has shown promising performance and extraction capability. Generally, it extracts triples from two types of sentences: those containing overlapping triples and those without overlapping triples [22]. Tagging-based approaches in the current methods commonly split the relation triple extraction task into two tagging-oriented subtasks. The initial subtask involves detecting all subjects within the sentence, while the subsequent subtask encompasses identifying objects and relations.

Despite their significant success, these methods still face some serious issues. If the subject extraction fails, the extraction of objects and relations associated with that subject will also fail. Clearly, this problem has a significant impact on relation extraction.

To tackle this concern, we introduce DMBRE (Deep Matrix-based Bidirectional Relation Extraction), a model that adopts the tagging-based extraction strategy but treats subjects and objects as base entities. This is because an entity might be correctly extracted in one context even if it wasn't in another, providing a multifaceted perspective. As a result,the triples related to that entity can still be correctly extracted in the other direction. Therefore, our approach effectively resolves the problem of failing to correctly extract subjects.

Based on the above understanding, our model is designed as follows. First, it extracts entity pairs from both the subject-to-object and object-to-subject directions. The work in these two directions is performed in parallel and connected through a shared encoder, allowing the features from both directions to influence each other. The features extracted from both directions can complement each other and their results can be mutually verified, which is beneficial for the extraction of the entire triple. In this model, a large number of entity pairs are extracted, including many unnecessary noise entity pairs. Therefore, a powerful relation classification approach is required. We propose a Deep Matrix-based Relation Extraction method to help eliminate unnecessary parts of the entity pairs. By utilizing the Deep Matrix-based Relation Extraction for a given entity pair, we can explore the deep connections between the subject and object.

II. METHOD

In this chapter, we first provide the task definition and symbol representation. The input is a sentence $S =$

1 Key Laboratory of Computing Power Network and Information Security, Ministry of Education, Shandong Computer Science Center, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China. {zjstudent@126.com}

2 Shandong Engineering Research Center of Big Data Applied Technology, Faculty of Computer Science and Technology, Qilu University of Technology (Shandong Academy of Sciences), Jinan, China.

3 Shandong Provincial Key Laboratory of Computer Networks, Shandong Fundamental Research Center for Computer Science, Jinan, China.

4 School of Intelligence and Information Engineering, Shandong University of Traditional Chinese Medicine, Jinan, China.

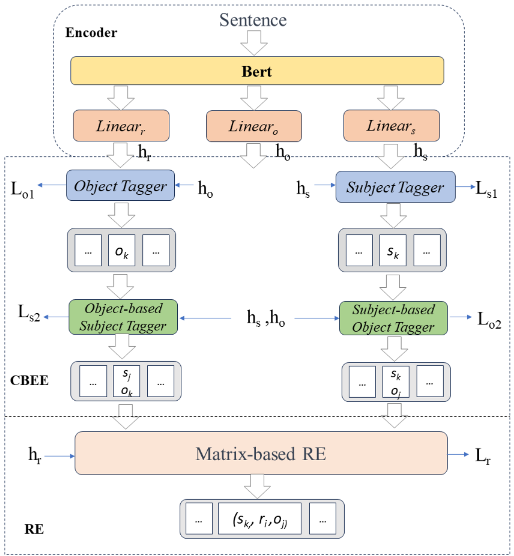

Fig. 1. The Architecture of DMBRE.DMBRE Architecture follows a structured design where modules of the same color share analogous inner structures. Solid lines depict the training phase, while the combination of dashed and solid lines represents the inference phase.

$\{W_1, W_2, ..., W_n\}$ with n tokens. The goal of the relation extraction task is to identify as many possible triples $\varepsilon = \{(h_i, r_i, t_i)\}_{i=1}^{N}$ in the sentence, where N is the number of triples, h_i represents the head entity, $t_i$represents the tail entity, and r_i represents the relation between them.

The structure of DMBRE is illustrated in the figure, and it mainly consists of the following components: Firstly, the encoder part, which utilizes a BERT-based encoder [15]. Secondly, the decoder part includes a Cross-Bidirectional Entity Extraction component (CBEE) and a Deep Relation Matrix Extraction Component (DRME). During training, multiple modules are jointly learned to accomplish the task.

A. Encoder

Firstly, we utilize a pre-trained BERT model [4] to encode the input sentence and extract feature information for each token in the sentence, providing initial representations. BERT is a language representation model built upon multi-layer bidirectional Transformers, aiming to provide deep contextual word representations in sentences. It has been widely adopted as an encoder in various models and demonstrated remarkable effectiveness for downstream tasks. In contrast to many existing models [16], [17] that use a unified feature representation for subject, object, and relation, we take into consideration that different types of entity triples may have distinct characteristics. Therefore, we adopt different representations for subject, object, and relation, denoted as h_s^i, h_o^i, and h_r^i, respectively, as shown in equations (1).

$$\begin{aligned} h_s^i &= W_s h^i + W_{ps} \\ h_o^i &= W_o h^i + W_{po} \\ h_r^i &= W_r h^i + W_{pr} \\ h_\alpha &= \text{Trans}(h_{\alpha-1}) \quad \alpha \in [1, N] \end{aligned} \tag{1}$$

where $W(.) \in R^{d\times d}$ represents the subword embedding matrix, $W_p(.)$ is the position embedding matrix,and p denotes the position index in the input sequence. h_α denotes the hidden state vector, representing the contextual representation of the input sentence at layer α, and N is the number of Transformer blocks in the model. The subword embedding matrix $W(.)$ is used to obtain the embeddings for each subword in the input sentence, while the position embedding matrix $W_p(.)$ is used to incorporate positional information into the representations, ensuring that the model considers the relative positions of tokens within the sequence. The hidden state vector h_α captures the contextual information of the input sentence at each layer of the Transformer model, and multiple layers are stacked together to obtain deeper contextual representations for the tokens.

Furthermore, due to the close relationship between subjects and objects, the extracted entity features can effectively assist in the extraction of the other entity. Therefore, we add the CLS vector of the subject to the object features and the CLS vector of the object to the subject features , as shown in equations (2).

$$\begin{aligned} X_s^i &= h_s^i + cls_o \\ X_o^i &= h_o^i + cls_s \end{aligned} \tag{2}$$

where the "CLS vector" refers to the special token in the BERT model, which is used to represent the semantics of the entire input sentence.

B. decoder

In this section, we will explain the implementation of the DMBRE decoder, inspired by the earlier formulas, we instantiate the Cross-Bidirectional Entity Extraction component (CBEE). The basic idea is to extract entity pairs from two different directions:

1) To start with, we extract the subject, and then based on this subject, we extract the object. We refer to this direction as s-o.
2) The other direction is the opposite, where we first extract the object, and then based on this object, we extract the corresponding subject. We refer to this direction as o-s.

These two directions share a similar internal structure, so in this section, we will only introduce the s-o direction.

1) Subject Tagger: This module directly decodes the encoded vectors generated by the BERT encoder and is used to identify all subjects in the sentence. Specifically, it employs a binary tagging approach to detect the starting and ending positions of the subjects. Each token is assigned a binary tag

(0/1) to indicate the possibility of the current token being the start or end of a subject. The probabilities for the start and end positions are calculated using equations (3).

$$\begin{aligned} P_{i,s}^{start} &= \sigma(W_s^{start} X_s^i + b_i^{start}) \\ P_{i,s}^{end} &= \sigma(W_s^{end} X_s^i + b_i^{end}) \end{aligned} \tag{3}$$

where $P_{i,s}^{start}$ represent the probabilities of recognizing the i-th token in the input sequence as the start and end positions of a subject. $W(.)$ represents trainable weights, $b(.)$ is the bias term, and σ is the sigmoid activation function.

2) Subject- Specific Tagger: This section is dedicated to retrieving relevant objects according to the identified subjects. It employs an iterative approach to choose the subjects and their corresponding objects. For each chosen subject, tokens in the sentence are assigned two probabilities. These probabilities denote the potential as start and end positions of an object linked to the selected subject. The probabilities are computed using the following equations.

$$\begin{aligned} v_s^{s_k} &= avgpool(h_s^{s_k,start}, ..., h_s^{s_k,end}) \\ p_{i,strat}^o &= \sigma(W_o^{start}(h_o^i \odot v_s^{s_k} + b_o^{start})) \\ p_{i,end}^o &= \sigma(W_o^{end}(h_o^i \odot v_s^{s_k} + b_o^{end})) \end{aligned} \tag{4}$$

here, $h_s^{s_k,start}, ..., h_s^{s_k,end}$ stand for vector representations of different tokens within the k-th entity, and $v_s^{s_k}$ signifies the representation of the k-th entity. The notation $avgpool(.)$ represents the average pooling operation. The probabilities $p_{i,start}^o$ and $p_{i,end}^o$ indicate whether the i-th token serves as the starting or ending point for an object linked to the k-th entity. The symbol $\odot$ denotes the Hadamard product operation. The matrix $W(.) \in R^{1\times d^h}$ is a trainable matrix, and $b(.) \in R^1$ denotes a bias vector.

3) Losses: During the training process, we jointly train the model by optimizing a combined objective function. As mentioned before, all modules in both directions are engaged in multitask learning, meaning each module in each direction has its own specific loss function. The loss functions for the two taggers mentioned above are denoted as L_{s1} and L_{o1}, and can be represented by the following formulas.

$$\begin{aligned} L(p,t) &= -(tlogp + (1-t)log(1-p)) \\ L_{s1} &= \frac{1}{2l}\sum_{start}^{end}\sum_{i=1}^{l} L(p_{i,m_s}, t_{i,m_s}) \\ L_{o1} &= \frac{1}{2l}\sum_{start}^{end}\sum_{i=1}^{l} L(p_{i,m_o}, t_{i,m_o}) \end{aligned} \tag{5}$$

where l represents the number of tokens in the input sentence, p denotes the predicted probabilities, t represents the true labels, and $L(p,t)$ represents the binary cross-entropy loss for a single token. The loss functions are separately calculated for the subject and object's starting and ending positions, and the results are averaged to obtain the total loss. Similarly, in the other direction, there are also two loss functions for the tagger, and the operation of these loss functions is similar. Let's denote them as L_{s2} and L_{o2}.

4) DRME: Through the aforementioned sequence labeling, all entity pairs in the sentence can be obtained. However, the cross-bi-directional framework generates a large number of entity pairs, many of which may contain noise, impacting model accuracy. To enhance robustness and precision, a Deep Relation Matrix Extraction Component is incorporated for relation extraction.

Using the previously extracted subject feature representation $r_{s,k}$ and object feature representation $r_{o,j}$, their corresponding values in the deep relation matrix are calculated to determine if they form the correct triple relation. Subsequently, the likelihood is computed using the following equation.

$$\begin{aligned} r_{s,k} &= avgpool(h_r^{s_k,start}, ..., h_r^{s_k,end}) \\ r_{o,j} &= avgpool(h_r^{o_k,start}, ..., h_r^{o_k,end}) \\ p_r^i &= \sigma(\begin{bmatrix} r_{s,k} \\ 1 \end{bmatrix} W_r^i \begin{bmatrix} r_{o,j} \\ 1 \end{bmatrix} + b_r) \end{aligned} \tag{6}$$

We adopt the DRME for several key advantages. During the entity pair extraction process, it simultaneously considers all possible relations and utilizes global information to determine correct entity pairs. This approach promotes the sharing of cross-relation information among entity pairs, ultimately enhancing the accuracy of relation extraction.Furthermore, DRME effectively encodes the representation of each entity into a comprehensive vector that assimilates all relevant information about the entity. This enables better capturing of contextual information surrounding the entities, thereby improving the performance of relation extraction.Another notable strength of DRME lies in its end-to-end learning capability. This feature allows the entire model to be trained within a unified framework, leading to better optimization for the relation extraction task.

DRME Loss We use a loss function based on binary cross-entropy to train this component, as follows.

$$L_{RE} = -\frac{1}{n^2}\sum_{i=1}^{l}\sum_{j=1}^{l}(t_{i,j}logp_{is,io} + (1-t)log(1-p_{is,io})) \tag{7}$$

where L_{RE} represents the loss associated with the deep relation matrix, specifically a binary cross-entropy loss. This loss quantifies the disparity between the predicted correspondence score and the actual correspondence label. We signifies the word count within the input sentence. I and j are the indices of the subject and object's starting positions, respectively,t represents the true correspondence label, which is a binary value (0 or 1), indicating whether the i-th word and the j-th word form a correct subject-object pair. $p_{is,io}$ is the predicted correspondence values, ranging from 0 to 1, representing the confidence that the i-th word and the j-th word form a subject-object pair. By using L_{RE}, the model can learn which triplets are valid and which are not, enabling it to distinguish between effective and ineffective subject-object pairs.

C. Joint Learning Mechanism

In traditional single-task learning, each task typically has a separate model, and their underlying representation layers are independent. In the given scenario with a total of 5 modules, during multi-task learning, these modules are transformed into relatively independent extraction and encoding tasks. Therefore, we adopt the popular Shared Representation approach, where shared parameters are set in the lower-level structure of the model, allowing multiple tasks to share the intermediate layers of the model through joint training. The loss function for the shared representation is a combination of the loss functions from each individual task. In multi-task learning, joint training may be employed, which involves summing up the loss functions of all tasks to form a total loss function. This total loss function is then optimized to update the shared representation layer's parameters. The combined loss function is as shown in formula(8).

$$L = L_{s1} + L_{s2} + L_{o1} + L_{o2} + L_{RE} \tag{8}$$

At the beginning, we updated the shared representation layer's parameters by minimizing the overall loss function. However, we noticed that the shared encoder could be influenced differently by each module, leading to variations in convergence speed. This inconsistency in convergence could cause issues. To address this problem, we proposed a Synchronized Learning mechanism. The fundamental idea behind this mechanism is to allocate a relatively small learning rate to the shared modules during multi-task learning. The equation for the Synchronized Learning mechanism is shown as follows.

$$L_i = \begin{cases} l, & T_i = 1, \\ \frac{b}{f(T_i)} & T_i > 1 \end{cases} \tag{9}$$

where L_i denotes the learning rate for the i-th module. Here, l stands for the base learning rate, T_i represents the count of tasks utilizing the i-th module, and α is an adjusting factor. When T_i is equal to 1, indicating exclusive usage by a single task, the module's learning rate matches the base learning rate. However, if T_i surpasses 1, signifying shared usage by multiple tasks, the module's learning rate becomes less than the base learning rate. Moreover, as the number of shared tasks increases, the learning rate diminishes further.This arrangement ensures that modules shared by multiple tasks have reduced learning rates to balance the impact of different tasks and promote stable learning across various modules in the multi-task learning setup.

III. EXPERIMENTS

A. Experiment Settings

Dataset and Evaluation Metrics The dataset chosen includes NYT [10], and WebNLG [7]. For NYT and WebNLG, there are two versions based on different annotation standards: (1) annotations with only the final marked entities, (2) annotations with entire entities. For simplicity, we refer to the first standard as NYT* and WebNLG*, and the second standard as NYT and WebNLG. The complete dataset with annotations provides a better representation of the model's true performance. For the Relation Triple Extraction task, there are two matching criteria: partial matching and complete matching. Partial matching considers a predicted triple correct if the predicted relation and the head of the subject and object entities are correct. Complete matching requires the predicted triple to exactly match the correct triple in terms of entities and relation. We use partial matching on NYT* and WebNLG* datasets and complete matching on NYT and WebNLG datasets. The evaluation is based on standard metrics, including micro-precision, recall, and F1 score.

Implementation Details Follow previous work [16], [17],we implemented the model using PyTorch and trained it with the Adam optimizer [8]. The batch size was set to 18 for NYT, NYT* datasets, while for WebNLG and WebNLG* datasets, the batch size was set to 8. The threshold for the deep relation matrix was set to 0.5. In Formula 9, the base learning rate (l) was set to e^{-4}, and the adjusting factor (b) was set to 1. Hyperparameter values were established using the development set outcomes, while other parameters were initialized at random.

B. Experimental Result

Compared Methods We compare our proposed model against a range of state-of-the-art models, including ETL-Span [18], WDec [9], RSAN [19], RIN [13], CasRel [17], TPLinker [16], StereoRel [14], PRGC [23], R-BPtrNet [3],and SPN [12].

Main Results The experimental results, as shown in Table 1, demonstrate that our proposed model achieves promising F1 scores across all datasets. It particularly stands out in terms of recall and F1 scores, approaching the top performance levels in these areas.It is noteworthy that the precision performance is slightly compromised, primarily due to the adoption of a cross-bi-directional structure, resulting in more noise pairs being extracted. This impacts the model's precision. However, our model still exhibits competitive performance compared to other models. The high recall enables the model to better capture true positives. In information extraction tasks, a high recall indicates the model's ability to comprehensively detect all existing entities and relations, thus reducing omission. It is this high recall that significantly contributes to the improved F1 scores in our model.

In the following sections, we will evaluate DMBRE from various perspectives to gain a comprehensive understanding of its performance.

Model	Partial Match						Exact Match					
	NYT*			WebNLG*			NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1.	Prec.	Rec.	F1	Prec.	Rec.	F1
ETL-Span [18]	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 [9]	-	-	-	-	-	-	88.1	76.1	81.7	-	-	-
RSAN [19]	-	-	-	-	-	-	85.7	83.6	84.6	80.5	83.8	82.1
RIN [13]	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 [17]	89.7	89.5	89.6	93.4	90.1	91.8	89.8	88.2	89.0	88.3	84.6	86.4
TPLinker [16]	91.3	92.5	91.9	91.8	92.0	91.9	91.4	92.6	92.0	88.9	84.5	86.7
StereoRel [14]	92.0	92.3	92.2	91.6	92.6	92.1	92.0	92.3	92.2	-	-	-
PRGC [23]	93.3	91.9	92.6	**94.0**	92.1	93	**93.5**	91.9	92.7	**89.9**	87.2	
R-BPtrNet [3]	92.7	92.5	92.6	93.7	92.8	93.3	-	-	-	-	-	-
SPN [12]	**93.5**	91.8	92.7	93.1	93.6	93.4	-	-	-	-	-	-
DMBRE(ours)	92.6	**93.6**	**93.1**	93.0	**93.8**	**93.6**	92.1	**93.5**	**92.8**	89.0	**88.8**	**88.9**

TABLE I: Main experiments using NYT*, WebNLG*, NYT, and WebNLG

	WebNLG*							
Model	Normal	SEO	EPO	T = 1	T = 2	T = 3	T = 4	T $\geq$ 5
CasRel [17]	89.4	92.2	94.7	89.3	90.8	94.2	92.4	90.9
TPLinker [16]	87.9	92.5	95.3	88	90.1	94.6	93.3	91.6
PRGC [23]	90.4	93.6	95.9	89.9	91.6	95	94.8	**92.8**
R-BPtrNet [3]	89.5	93.9	**96.1**	88.5	91.4	**96.2**	**94.9**	94.2
DMBRE(ours)	**90.3**	**94.3**	94.6	**90.2**	**91.9**	95.4	94.7	92.7
	NYT*							
Model	Normal	SEO	EPO	T = 1	T = 2	T = 3	T = 4	T $\geq$ 5
CasRel [17]	87.3	91.4	92.0	88.2	90.3	91.9	94.2	83.7
TPLinker [16]	90.1	93.4	94	90.0	92.8	93.1	96.1	90.0
PRGC [23]	91.0	94.0	94.5	91.1	93.0	93.5	95.5	93.0
R-BPtrNet [3]	90.4	94.4	**95.2**	89.5	93.1	93.5	**96.7**	91.3
DMBRE(ours)	**91.2**	**94.6**	94.1	**91.2**	**93.5**	**94**	95.0	**92.6**

TABLE II: F1 scores across sentences with varying overlapping patterns and different triplet quantities. The results for CasRel were directly copied from TPLinker. "T" indicates the number of triples contained within a sentence.

Model	Partial Match						Exact Match					
	NYT*			WebNLG*			NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1.	Prec.	Rec.	F1	Prec.	Rec.	F1
DMBREs2o	91.6	91.5	91.3	92.1	90.1	90.8	91.4	90.7	91.0	87.3	87.4	87.0
DMBREo2s	91.4	91.2	91.2	91.5	90.7	91.3	91.2	90.8	90.7	88.1	87.6	87.8
DMBRE	**92.6**	**93.6**	**93.1**	**93.0**	**93.8**	**93.4**	**92.1**	**93.5**	**92.8**	**89.0**	**88.8**	**88.9**

TABLE III: F1 results of the ground entity extraction

Extracting triples In evaluating DMBRE's performance, we consider its ability to extract triplets from sentences containing multiple and overlapping triplets. The model's robustness and efficiency in information retrieval are crucial as they directly impact downstream tasks and improve the model's comprehension and modeling capabilities of complex semantic structures. Given that many contemporary models are being tested on this capacity, we also assess DMBRE's performance in this aspect. Following the settings of earlier state-of-the-art models, we classify sentences from different subsets of NYT and WebNLG based on the number of overlapping triplets and the presence of multiple triplets. Based on the data presented in the table II, DMBRE exhibits a significant advantage in extracting sentences containing multiple and overlapping triplets, outperforming the baseline models on these datasets. Its proficiency in handling overlapping triplets can be attributed to its strong contextual understanding capabilities. Overlapping triplets in a sentence indicate the presence of multiple triplets with potential cross-intersections or overlaps between their entities. Traditional relation extraction models may struggle in such scenarios as they often consider only local information and overlook the global context.

Next, we will further elaborate on the performance of the cross-bi-directional framework through experiments:

In the initial evaluation, we perform entity pair extraction separately for each direction and observe the results for two specific cases: (i) DMBREs2o, which exclusively performs entity pair extraction in the s2o direction, and (ii) DMBREo2s, which solely conducts entity pair extraction in the o2s direction. The outcomes demonstrate a general decline in extraction performance for each single direction. This finding underscores the importance of our bi-directional structure, as it effectively compensates for the limitations of single-directional extraction, leading to an overall enhancement in the model's performance.

C. CONCLUSIONS

In this paper,we propose a Deep Matrix-based Bidirectional Relation Extraction model.Unlike single-directional models, our bidirectional structure effectively addresses the issue of failed triplet extraction due to entity extraction failures. Additionally, to overcome the challenge of inconsistent convergence rates caused by different learning rates in various components, we introduce the synchronous learning mechanism. Extensive experiments on standard datasets demonstrate the effectiveness of our model. The results indicate that our model outperforms the majority of baseline models and exhibits robust generalization capabilities.

ACKNOWLEDGMENTS

This work is supported in part by The Key R&D Program of Shandong Province (2021SFGC0101), The 20 Planned Projects in Jinan(202228120), National Key Research and Development Plan under Grant No.2019YFB1404700.

REFERENCES

[1] G. Bekoulis, J. Deleu, T. Demeester, and C. Develder, "Joint entity recognition and relation extraction as a multi-head selection problem," *Expert Systems with Applications*, vol. 114, pp. 34–45, 2018.

[2] Y. S. Chan and D. Roth, "Exploiting syntactico-semantic structures for relation extraction," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, 2011, pp. 551–560.

[3] Y. Chen, Y. Zhang, C. Hu, and Y. Huang, "Jointly extracting explicit and implicit relational triples with reasoning pattern enhanced binary pointer network," in *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021, pp. 5694–5703.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[5] M. Eberts and A. Ulges, "Span-based joint entity and relation extraction with transformer pre-training," *arXiv preprint arXiv:1909.07755*, 2019.

[6] T.-J. Fu, P.-H. Li, and W.-Y. Ma, "Graphrel: Modeling text as relational graphs for joint entity and relation extraction," in *Proceedings of the 57th annual meeting of the association for computational linguistics*, 2019, pp. 1409–1418.

[7] C. Gardent, A. Shimorina, S. Narayan, and L. Perez-Beltrachini, "Creating training corpora for nlg micro-planning," in *55th annual meeting of the Association for Computational Linguistics (ACL)*, 2017.

[8] D. Kinga, J. B. Adam *et al.*, "A method for stochastic optimization," in *International conference on learning representations (ICLR)*, vol. 5. San Diego, California;, 2015, p. 6.

[9] T. Nayak and H. T. Ng, "Effective modeling of encoder-decoder architecture for joint entity and relation extraction," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 8528–8535.

[10] S. Riedel, L. Yao, and A. McCallum, "Modeling relations and their mentions without labeled text," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, September 20-24, 2010, Proceedings, Part III 21*. Springer, 2010, pp. 148–163.

[11] E. F. Sang and F. De Meulder, "Introduction to the conll-2003 shared task: Language-independent named entity recognition," *arXiv preprint cs/0306050*, 2003.

[12] D. Sui, X. Zeng, Y. Chen, K. Liu, and J. Zhao, "Joint entity and relation extraction with set prediction networks," *IEEE Transactions on Neural Networks and Learning Systems*, 2023.

[13] K. Sun, R. Zhang, S. Mensah, Y. Mao, and X. Liu, "Recurrent interaction network for jointly extracting entities and classifying relations," *arXiv preprint arXiv:2005.00162*, 2020.

[14] X. Tian, L. Jing, L. He, and F. Liu, "Stereorel: Relational triple extraction from a stereoscopic perspective," in *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021, pp. 4851–4861.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[16] Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu, and L. Sun, "Tplinker: Single-stage joint extraction of entities and relations through token pair linking," *arXiv preprint arXiv:2010.13415*, 2020.

[17] Z. Wei, J. Su, Y. Wang, Y. Tian, and Y. Chang, "A novel cascade binary tagging framework for relational triple extraction," *arXiv preprint arXiv:1909.03227*, 2019.

[18] B. Yu, Z. Zhang, X. Shu, Y. Wang, T. Liu, B. Wang, and S. Li, "Joint extraction of entities and relations based on a novel decomposition strategy," *arXiv preprint arXiv:1909.04273*, 2019.

[19] Y. Yuan, X. Zhou, S. Pan, Q. Zhu, Z. Song, and L. Guo, "A relation-specific attention network for joint entity and relation extraction," in *International joint conference on artificial intelligence*. International Joint Conference on Artificial Intelligence, 2021.

[20] D. Zelenko, C. Aone, and A. Richardella, "Kernel methods for relation extraction," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1083–1106, 2003.

[21] D. Zeng, H. Zhang, and Q. Liu, "Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 34, no. 05, 2020, pp. 9507–9514.

[22] X. Zeng, D. Zeng, S. He, K. Liu, and J. Zhao, "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)*, 2018, pp. 506–514.

[23] H. Zheng, R. Wen, X. Chen, Y. Yang, Y. Zhang, Z. Zhang, N. Zhang, B. Qin, M. Xu, and Y. Zheng, "Prgc: Potential relation and global correspondence based joint relational triple extraction," *arXiv preprint arXiv:2106.09895*, 2021.

[24] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, "Joint extraction of entities and relations based on a novel tagging scheme," *arXiv preprint arXiv:1706.05075*, 2017.

[25] G. Zhou, J. Su, J. Zhang, and M. Zhang, "Exploring various knowledge in relation extraction," in *Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05)*, 2005, pp. 427–434.