



Jianming Liang¹, Qing He^{1,2,*}, Damin Zhang¹ and Shuangshuang Fan³

- ¹ College of Big Data & Information Engineering, Guizhou University, Guiyang 550025, China; liangjm131@163.com (J.L.); dmzhang@gzu.edu.cn (D.Z.)
- ² Guizhou Provincial Key Laboratory of Public Big Data, Guizhou University, Guiyang 550025, China
- ³ Department of Information and Electronics, Science and Technology College of NCHU,
- Nanchang 332020, China; ssfan138@163.com
- Correspondence: qhe@gzu.edu.cn

Abstract: In recent years, scholars have paid increasing attention to the joint entity and relation extraction. However, the most difficult aspect of joint extraction is extracting overlapping triples. To address this problem, we propose a joint extraction model based on Soft Pruning and GlobalPointer, short for SGNet. In the first place, the BERT pretraining model is used to obtain the text word vector representation with contextual information, and then the local and non-local information of the word vector is obtained through graph operations. Specifically, to address the lack of information caused by the rule-based pruning strategies, we utilize the Gaussian Graph Generator and the attention-guiding layer to construct a fully connected graph. This process is called soft pruning for short. Then, to achieve node message passing and information integration, we employ GCNs and a thick connection layer. Next, we use the GlobalPointer decoder to convert triple extraction into quintuple extraction to tackle the problem of problematic overlapping triples extraction. The GlobalPointer decoder, unlike the typical feedforward neural network (FNN), can perform joint decoding. In the end, to evaluate the model performance, the experiment was carried out on two public datasets: the NYT and WebNLG. The experiments show that SGNet performs substantially better on overlapping extraction and achieves good results on two publicly available datasets.

Keywords: joint extraction; overlapping entity; overlapping relation extraction; soft pruning; GlobalPointer

1. Introduction

The entities and the relationships between entities are the main elements in the knowledge graph (KG) [1], and its general form is (subject entity, relation, object entity), referred to as entity-relation triplet for short. Most current knowledge graphs are stored in the form of triples, such as DBpedia [2], Free-base [3], NELL [4], Probase [5], Wikidata [6], etc. Extracting triples from unstructured text is a fundamental task of information extraction in natural language processing (NLP) and a key step in knowledge graph construction (KGC) [7,8].

Pipeline-based relational triplet extraction approaches were used at the beginning, such as DSPT [9], and kernel methods [10]. The work is decomposed into two distinct subtasks using the pipeline approach: named entity identification (NER) [11,12] and relationship extraction (RE) [13,14]. This method first identifies the entities and then predicts the relationship of each entity pair. The pipeline approach has the advantages of simplicity, flexibility, and ease of implementation. However, this method tends to overlook the interactions between entities and relationships and is prone to error propagation [15].

To establish the interaction between these two sub-tasks, the joint extraction method is presented. Compared with the pipeline method, the joint extraction method integrates the information of entities and relations and effectively reduces the error propagation; thus,



Citation: Liang, J.; He, Q.; Zhang, D.; Fan, S. Extraction of Joint Entity and Relationships with Soft Pruning and GlobalPointer. *Appl. Sci.* 2022, *12*, 6361. https://doi.org/10.3390/ app12136361

Academic Editor: Luis Javier Garcia Villalba

Received: 11 May 2022 Accepted: 17 June 2022 Published: 22 June 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https:// creativecommons.org/licenses/by/ 4.0/). this method has received more and more attention from scholars. The joint extraction methods are divided into the feature-based model [16–19] and the neural network-based model [20,21]. The feature-based model requires a more complex preprocessing, which leads to the introduction of additional error information. Furthermore, feature-based models necessitate manual feature extraction, which increases the burden significantly. In contrast, feature learning based on neural network models is performed by machines without human intervention.

The joint learning approach can integrate information from both because it can extract entities and relations simultaneously [22]. An ideal joint extraction system should be able to adaptively handle multiple situations, especially triple extraction where entities overlap, i.e., multiple relations share a common entity, which most methods cannot handle. An example of triple overlap is shown in Figure 1. The triple overlapping problem is separated into three categories: normal, single entity overlap (SEO), and entity pair overlap (EPO). Some researchers have studied the problem. HRL [23] introduces reinforcement learning methods into seq2seq models to enhance interactions between entities, which resulted in a significant improvement. GraphRel [24] utilizes the graph convolutional networks (GCNs) [25] to build a relational graph, which takes into account the relation between all words.

	Texts	Triples		
Normal	The capital of [China] is [Beijing]	(China, capital, Beijing)		
SEO	[Stephen Chow] starred in "[The King of Comedy]" and "[A Chinese Odyssey]".	(Stephen Chow, actor, The King of Comedy) (Stephen Chow, actor, A Chinese Odyssey)		
EPO	"[Kung Fu]" directed and starring [Stephen Chow] will be released in the near future.	(Stephen Chow, actor, Kung Fu) (Stephen Chow, director, Kung Fu)		

Figure 1. Examples of overlapping triples are divided into normal, single entity overlap (SEO), and entity pair overlap (EPO).

Despite the success of federated learning approaches, there are still drawbacks. First, in previous work using graph neural networks to obtain representations of nodes, the graph input to the graph convolutional network is pruned. However, this is prone to missing information and does not guarantee that the neural network can learn the potential representations of all nodes. Secondly for entity- and relation-independent decoding, if wrong entities are identified, each relation is also assigned to these entities, resulting in a large number of invalid triples. When there are multiple relationships between entity pairs, the classifier becomes confused and cannot make accurate judgments [26].

To solve the above problems, this paper proposes a joint entity and relation extraction with Soft Pruning and GlobalPointer (SGNet), whose purpose is to improve the extraction accuracy of triples. Specifically, firstly, the graph module is used to extract the node features to obtain the token's local and global information. In terms of decomposition, to avoid pruning, the graph we construct is fully connected. The node in the graph is each token, and the edge is determined by KL divergence to calculate the distribution difference between the two nodes, and the distribution of the nodes is generated by a trainable Gaussian Graph Generator. Then, the GCNs and dense connection layer are used to obtain local and non-local information of nodes. Finally, to realize joint decoding, we convert the triple (*s*, *r*, *o*) extraction procedure into quintuple (*sh*, *st*, *r*, *oh*, *ot*) extraction, where subject head, subject tail, object head, and object tail are represented as *sh*, *st*, *oh*, *on*, *ot*), (*sh*, *oh* | *p*), and (*st*, *ot* | *p*) to score.

Our contributions are as follows:

- (1) We employ a Gaussian Graph Generator to initialize the text graph to avoid the problem of missing information caused by pruning. Each word in the sentence is a node in the graph. Edges are obtained by computing the distribution difference between two nodes by KL divergence to encourage information propagation between nodes with high distribution differences.
- (2) We decompose the quintuple extraction problem into scoring the four token pairs after transforming the triple extraction into a quintuple extraction task. Constructing (sh, st) matrices and (oh, ot) matrices using GlobalPointer, as well as (sh, oh | p) matrices and (st, ot | p) matrices under certain relations, allows for joint entity–relational extraction.
- (3) We conduct experiments to evaluate our model on the two NYT and WebNLG public datasets. The experimental results show that our model outperforms the baseline model in extracting both overlapping and non-overlapping triples, demonstrating the effectiveness of the graph module and joint decoding module.

2. Related Work

The early relationship extraction work generally adopts the pipeline method [27,28], that is, the named entity recognition task is performed on the unstructured text first, and then the relationship extraction task is performed, that is, the entity is extracted first and then the relationship between the entity pairs is determined. While the pipeline approach is simple to use, both extraction models are quite flexible in that the entity and relationship models can use different datasets and do not require both annotated entity and relationship databases. However, this will result in the following drawbacks: (1) Error accumulation: the performance of relation extraction in the next phase will be affected by the error of entity extraction. (2) Entity redundancy: since the entities are paired in pairs initially, then the relation is categorized, and the candidate entities with no association will provide duplicate information, increasing the error rate and complexity. (3) Interaction separation: the dependency and internal relationship between the two subtasks are neglected.

To overcome the above drawbacks, some scholars gradually proposed the joint extraction model. It means that only one model can be used to extract the triples in the text, which strengthens the connection between the two subtasks of entity extraction and relation extraction, and alleviates error propagation [29]. The joint extraction model for entities and relations can be categorized into two types: joint extraction models based on parameter sharing and joint extraction models based on joint decoding. The parameter sharing model is to share the parameters of the input features or internal hidden layers. Miwa et al. reported that a recurrent-neural-network-based joint extraction model uses a bidirectional long short-term memory network (Bi-LSTM) to encode entities first, and uses Tree-LSTM, which considers dependency tree-based information to model the relationship between entities [30]. However, this model only works at the sentence level and is only suitable for simple dependency parsing. Katiyar et al. reported that pointer network decoding queries all entities before their position (forward query) for the current entity, and calculates the attention score [31]. Zeng et al. reported that a joint extraction model with a fusion copy mechanism is based on the sequence-to-sequence (Seq2Seq) idea, but the model decoder only copies the tail bytes of the entity, resulting in the incomplete extraction of multi-byte entities [32]. The joint extraction model based on parameter sharing has no constraints on sub-models, but the interaction between the entity and relational models is weak due to the use of an independent decoding technique.

To enhance the interaction between the two sub-models, a joint decoding algorithm is proposed. Dai et al. reported that a unified joint extraction model, which labels entity and relation labels according to query word position p, then detects entities at p and identifies entities at other positions that have relations with the former [15]. Arzoo et al. used conditional random field (CRF) to simultaneously model entity and relationship models and obtained the output results of entity and relationship through the Viterbi decoding algorithm [33]. Zhang et al. reported a globally optimized end-to-end relation extraction

neural model and proposed new LSTM features. Furthermore, they proposed a new method to integrate syntactic information to facilitate global learning [34]. Zheng et al. proposed a special label type to transform entity recognition and relation classification into a sequence labeling problem. The sentence is encoded by the encoding layer, and then the hidden layer vector is input into the decoding layer to directly obtain triples. The extraction process is not divided into two sub-processes: entity recognition and relationship classification [35]. Wei et al. reported that in a cascaded binary-based annotation framework, i.e., firstly, a pointer is used to annotate the start and end positions of subject entities, all possible subject entities in the sentence are extracted, and then a pointer annotation is used to identify all possible relations and object entities of subject entities [26]. However, the difficulty of joint entity relation extraction lies in overlapping entity relation extraction.

To overcome this difficulty in extracting overlapping triples, Yu et al. proposed a decomposition strategy, which decomposes the extraction task into first extracting head entities, and then extracting entity relations, and the two tasks share the coding layer. The two subtasks are further transformed into a multi-sequence labeling problem using a span-distance-based labeling scheme [36]. Wang et al. introduced a novel handshake tagging strategy to make the following judgments for a word in a sentence: whether it is the beginning or the end of an entity, and whether it is the head or the tail of an entity under a particular relation. Such a judgment is made to improve the accuracy of overlapping triples recognition [37].

Different from the previous work of extracting triples, in this paper, we directly convert the triples into quintuples, then score the elements in the quintuple one by one, and finally use the joint decoding module to parse out the triples.

3. Methodology

Figure 2 shows the overall structure of our model, which consists of three parts: the BERT model, the graph module, and the joint decoding module. We will describe each section in detail below.



Figure 2. The model of SGNet. It includes the BERT model, graph model, and joint decoding model, where *sh* stands for subject head, *st* for subject tail, *oh* for object head, and *ot* for object tail.

3.1. BERT Model

Before performing the graph operation, the sentence must be transformed into word embedding. We employ the BERT [38] as a pretraining model to encode sentence semantic vectors in this paper. Compared with the traditional Word2Vec [39] word embedding method, the BERT model takes word position into account. Because a word can express distinct meanings in different places, word position information cannot be disregarded throughout the word embedding process.

The BERT is a pretraining model that includes predictive context information and location information. Given a sequence *X* with *n* tokens, we map *X* to a BERT input sequence $X_{Input} = [x_0, x_1, ..., x_{n+1}]$. Here, x_0 represents the "[CLS]" token at the start of the sentence, and x_{n+1} represents the "[SEP]" token at the end of the sentence. After BERT encoding, the corresponding token is represented as $H' = [v_0, v_1, v_2, ..., v_{n+1}]$. Here, v_0 of token "[CLS]" is considered as the task-specific token of the entire sequence, and $H = [v_1, v_2, ..., v_{n+1}]$ is the word embedding we employ in the downstream task.

3.2. Graph Model

As illustrated in Figure 3, our graph module consists of graph generation, attentionguided layer, densely connected layer, and linear combination layer.



Figure 3. The graph module is shown with an example of four nodes and an adjacency matrix. The node embeddings and adjacency matrix are generated with KL divergence as inputs. Then, employing multi-head attention to construct *N* attention-guided adjacency matrices, the resulting matrices are fed into *N* separate densely connected layers, generating new representations. Finally, a linear combination layer is applied to integrate the outputs of the *N* different densely connected layers.

3.2.1. Gaussian Graph Generator

We utilize the Gaussian Graph Generator [40] to construct the graph's edge to minimize the inaccuracy produced by the natural language tools. Specifically, we encode each node v_i of BERT model output into Gaussian distribution as follows:

$$\begin{cases} f_{\theta}(v_i) = \{\mu_i\} \\ Af'_{\theta}(v_i) = \{\sigma_i\} \end{cases}$$
(1)

where f_{θ} and f_{θ}' are two trainable neural networks and A is a nonlinear activation function. The SoftPlus function is used as the activation function because the standard deviation of a Gaussian distribution is confined (0, + ∞).

3.2.2. KL Divergence

In probability theory or information theory, KL divergence (Kullback–Leibler divergence), also known as relative entropy, is a way to describe the difference between two probability distributions p and Q. It can measure data objects whose geometric distances

(such as cosine distance, Euclidean distance) are difficult to measure [41]. When the two distributions are closer, the KL divergence value is smaller. KL divergence is defined as follows:

$$D_{KL}(p||q) = \sum_{i=1}^{n} p(x_i) \log\left(\frac{p(x_i)}{q(x_i)}\right)$$
(2)

where p(x) is the target distribution, q(x) is the matching distribution, x_i is the discrete random variable, and N is the length of the distribution.

To evaluate the differences in the distributions of the two nodes, KL divergence is utilized to determine the connection strength between nodes. The propagation of messages between token representations with considerable semantic variances is encouraged to obtain the potential links between tokens. As an example, the semantic gap between "we" and "our" is extremely tiny, so the strength of association between two words is small and the weight assigned is negligible; the semantic gap between "Jobs" and "Apple" is large, so the connection strength of the two words is vast and the weight assigned is significant. The edge weight between the *i*-th and *j*-th nodes is calculated as follows:

$$e_{ij} = D_{KL}(N_i(\mu_i, \sigma_i^2)) || N_j(\mu_j, \sigma_j^2))$$
(3)

where D_{KL} denotes KL divergence calculation. $N_i(\mu_i, \sigma_i^2)$ and $N_j(\mu_j, \sigma_j^2)$ are the distributions of the *i*-th node and *j*-th node. We obtain the graph G = (V, A) with a directed graph due to the asymmetry of the KL divergence. V denotes the set of all nodes, and A denotes the adjacency matrix.

GCNs are neural networks that operate directly on graph structures. Let us take a look at how the nodes are updated first for multi-layer GCNs. Given a graph with *n* nodes, we can represent the graph with an $n \times n$ adjacency matrix A. In traditional GCN operations, $A_{ij} = 1$ and $A_{ji} = 1$ if there is an edge going from node *i* to node *j*, otherwise $A_{ij} = 0$ and $A_{ji} = 0$. The purpose of using GCN is to aggregate adjacent nodes and learn the information of K-order nodes. The convolution calculation of node *i* at the *l*-th layer takes h^{l-1} as input and outputs $h^{(l)}$:

$$h_i^{(l)} = \rho\left(\sum_{j=1}^n A_{ij} W^{(l)} h_j^{(l-1)} + b^{(l)}\right)$$
(4)

where A denotes the adjacency matrix and W represents the weight matrix. ρ is the activation function, and $h_i^{(0)}$ is the initial input representation v_i .

We know that in most previous research, the input graph was pruned. However, rule-based pruning strategies might eliminate some crucial information from the entire tree. As a result, we use the attention guidance layer to turn the graph into *N* fully connected graphs [42]. The self-attention mechanism is used to create the adjacency matrix, which can be written as follows:

$$\mathbf{A}^{(t)} = softmax \left(\frac{Q \mathbf{W}_{i}^{Q} \times \left(K \mathbf{W}_{i}^{K} \right)^{T}}{\sqrt{d}} \right) V$$
(5)

where Q and K are BERT output and d denotes the dimension of the BERT output. With the above operation, we do not need to rely on external NLP toolkits, and avoid pruning operations.

To gain more information about the structure of large graphs, we refer to the dense connection [43]. With the help of dense connections, we can train deeper models, allowing rich local and non-local information to be captured for learning a better graph representa-

tion. The *p* is represented as the connection between the initial node representation and the layer 1, 2, ..., l - 1 node representation.

$$p_j^{(l)} = [v_j; h_j^{(1)}; h_j^{(2)}; \dots; h_j^{(l-1)}]$$
(6)

Each densely connected layer has L sub-layers. Each sub-layer has multi-layer GCNs. Our graph model concatenates the outputs of each sub-layer to form new representations. Therefore, unlike GCN models, where the hidden dimension is more than or equal to the input dimension, our graph module's hidden dimension diminishes as the number of layers grows, improving parameter efficiency.

Thus, for *N* dense connections and the resulting *N* attention-guided adjacency matrices, we obtain the output representation of the *t*-th graph according to the following equation:

$$h_{t_i}^{(l)} = \rho\left(\sum_{j=1}^n A_{ij}^{(t)} W_t^{(l)} p_j^{(l)} + b_t^{(l)}\right)$$
(7)

where t = 1, 2, ..., N and $A^{(t)}$ is *t*-th the attention guided adjacency matrix. $W_t^{(l)} \in \mathbb{R}^{d_{hidden} \times d^{(l)}}$ represents the weight matrix, where $d^{(l)} = d + d_{hidden} \times (l - 1)$.

Finally, the linear combination layer is applied to integrate the outputs of the N different densely connected layers. Formally, the output representation of the graph module after the linear combination layer can be written as

$$h_{comb} = [h^{(1)}, h^{(2)}, \dots, h^{(N)}]$$

$$h_{out} = W_{comb}h_{comb} + b_{comb}$$
(8)

where $h_{comb} \in \mathbb{R}^{d \times N}$ is obtained by connecting *N* densely connected layers, b_{comb} is a bias vector for the linear transformation, and $W_{comb} \in \mathbb{R}^{(d \times N) \times d}$ is the weight matrix.

3.3. GlobalPointer Joint Decoder

At first glance, joint extraction appears to be the extraction of triples (s, p, o) (i.e., subject, predicate, object), but it really is the extraction of quintuples (sh, st, p, oh, ot), where sh and st are the head and tail positions of subject entity, respectively, and oh and ot are the head and tail positions of object entity, respectively. Here, we can convert the triple in Figure 1 into a quintuple, as shown in Figure 4. From the perspective of the probability map, it is only necessary to design a quintet scoring function. However, if we count all quintuples one by one, the total is far too enormous. Suppose the sentence length is m and there are n relations, then the number of all quintuples is

$$n \times \frac{m(m+1)}{2} \times \frac{m(m+1)}{2} = \frac{1}{4}nm^2(m+1)^2$$
(9)

	Texts	Quintuples		
Normal	The capital of [China] is [Beijing]	(China, China, capital, Beijing, Beijing)		
SEO	[Stephen Chow] starred in "[The King of Comedy]" and "[A Chinese Odyssey]".	(Stephen, Chow, actor, The, Comedy) (Stephen, Chow, actor, A, Odyssey)		
EPO	"[Kung Fu]" directed and starring [Stephen Chow] will be released in the near future.	(Stephen, Chow, actor, Kung, Fu) (Stephen, Chow, director, Kung, Fu)		

Figure 4. Examples of overlapping quintuples are divided into normal, single entity overlap (SEO), and entity pair overlap (EPO).

It is tough for us to perform the aforementioned equation. As a result, a simplification is required. We may use the following decomposition strategy for this:

$$S(sh, st, p, oh, ot) = S(sh, st) + S(oh, ot) + S(sh, oh|p) + S(st, ot|p)$$
(10)

where S(sh, st) and S(oh, ot) denote the head and tail scores of the subject entity and object entity, respectively, and the subject entity and object entity are parsed out by S(sh, st) > 0 and S(oh, ot) > 0. S(sh, oh | p) means that the head features of subject and object are used as their representations to conduct one matching. The relation can be resolved when S(sh, oh | p) > 0. If there is entity nesting, it is necessary to consider S(st, ot | p).

S(sh, st), S(oh, ot) is used to identify the entity corresponding to the subject and object, and it is equivalent to having a named entity recognition (NER) task, so we can use a GlobalPointer [44] to complete it. S(sh, oh | p) is to identify the specific relationship p of (sh, oh) pair; here, we also use GlobalPointer to complete it, but to avoid the sh > oh to remove the default lower triangle mask of GlobalPointer; finally, S(st, ot | p) is the same as S(sh, oh | p).

3.4. Training and Prediction

The purpose of this paper's training is to use positive and negative samples to achieve model inference. As a result, we employ multi-label cross-entropy as the training loss function, which is defined as

$$l = \log\left(1 + \sum_{i \in P} e^{-S_i}\right) + \log\left(1 + \sum_{i \in \mathcal{N}} e^{-S_i}\right)$$
(11)

where P and N are the sets of positive and negative categories. The position of the positive category is 1 and the position of the negative category is 0. Thus, S_i denotes the *i*-th position in the label corresponding to the *i*-th value of the prediction.

However, there is a fatal error in the loss function defined in this way. The number of positive categories is much smaller than the number of negative categories. This results in a large cost for both creation and transmission. Therefore, the loss function for the above can be adjusted as follows:

$$L = \log\left(1 + \sum_{i \in P} e^{-S_i}\right) + \log\left(1 + \sum_{i \in A} e^{-S_i}\right) + \log\left(1 - e^{\log\left(\sum_{i \in P} e^{S_i}\right) - \log\left(1 + \sum_{i \in A} e^{-S_i}\right)}\right)$$
(12)

where $A = P \cup N$. After the above operation, the size of the label matrix is greatly reduced.

4. Experiment

4.1. Datasets

To validate the performance of the model, we opted to conduct experiments on two public datasets: NYT [45] and WebNLG [46]. The labels in both datasets are triples, and the entities in the triples are not the entire entity span, but the last word of the entity in the sentence. For example, for the entity New York, the entity record in the dataset is York. The dataset we use is from [32], and its statistical results are shown in Table 1.

Calagora	N	/T	Webl	NLG
Category	Train	Test	Train	Test
Normal	37,013	3266	1596	246
EPO	9782	978	227	26
SEO	14,735	1297	3406	457
ALL	56,195	5000	5019	703

Table 1. Statistics of datasets. Note that a sentence can belong to both EPO class and SEO class.

NYT: NYT is a widely used dataset for relation extraction tasks. This dataset is generated by aligning relations in freebase with the New York Times (NYT) corpus in a remotely supervised manner and it contains 24 predefined relations and 1.18M sentences. **WebNLG**: WebNLG is a general dataset for evaluating relation extraction models; the dataset uses triples in DBPedia, so it contains more relations, and it contains 246 predefined relations.

4.2. Evaluation Metrics

In our experiments, we maintain consistency with our previous work. For the extracted relationship triples, when the head of the subject entity and object entity and the relation is predicted correctly, we consider that the triple is correct. We follow common evaluation guidelines and report the standard micro precision (Prec.), recall (Rec.), and F1 score. (1) The precision indicates the proportion of correctly predicted positive samples to the actual predicted positive samples. (2) The recall denotes the proportion of correctly predicted positive samples to positive samples. (3) The F1 score is a summed average of precision and recall. The calculations are as follows:

$$Precision = \frac{TP}{TP+FP}$$

$$Recall = \frac{TP}{TP+FN}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision+Recall}$$
(13)

where *TP* means correctly predicting positive samples as positive, *FN* means wrongly predicting positive samples as negative, and *FP* means wrongly predicting negative samples as positive.

1

4.3. Implementation Details

Our model SGNet is implemented with PyTorch, the parameter optimization method uses AdamW [47], and the learning rate is 3×10^{-5} . The pretrained BERT model leverages the base-cased English model² and sets the maximum length of input sentences to 100. The batch size for training is set to 16 and the batch size for validation is 32 in the NYT and the WebNLG. The GPU used in the experiment is NVIDIA GeForce RTX 3090, each video memory is 24 G, the running memory is 46 G, and the programming language is Python3.7. We choose the model that performs best on the validation set, then feed the test set to that model to obtain the output. Furthermore, we keep the parameters in the graph module the same as in [40]. Our paper uses a two-layer graph module, and the first-layer graph module does not need to be guided by the attention mechanism.

4.4. Result and Analysis

4.4.1. Main Results

We compare the following baseline models with our proposed model:

- **NovelTagging** [35] applies a novel tagging method to transform the joint extraction of entities and relations into a sequence tagging problem, but it cannot tackle the overlap problem.
- **CopyRE** [32] uses seq2seq to generate all triples to solve the overlap problem for a sentence, but such an approach only considers a single token and not multiple tokens.
- **GraphRel** [24]: A model that generates a weighted relation graph for each relation type, and applies a GCN to predict relations between all entity pairs.
- OrderCopyRE [23]: An improved model of CopyRE that uses reinforcement learning to generate multiple triples.

- **ETL-Span** [36] decomposes the joint extraction task into two subtasks. The first subtask is to distinguish all head entities that may be related to the target relation, and the second subtask is to determine the corresponding tail entity and relation for each extracted head entity.
- **WDec** [48]: An improved model of CopyRE, which solves the problem that CopyRE misses multiple tokens.
- **CasRel** [26] identifies the head entity first and then the tail entity under a particular relationship.
- **DualDec** [49] designs an efficient cascaded dual-decoder approach to address the extraction of overlapping relation triplets, which consists of a text-specific relation decoder and a relation-corresponding entity decoder.
- **RMAN** [50] not only considers the semantic features in the sentence but also leverages the relation type to label the entities to obtain a complete triple.

Table 2 reports the main results of our model and the other baseline models on both datasets. As can be seen from the table, our model outperforms all baseline models in the F1 score. For the NYT dataset, our model achieves the best results in recall and F1 scores, with F1 scores improving by 1.7%, 0.8%, and 5.9% over the CasRel framework, DualDec model, and RSAN model, respectively. For the WebNLG dataset, our model performs well, with F1 scores improving by 0.1%, 1.0%, and 7.4% over the CasRel framework, DualDec model, and RSAN model, respectively.

NC 11		NYT			WebNLG	
Model	Prec.	Rec.	F1	Prec.	Rec.	F1
NovelTagging	62.4	31.7	42.0	52.5	19.3	28.3
CopyRE _{OneDecoder}	61.0	56.6	58.7	37.7	36.4	37.1
CopyRE _{MultiDecoder}	61.0	56.6	58.7	37.7	36.4	37.1
GraphRel _{1p}	62.9	57.3	60.0	42.3	39.4	37.1
GraphRel _{2p}	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	84.9	72.3	78.1	84.0	91.5	87.6
WDec	94.5	76.2	84.4	-	-	-
CasRel	89.7	89.5	89.6	93.4	90.1	91.8
DualDec	90.2	90.9	90.5	90.3	91.5	90.9
RMAN	87.1	83.8	85.4	83.6	85.3	84.5
SGNet _{WG}	90.5	89.8	90.2	90.6	90.0	90.2
SGNet	91.2	91.4	91.3	91.8	91.9	91.9

Table 2. Main results. Bold marks the highest score. SGNet_{WG} does not contain graph modules.

Bold data in the table indicates the best results.

In the experiment, to verify the effectiveness of the graph module and the joint decoding module, we removed the graph module and only kept the BERT model and the joint decoding module, which were named SGNet_{WG}. For the NYT dataset, the F1 score of the SGNet_{WG} model is 0.6% higher than that of CasRel, which only uses the BERT module and the decoding module. Furthermore, compared with SGNet_{WG}, our model improves by 1.1% and 1.7% on the NYT and WebNLG, respectively. These encouraging results show that the graph module and joint decoding module can effectively assist the model to extract the relation triples. Although SGNet_{WG} does not perform very well for the WebNLG dataset, SGNet performs better.

Previous studies have shown that it is already difficult to improve the F1 score obtained by the model on the dataset when it reaches 90+, which is close to the human limit [26]. Therefore, our model's results on the WebNLG dataset are close to those of the CasRel model. On the other hand, for the WebNLG dataset itself, the small amount of training data (5019) makes it particularly difficult to extract 246 predefined relations.

4.4.2. Result Analysis on Different Sentence Types

Since overlapping triple extraction is a challenging problem for joint extraction, the majority of the models in Table 2 have low F1 scores. Specifically, the statistics in Table 1 show that most of the sentences in the NYT dataset belong to the normal class, while in the WebNLG dataset, most of the sentences belong to the EPO and SEO classes. This fails most previous models to extract overlapping triples.

To explore the performance of our model on extracting triples, we extracted triplets from sentences with different triplets and compared them with baseline models, and the results are shown in Table 3. As seen in the table, the majority of the baseline models on both datasets exhibit a diminishing trend in their capacity to extract triples as the number of triples (*N*) in the phrase grows, whereas our model shows a rising trend. At N = 4, our model surpasses the DualDec model, which has the highest F1 score of the baseline models, while both show an upward trend. Our model outperforms the DualDec model in F1 scores for the NYT dataset, with F1 scores improving by 0.7%, 0.8%, 0.3%, 0.4%, and 0.5%, respectively. Compared with DualDec, our model improves by 3.6%, 0.6%, 4.8%, 2.5%, and 6.3% on the WebNLG dataset, respectively. It shows that our model can deal with complex multi-relationship problems in sentences.

Table 3. F1 score of extracting relational triples from sentences with a different number (denoted as *N*) of triples.

			NYT					WebNLG		
Method	N = 1	<i>N</i> = 2	<i>N</i> = 3	N = 4	$N \ge 5$	N = 1	<i>N</i> = 2	<i>N</i> = 3	N = 4	$N \ge 5$
CopyRE _{OneDecoder}	66.6	52.6	49.7	48.7	20.3	65.2	33.0	22.2	14.2	13.2
CopyRE _{MultiDecoder}	67.1	58.6	52.0	53.6	30.0	59.2	42.5	31.7	24.2	30.0
GraphRel _{1p}	69.1	59.5	54.4	53.9	37.5	63.8	46.3	34.7	30.8	29.4
GraphRel _{2p}	71.0	61.5	57.4	55.1	41.1	66.0	48.3	37.0	32.1	32.1
OrderCopyRE	71.7	72.6	72.5	77.9	45.9	63.4	62.2	64.4	57.2	55.7
ETL-Span	85.5	82.1	74.7	75.6	76.9	82.1	86.5	91.4	89.5	91.1
CasRel	88.2	90.3	91.9	94.2	83.7	89.3	90.8	94.2	92.4	90.9
DualDec	88.5	90.8	92.4	95.5	90.1	85.8	90.5	88.9	89.9	85.4
RMAN	84.3	86.0	86.6	92.5	76.1	-	-	-	-	-
SGNet	89.2	91.6	92.7	95.9	90.6	89.4	91.1	93.7	93.4	91.7

Bold data in the table indicates the best results.

Further, to investigate the performance of our model in extracting overlapping relational triples, we extracted triples of different overlapping triples types from the sentences. According to different overlapping categories, the sentences used in the test can be divided into three types, including normal, SEO, and EPO. As shown in Figure 5, our method achieves the best result. For normal class, i.e., not containing overlapping relationships, our model has a 0.7% higher F1 score than the highest baseline model on the NYT dataset and outperforms most of the baseline models on the WebNLG dataset, and its F1 score was only 0.2% lower than CasRel. For single entity overlap (SEO) and entity pair overlap (EPO), the F1 score of our model is higher than the best baseline model on both datasets. Compared to the baseline model with the highest F1 score, our model improves by 0.7% and 0.3%on the SEO class of NYT and WebNLG, respectively. Additionally, our model improves by 0.6% and 0.3% for the EPO class. Compared to the CasRel model, the SGNet_{WG} model improves by 0.2% for the NYT dataset and decreases by 0.2% for the WebNLG dataset in the SEO class, and it improves by 0.1% for NYT dataset and 0.2% for the WebNLG dataset in EPO class. Similarly, for SGNet_{WG}, compared with SGNet, the improvement of SGNet model is small, but higher than the CasRel model. Since the CasRel model has a certain effect in dealing with overlapping triples, and the effect of the SGNet model is due to this, our model is more capable of dealing with overlapping triples. Moreover, as we described earlier, when the model F1 score reaches 90+, it has saturated and there is very little room for improvement.



Figure 5. F1 score of extracting relational triples from sentences with the different overlapping patterns. (**a**) F1 of Normal Class, (**b**) F1 of SEO Class, (**c**) F1 of EPO Class.

4.4.3. Case Study

Table 4 shows the case study of our proposed SGNet. Due to that our model decodes by decomposing triples into quintuples, the output of the model is also triples. For the selection of sentences in the table, we divided into normal and overlapping cases. First, for the first sentence, the sentence contains only normal triples, and our model is able to extract them. The second example exists where the SEO and EPO triples overlap. As shown in the table, our model can completely extract the overlapping triples in the sentence, which shows the effectiveness of our model.

Table 4. Case study for SGNet.

Sentence	SGNet		
Barcelona will discharge Ronaldinho to Brazil, Deco to Portugal and the young star Lionel Messi to Argentina.	(Ronaldinho, person, Brazil) (Messi, person, Argentina)		
Ms. Rice met with China's leaders in Beijing in March specifically to ask them to pressure North Korea.	(Beijing, administrative_division, China) (China, location, Beijing) (China, country, Beijing)		

4.4.4. Model Efficiency

We evaluate the model efficiency according to the *Training Time* and *Inference Time* of the model in two datasets NYT and WebNLG, and take the similar decoding model TPLinker [37] as the comparison model. The results are recorded in Table 5. From the table we can observe that without using the graph module our model is better than TPLinker in terms of parallel processing. This encouraging result is due to the multi-label loss function of the model we mentioned earlier. As we said before, there are far fewer positive classes than negative classes, so the dimension of the label matrix is greatly reduced, which improves the training speed.

Table 5. Comparison of the model efficiency. Training time(s) means the time required to train one epoch, and inference time (ms) is the time to predict triples of one sentence. Our re-implementation is marked by *.

Dataset	Model	Training Time	Inference Time	F1
	TPLinker *	1592	46.2	90.6
NYT	SGNet _{WG}	1390	43.2	90.2
	SGNet	2165	69.6	91.3
	TPLinker *	599	40.1	90.9
WebNLG	SGNet _{WG}	142	37.4	90.2
	SGNet	631	63.4	91.9

Bold data in the table indicates the best results.

In addition, it can be seen from the results that due to the introduction of the graph module into our model, the F1 score of the model is improved, but the negative effect is that the training time and inference time are longer. However, the training time and inference time of the two models are not much different. Therefore, in the case of increasing the number of parameters of the model, the efficiency of our model is better than that of TPLinker.

5. Discussion

In the introduction of our paper, we first analyze the pipeline-based approach, which leads to the joint learning approach since the pipeline approach ignores the interaction between entities and relations and is prone to error propagation. Our model is also based on the joint learning approach. First of all, we consider a joint decoding approach to decompose triples into quintuples to strengthen the information exchange between tasks and prevent the problem of information imbalance in the feature extraction stage. In experiments, the F1 score of SGNet_{WG} is higher than most of the models by comparing with other models, and in the exploration of triple overlapping experiments, the F1 score of SGNet_{WG} is higher than the best baseline model. Next, in order to better learn the semantic relationship information of sentences, we introduce a graph neural network. Modeling text as a graph, nodes can better pick up local and non-local information. In the experimental results, it can be seen that the F1 score of SGNet_{WG} is improved compared with SGNet.

Finally, due to the deepening of the model depth, the training time and inference time of our model are longer in the analysis of the model software complexity, but it is acceptable. In exchange for this, the advantage is that the extraction accuracy is improved.

6. Conclusions

We investigate extracting text features via graph operations since unstructured text belongs to non-Euclidean data and graphs can better represent such non-Euclidean data. However, recent research indicates that the pruning process may destroy some vital information in the complete tree; hence, the approach taken in this article is to construct a fully linked graph. Furthermore, traditional graphs are built by external toolkits, and they are fixed and cannot be modified. For the graph generated by the graph generator, the parameters of the generator can be optimized by training to make it have a certain generalization ability. Because conventional GCN only acquires local node features and frequently ignores global text features, we adopt a dense connection layer in this research to obtain additional global information. Furthermore, to solve the problem that the model has difficulty extracting overlapping triples, we use the GlobalPointer joint decoding method. The extraction of triples is subdivided into quintuple extraction. The extraction of entity head and entity tail under a specific relationship will be transformed into a form similar to scaled dot-product attention so that the overlapping entities can be identified with multiple specific relationship matrices, and then the overlapping problem can be tackled. The experiment results on the NYT and WebNLG datasets demonstrated the effectiveness of our model. In addition, our experiments show that our model can effectively extract overlapping triples and handle complex sentence multi-relation problems.

In the future, we will further improve the performance of this model to apply it in knowledge graphs or other fields.

Author Contributions: Conceptualization, J.L.; methodology, J.L.; software, J.L.; validation, J.L., Q.H., D.Z. and S.F.; formal analysis, Q.H. and D.Z.; investigation, J.L.; resources, Q.H. and D.Z.; writing—original draft preparation, J.L.; writing—review and editing, J.L., Q.H. and S.F.; visualization, J.L. All authors have read and agreed to the published version of the manuscript.

Funding: The research was funded by the National Natural Science Foundation of China "Research on the Evidence Chain Construction from the Analysis of the Investigation Documents (62166006)", supported by Guizhou Provincial Science and Technology Projects (Guizhou Science Foundation-ZK [2021] General 335).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The study did not report any data.

Conflicts of Interest: There are no conflict to declare.

References

- Liu, Q.; Li, Y.; Duan, H.; Liu, Y.; Qin, Z.G. Knowledge Graph Construction Techniques. J. Comput. Res. Dev. 2016, 53, 582–600. [CrossRef]
- Auer, S.; Bizer, C.; Kobilarov, G.; Lehmann, J.; Cyganiak, R.; Ives, Z. DBpedia: A nucleus for a Web of open data. In Proceedings of the 6th International Semantic Web Conference, ISWC 2007 and 2nd Asian Semantic Web Conference, ASWC 2007, Busan, Korea, 11–15 November 2007; pp. 722–735. [CrossRef]
- Bollacker, K.; Evans, C.; Paritosh, P.; Sturge, T.; Taylor, J. Freebase: A collaboratively created graph database for structuring human knowledge. In Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data 2008, SIGMOD'08, Vancouver, BC, Canada, 9–12 June 2008; pp. 1247–1249. [CrossRef]
- 4. Carlson, A.; Betteridge, J.; Kisiel, B.; Settles, B.; Hruschka, E.R., Jr.; Mitchell, T.M. Toward an architecture for never-ending language learning. In Proceedings of the AAAI-10/IAAI-10-Proceedings of the 24th AAAI Conference on Artificial Intelligence and the 22nd Innovative Applications of Artificial Intelligence Conference, Atlanta, GA, USA, 11–15 July 2010; pp. 1306–1313. Available online: https://ojs.aaai.org/index.php/AAAI/article/view/7519 (accessed on 5 January 2010).
- Wu, W.; Li, H.; Wang, H.; Zhu, K.Q. Probase: A probabilistic taxonomy for text understanding. In Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data, SIGMOD '12, Scottsdale, AZ, USA, 21–24 May 2012; pp. 481–492. [CrossRef]
- 6. Vrandei, D.; Krotzsch, M. Wikidata: A free collaborative knowledgebase. Commun. ACM 2014, 57, 78–85. [CrossRef]
- Han, X.; Liu, Z.; Sun, M. Neural knowledge acquisition via mutual attention between knowledge graph and text. In Proceedings of the 32nd AAAI Conference on Artificial Intelligence, AAAI 2018, New Orleans, LA, USA, 2–7 February 2018; pp. 4832–4839. Available online: https://ojs.aaai.org/index.php/AAAI/article/view/11927 (accessed on 26 April 2018).
- Luan, Y.; He, L.; Ostendorf, M.; Hajishirzi, H. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 Octomber–4 November 2018; pp. 3219–3232. [CrossRef]
- Qian, L.; Zhou, G.; Kong, F.; Zhu, Q.; Qian, P. Exploiting constituent dependencies for tree kernel-based semantic relation extraction. In Proceedings of the 22nd International Conference on Computational Linguistics, Manchester, UK, 18–22 August 2008; Volume 1, pp. 697–704. [CrossRef]
- 10. Zelenko, D.; Aone, C.; Richardella, A. Kernel Methods for Relation Extraction. J. Mach. Learn. Res. 2003, 3, 1083–1106. [CrossRef]
- 11. Nadeau, D.; Sekine, S. A Survey of Named Entity Recognition and Classification. Lingvisticae Investig. 2007, 30, 3–26. [CrossRef]
- 12. Li, J.; Sun, A.; Han, J.; Li, C. A Survey on Deep Learning for Named Entity Recognition. *IEEE Trans. Knowl. Data Eng.* 2022, 34, 50–70. [CrossRef]
- 13. Zhang, Y.; Lu, Z. Exploring semi-supervised variational autoencoders for biomedical relation extraction. *Methods* **2019**, *166*, 112–119. [CrossRef]
- Zhang, Y.; Qi, P.; Manning, C.D. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018; pp. 2205–2215. [CrossRef]
- Dai, D.; Xiao, X.Y.; Lyu, Y.J.; Dou, S.; She, Q.Q.; Wang, H.F. Joint Extraction of Entities and Overlapping Relations Using Position-Attentive Sequence Labeling. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence/31st Innovative Applications of Artificial Intelligence Conference/9th AAAI Symposium on Educational Advances in Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 6300–6308. [CrossRef]
- Ren, X.; Wu, Z.; He, W.; Qu, M.; Voss, C.R.; Ji, H.; Abdelzaher, T.F.; Han, J. CoType: Joint Extraction of Typed Entities and Relations with Knowledge Bases. In Proceedings of the 26th International Conference on World Wide Web (WWW), Perth, Australia, 3–7 May 2017; pp. 1015–1024. [CrossRef]
- 17. Li, Q.; Ji, H. Incremental Joint Extraction of Entity Mentions and Relations. In Proceedings of the 52nd Annual Meeting of the Association-for-Computational-Linguistics (ACL), Baltimore, MD, USA, 22–27 June 2014; pp. 402–412.
- Yu, X.; Lam, W. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In Proceedings of the 23rd International Conference on Computational Linguistics: Posters, Beijing, China, 23–27 August 2010; pp. 1399–1407.
- Miwa, M.; Sasaki, Y. Modeling Joint Entity and Relation Extraction with Table Representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1858–1869. [CrossRef]
- Zheng, S.C.; Hao, Y.X.; Lu, D.Y.; Bao, H.Y.; Xu, J.M.; Hao, H.W.; Xu, B. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing* 2017, 257, 59–66. [CrossRef]
- Gupta, P.; Schütze, H.; Andrassy, B. Table Filling Multi-Task Recurrent Neural Network for Joint Entity and Relation Extraction. In Proceedings of the COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; pp. 2537–2547. Available online: https://aclanthology.org/C16-1239 (accessed on 11 December 2016).
- Tan, Z.; Zhao, X.; Wang, W.; Xiao, W.D. Jointly Extracting Multiple Triplets with Multilayer Translation Constraints. In Proceedings of the 33rd AAAI Conference on Artificial Intelligence/31st Innovative Applications of Artificial Intelligence Conference/9th AAAI Symposium on Educational Advances in Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 7080–7087. [CrossRef]

- Zeng, X.; He, S.; Zeng, D.; Liu, K.; Zhao, J. 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 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 367–377. [CrossRef]
- Fu, T.J.; Li, P.H.; Ma, W.Y. 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 (ACL), Florence, Italy, 28 July–2 August 2019; pp. 1409–1418. [CrossRef]
- 25. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, 24–26 April 2017.
- Wei, Z.P.; Su, J.L.; Wang, Y.; Tian, Y.; Chang, Y.; Assoc Computat, L. A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. In Proceedings of the 58th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Electr Network, 5–10 July 2020; pp. 1476–1488. [CrossRef]
- Chan, Y.S.; Roth, D. Exploiting syntactico-semantic structures for relation extraction. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; Volume 1, pp. 551–560. Available online: https://aclanthology.org/P11-1056 (accessed on 1 January 2011).
- 28. GuoDong, Z.; Jian, S.; Jie, Z.; Min, Z. Exploring various knowledge in relation extraction. In Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Ann Arbor, MI, USA, 25–30 June 2005; pp. 427–434. [CrossRef]
- Wang, S.; Zhang, Y.; Che, W.; Liu, T. Joint extraction of entities and relations based on a novel graph scheme. In Proceedings of the 27th International Joint Conference on Artificial Intelligence, IJCAI 2018, Stockholm, Sweden, 13–19 July 2018; pp. 4461–4467. [CrossRef]
- Miwa, M.; Bansal, M. 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 (ACL), Berlin, Germany, 7–12 August 2016; pp. 1105–1116. [CrossRef]
- Katiyar, A.; Cardie, C. Going out on a limb: Joint Extraction of Entity Mentions and Relations without Dependency Trees. In Proceedings of the 55th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 917–928. [CrossRef]
- Zeng, X.R.; Zeng, D.J.; He, S.Z.; Liu, K.; Zhao, J. 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 (ACL), Melbourne, Australia, 15–20 July 2018; pp. 506–514. [CrossRef]
- Katiyar, A.; Cardie, C. Investigating LSTMs for joint extraction of opinion entities and relations. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, Berlin, Germany, 7–12 August 2016; pp. 919–929. [CrossRef]
- Zhang, M.; Zhang, Y.; Fu, G. End-to-end neural relation extraction with global optimization. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, 9–11 September 2017; pp. 1730–1740. [CrossRef]
- Zheng, S.C.; Wang, F.; Bao, H.Y.; Hao, Y.X.; Zhou, P.; Xu, B. 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 (ACL), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1227–1236. [CrossRef]
- Yu, B.W.; Zhang, Z.Y.; Shu, X.B.; Liu, T.W.; Wang, Y.B.; Wang, B.; Li, S.J. Joint Extraction of Entities and Relations Based on a Novel Decomposition Strategy. In Proceedings of the 24th European Conference on Artificial Intelligence (ECAI), European Assoc Artificial Intelligence, Electr Network, 29 August–8 September 2020; pp. 2282–2289. [CrossRef]
- Wang, Y.; Yu, B.; Zhang, Y.; Liu, T.; Zhu, H.; Sun, L. 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), 13–18 September 2020; pp. 1572–1582. [CrossRef]
- Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017.
- Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. 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, NAACL HLT 2019, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
- Xue, F.Z.; Sun, A.X.; Zhang, H.; Chng, E.S. GDPNet: Refining Latent Multi-View Graph for Relation Extraction. In Proceedings of the 35th AAAI Conference on Artificial Intelligence/33rd Conference on Innovative Applications of Artificial Intelligence/11th Symposium on Educational Advances in Artificial Intelligence, Electr Network, 2–9 February 2021; pp. 14194–14202.
- 41. Lee, Y.; Lee, Y. Toward scalable internet traffic measurement and analysis with Hadoop. *SIGCOMM Comput. Commun. Rev.* 2012, 43, 5–13. [CrossRef]
- Guo, Z.J.; Zhang, Y.; Lu, W. Attention Guided Graph Convolutional Networks for Relation Extraction. In Proceedings of the 57th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Florence, Italy, 28 July–2 August 2019; pp. 241–251. [CrossRef]
- Huang, G.; Liu, Z.; van der Maaten, L.; Weinberger, K.Q. Densely Connected Convolutional Networks. In Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 2261–2269. [CrossRef]

- 44. Su, J.L. GlobalPointer: Deal with Nested and Non-Nested NER in a Unified Way. 2021. Available online: https://spaces.ac.cn/ archives/8373 (accessed on 1 May 2021).
- Riedel, S.; Yao, L.; McCallum, A. Modeling relations and their mentions without labeled text. In *Machine Learning and Knowledge Discovery in Databases, Proceedings of the European Conference, ECML PKDD 2010, Barcelona, Spain, 19–23 September 2010*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 148–163. [CrossRef]
- Gardent, C.; Shimorina, A.; Narayan, S.; Perez-Beltrachini, L. Creating Training Corpora for NLG Micro-Planning. In Proceedings of the 55th Annual Meeting of the Association-for-Computational-Linguistics (ACL), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 179–188. [CrossRef]
- 47. Loshchilov, I.; Hutter, F. Decoupled weight decay regularization. arXiv 2019, arXiv:1711.05101.
- Nayak, T.; Ng, H.T. Effective Modeling of Encoder-Decoder Architecture for Joint Entity and Relation Extraction. In Proceedings of the 34th AAAI Conference on Artificial Intelligence/32nd Innovative Applications of Artificial Intelligence Conference/10th AAAI Symposium on Educational Advances in Artificial Intelligence, New York, NY, USA, 7–12 February 2020; pp. 8528–8535. [CrossRef]
- 49. Ma, L.; Ren, H.; Zhang, X. Effective Cascade Dual-Decoder Model for Joint Entity and Relation Extraction. *arXiv* 2021, arXiv:2106.14163.
- Lai, T.; Cheng, L.; Wang, D.; Ye, H.; Zhang, W. RMAN: Relational multi-head attention neural network for joint extraction of entities and relations. *Appl. Intell.* 2022, 52, 3132–3142. [CrossRef]