applied sciences

MDPI

Article

BTDM: A Bi-Directional Translating Decoding Model-Based Relational Triple Extraction

Zhi Zhang, Junan Yang *, Hui Liu and Pengjiang Hu

College of Electronic Engineering, National University of Defense Technology, Hefei 230037, China; zhangzhi@nudt.edu.cn (Z.Z.)
* Correspondence: yangjunan@ustc.edu.cn

Abstract: The goal of relational triple extraction is to extract knowledge-rich relational triples from unstructured text. Although the previous methods obtain considerable performance, there are still some problems, such as error propagation, the overlapping triple problem, and suboptimal subject–object alignment. To address the shortcomings above, in this paper, we decompose this task into three subtasks from a fresh perspective: entity extraction, subject–object alignment and relation judgement, as well as propose a novel bi-directional translating decoding model (BTDM). Specifically, a bidirectional translating decoding structure is designed to perform entity extraction and subject–object alignment, which decodes entity pairs from both forward and backward extraction. The bidirectional structure effectively mitigates the error propagation problem and aligns the subject–object pairs. The translating decoding approach handles the overlapping triple problem. Finally, a (entity pair, relation) bipartite graph is designed to achieve practical relationship judgement. Experiments show that our model outperforms previous methods and achieves state-of-the-art performance on NYT and WebNLG. We achieved F1-scores of 92.7% and 93.8% on the two datasets. Meanwhile, in various complementary experiments on complex scenarios, our model demonstrates consistent performance gain in various complex scenarios.

Keywords: entity and relation extraction; information extraction; nature language processing; error propagation; overlapping triples

check for updates

Citation: Zhang, Z.; Yang, J.; Liu, H.; Hu, P. BTDM: A Bi-Directional Translating Decoding Model-Based Relational Triple Extraction. *Appl. Sci.* **2023**, *13*, 4447. https://doi.org/10.3390/app13074447

Academic Editors: Chuan-Ming Liu and Wei-Shinn Ku

Received: 12 March 2023
Revised: 30 March 2023
Accepted: 30 March 2023
Published: 31 March 2023

Copyright: © 2023 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/).

1. Introduction

One of the most important information extraction tasks in natural language processing is extracting entities and relations from unstructured text to generate structured relational triples. Typically, the structured triple takes the form (subject, relation, object), where both the subject and the object are entities tied together by a relation. Extracting the triple of structured knowledge from an unstructured text can serve downstream tasks such as knowledge graphs [1,2], question answering [3,4], and biomedical tasks [5,6].

Early studies [7–10] on relational triple extraction split the work into two parts, identifying entities [11,12] and predicting their relations [7,13]. Specifically, the first step is identifying all entities in the text, then enumerating entities for relation judgement. This method is known as the pipeline-based approach. The advantage of this strategy is that it may leverage existing techniques to identify the named entity and classify the relations. However, there are two drawbacks as follows: (1) the link between entity identification and relation prediction is disregarded, and (2) errors in one subtask can easily propagate to other tasks.

To solve the above shortcomings, joint extraction [14–16], which is the simultaneous extraction of entities and relationships, is studied. CasRel [17] identifies all the subjects and then identifies the relations and objects related to the subjects. TDEER [18] extracts the possible relations and subjects, respectively, then decodes the objects using an attention mechanism. PRGC [19] first identifies the relation and entity pairs and then aligns them.

These sequence labeling approaches make use of multiple tagging sequence networks to determine the beginning and ending positions of entities.

Although the above methods have achieved a promising performance, they have a common feature in that they extract subjects before objects and relations or relations before entities. This method has a fatal drawback in that the subject extraction failure will lead to the entire triple being impossible to extract. Specifically, as shown in Figure 1 for the triple *(Stephen Chow, Nationality, China)*, failure to extract the subject *Stephen Chow* will lead to the failure to extract the object *China* and relation *Nationality*. The extraction of triples is sensitive to whether the subject is extracted, a problem that exists in most current methods. The problem is detrimental to the identification of entity pairs; furthermore, it significantly impairs the performance of the whole relational triple-extraction task.

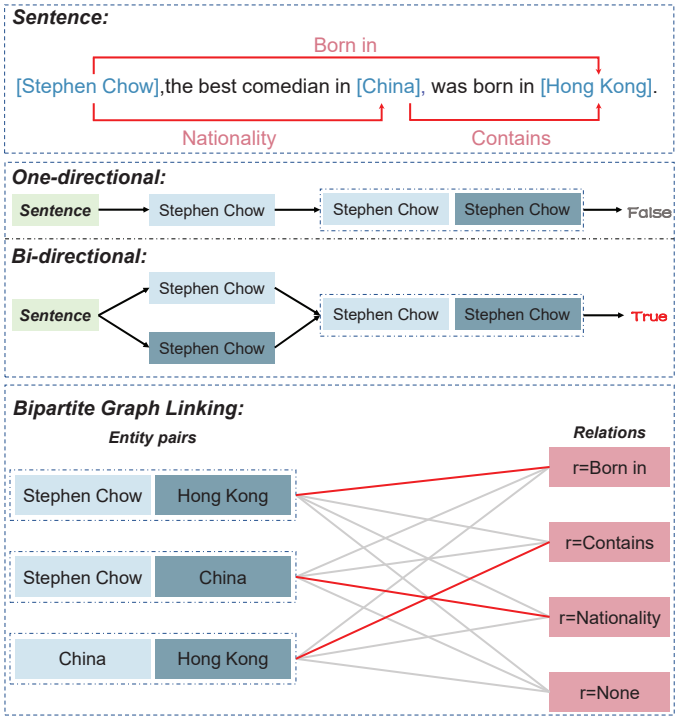


Figure 1. An example of single-directional and bidirectional extraction, as well as the bipartite graph that links entity pairs and relationships.

In this paper, to mitigate error propagation, we designed a bidirectional framework to identify entity pairs. We gave the subject and object equal importance in this framework, which extracts entity pairs using forward extraction (subject $\rightarrow$ entity pair, denoted as *s2p*) and backward extraction (object $\rightarrow$ entity pair, denoted as *o2p*), respectively. Meanwhile, the features in both directions flow into each other, which enhances the convergence speed and recognition accuracy. Secondly, to achieve the efficient and accurate alignment of entity pairs, a translating decoding approach is presented based on the attention mechanism. This approach decodes entity pairs by translating relational features and the results of entity recognition. Finally, to fully mine the relations between entity pairs, we add relational features to the bidirectional decoding to strengthen the interaction between entity pairs and relations. We also design a (entity pair, relation) bipartite graph, which maintains a relation matrix for each relation, thus effectively implementing a mapping of entity pairs to relationships.

We implemented the above approach in BTDM. This consists of an encoder module, a bidirectional entity, and alignment of the subject–object module and relation judgement module. According to the experimental results of two widely used benchmark datasets, the proposed BTDM outperforms earlier techniques and reaches state-of-the-art performance. The main contributions of this paper are as follows:

1. We propose a new bidirectional perspective to decompose the relational triple-extraction task into three subtasks: entity extraction, subject–object alignment, and relation judgement.
2. Following our perspective, we propose a novel end-to-end model BTDM, which greatly mitigates error propagation, handles the overlapping triple problem, and efficiently aligns the subject and object.
3. We conduct extensive experiments on several public NYT and WebNLG datasets. We compared the proposed method with 12 baselines, demonstrating that the proposed model achieves a state-of-the-art performance.

2. Problem Definition

In this section, we first provide a principled problem definition and present our view of the joint relational triple-extraction task.

2.1. Problem Formulation

Given a sentence $S = \{x_1, x_1, x_3, \ldots, x_n\}$ with n tokens, x_i denotes the i-th token in sequence S. The goal of the relation extraction model is to identify all the structured relational triples $T(S) = \{(s, r, o)|s, o \in E, r \in R\}$, where $E = \{e_1, e_2, \ldots, e_m\}$ is a set of entities, and $R = \{r_1, r_2, \ldots, r_k\}$ is a predefined set of relations.

2.2. Our View of the Problem

In this paper, the task of joint relational triple extraction is divided into three subtasks:

Entity Extraction. For the given sentence S, this subtask identities all subjects and objects separately. The output of this subtask are the subjects set $Y_s(S) = \{s_i\}$ and objects set $Y_o(S) = \{o_i\}$.

Subject–Object Alignment. For the given sentence S and the identified set of subjects $Y_s(S)$ and set of objects $Y_o(S)$, this subtask identifies all objects that correspond to the subjects set $Y_s(S)$ and the subjects that correspond to the objects set $Y_o(S)$. The output of this subtask are two entity pair sets $Y_{s2p}(S, s_i|s_i \in Y_s(S)) = \{(s_i, o_j)\}$ and $Y_{o2p}(S, o_i|o_i \in Y_o(S)) = \{(s_j, o_i)\}$. Then, the model can obtain the final identification of the entity pairs set $Y_{(s,o)}(S) = \{(s, o)\}$ by aligning the two sets $Y_{s2o}(S, s_i|s_i \in Y_s(S))$ and $Y_{o2s}(S, o_i|o_i \in Y_o(S))$.

Relation Judgement. For the given sentence S and identified set of entity pairs $Y_{(s,o)}(S)$, this subtask forms all possible triples by identifying all possible relations for each entity pair (s, o) in the set $Y_{(s,o)}(S) = \{(s, o)\}$. The output of this task is $T(S) = \left\{(s, r, o)|(s, o) \in Y_{(s,o)}(S), r \in R\right\}$.

3. Method

In this section, each component of the BTDM model is elaborated. Figure 2 depicts an overview of BTDM.

3.1. BTDM Encoder

The BTDM encoder takes as input a sentence S with n tokens. To obtain a sentence contextual representation $\boldsymbol{h}_i$ for each token, we generated an initial representation with a large-scale pre-trained language model (PLM). We utilized the BERT [20] encoder so that our comparisons were consistent, but alternative encoders, such as Glove [21] or RoBERTa [22], would work just as well. The detailed operations are as follows:

$$\begin{aligned} \boldsymbol{h} &= \left\{\boldsymbol{h}_1, \boldsymbol{h}_2, ..., \boldsymbol{h}_n | \boldsymbol{h}_i \in \mathbb{R}^{d \times 1}\right\} \\ &= \text{PLM}(\{x_1, x_2, ..., x_n\}) \end{aligned} \tag{1}$$

where d is the embedding dimension, $\boldsymbol{h}_i$ is the representation of i-th token in sentence S, and $\boldsymbol{h}$ is the contextual representation. Previous excellent works, such as CasRel [17] or SPN [23] PRGC [19], all used the single text feature h to identify triples. However, we believe that each of the three elements of the triple should have a unique set of features.

We obtained the relational embedding and added it to the state of subject–object alignment, which also helps to strengthen the interaction between entity pairs and relations. For this reason, we further extracted subject features, object features, and relation features on the basis of text features $\boldsymbol{h}$. Notably, $\boldsymbol{h}$ is fed into three different feed-forward networks(FFN) to generate subject features, object features, and relation features (denoted as $\boldsymbol{hs}$, $\boldsymbol{ho}$, $\boldsymbol{hr}$, respectively). The detailed operations are as follows:

$$\begin{aligned} \boldsymbol{hs} &= \text{FFN_S}(\boldsymbol{h}) \\ \boldsymbol{hr} &= \text{FFN_R}(\boldsymbol{h}) \\ \boldsymbol{ho} &= \text{FFN_O}(\boldsymbol{h}) \end{aligned} \tag{2}$$

3.2. BTDM Decoder

In this section, we will introduce the BTDM decoder, which consists of three parts corresponding to three subtasks: entity extraction, subject–object alignment, and relation judgement.

3.2.1. Entity Extraction

Our proposed BTDM is a bidirectional extraction framework. Due to the space limitation, we only introduce the details of BTDM in the *s2p* direction, and the details of the model in the backward direction *o2p* are similar.

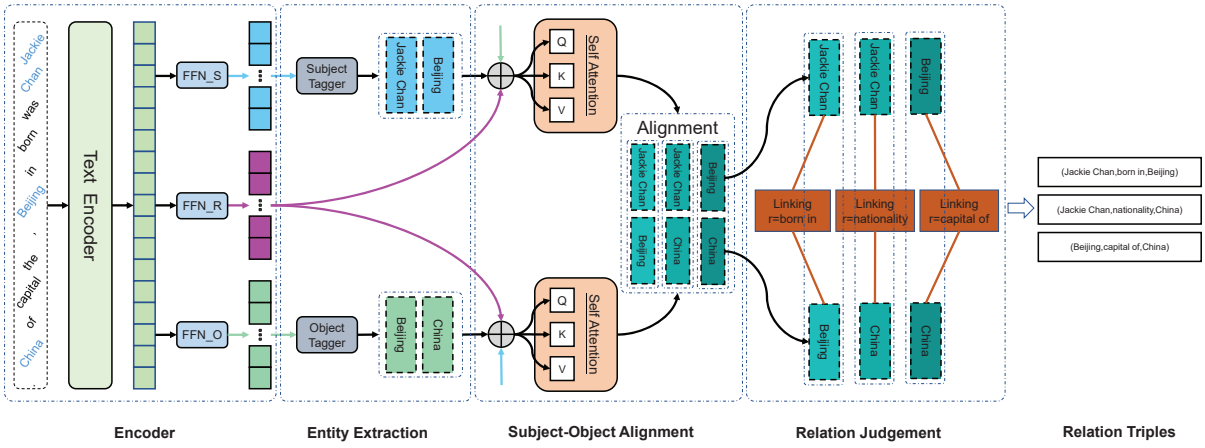


Figure 2. The structure of our the proposed approach, displaying the processing for handling one sentence that contains three relational triples *(Jackie Chan, born in, Beijing)*, *(Jackie Chan, nationality, China)*, and *(Beijing, capital of, China)*. In this example, BTDM first identifies entity pairs by translating and decoding in both directions, and then aligns them. Then, the relation is judged by linking a (entity pairs–relations) bipartite graph. Please note that all invalid links have been removed for the sake of clarity.

The subject tagger module extracted the subject for the *s2p* direction. Following prior research [17,18], we employed the span-based tagging strategy to effectively identify the subject and its position. In particular, the entity extraction module employed two binary classifiers to identify the start and end positions of the subject by assigning a binary(0/1) tag to each token. The binary tag indicates whether the current token position is the start or the end of a subject. If the token at the present position is the start or end of a subject, then Tag 1 is assigned; otherwise, Tag 0 is assigned. The detailed operations of the entity extraction on each token in the sentence are as follows:

$$\begin{aligned} p_i^{s_start} &= \sigma(\boldsymbol{W}_{s_start}\boldsymbol{hs}_i + \boldsymbol{b}_{s_start}) \\ p_i^{s_end} &= \sigma(\boldsymbol{W}_{s_end}\boldsymbol{hs}_i + \boldsymbol{b}_{s_end}) \end{aligned} \tag{3}$$

where $p_i^{s_start}$ and $p_i^{s_end}$ represent the possibility that the i-th token in a sentence is the start or end position of a subject, respectively. If the possibility exceeds the threshold, we set the predefined value; then, the current token is assigned Tag 1. Otherwise, the current token is assigned Tag 0; σ denotes an activation function.

The process of the *o2p* direction is similar and will not be introduced here. At this stage, we obtain all possible sets of subjects $Y_s(S)$and sets of objects $Y_o(S)$, respectively.

3.2.2. Subject–Object Alignment

This section only introduces entity pair extraction in the *s2p* direction. The subtask of subject–object alignment is to identify all possible objects corresponding to all subjects extracted in the previous phase. In order to do this, we iterated the subject of detected subjects set $Y_s(S)$. We used the attention mechanism to mine the relation between the identified subject, object features, and relational features, translating and decoding all possible entities corresponding to each detected subject.

Specifically, we obtaind a fused representation of specific subject, relation features, and object features, then used the self-attention mechanism to obtain a selective representation. The detailed operations are as follows:

$$\begin{aligned} \boldsymbol{Q} &= \boldsymbol{W}_{query}(\boldsymbol{v}_{sub_k} \circ \boldsymbol{ho} + \boldsymbol{hr}) + b_{query} \\ \boldsymbol{K} &= \boldsymbol{W}_{key}(\boldsymbol{v}_{sub_k} \circ \boldsymbol{ho} + \boldsymbol{hr}) + b_{key} \\ \boldsymbol{V} &= \boldsymbol{W}_{value}(\boldsymbol{v}_{sub_k} \circ \boldsymbol{ho} + \boldsymbol{hr}) + b_{value} \\ \boldsymbol{A} &= \text{Softmax}(\frac{\boldsymbol{QK^T}}{\sqrt{d_k}})\boldsymbol{V} \end{aligned} \tag{4}$$

where d_k is the dimension of the attention network, and $\circ$ denotes a hadamard product operation. Note that the subject is frequently composed of more than one token. We take the maxpool vector representation of the k-th subject as v_{sub_k}.

Next, a binary classifier is used to identify the object, including its start and end position. The detailed operations are as follows:

$$\begin{aligned} p_i^{s2p_start} &= \sigma(\boldsymbol{W}_{s2p_start}\boldsymbol{A_i} + \boldsymbol{b}_{s2p_start}) \\ p_i^{s2p_end} &= \sigma(\boldsymbol{W}_{s2p_end}\boldsymbol{A_i} + \boldsymbol{b}_{s2p_end}) \end{aligned} \tag{5}$$

where $p_i^{s2p_start}$ and $p_i^{s2p_end}$ represent the possibility that the i-th token in the sentence is the start or end position of the object corresponding to the subject, respectively.

Finally, the final set of entity pairs $Y_{(s,o)}(S)$ is formed by fusing the two sets of entity pairs identified in both directions.

3.2.3. Relation Judgement

After subject–object alignment, we obtaind all the entity pairs with possible relations; as shown in Figure 2, we defined a (entity pair, relation) bipartite graph for relation judgement, which takes the projected entity pairs $\boldsymbol{E}_{head}$ representations and $\boldsymbol{E}_{tail}$.

$$\begin{aligned} \boldsymbol{E}_{head} &= \boldsymbol{W}_h\boldsymbol{E}_s + \boldsymbol{b}_h \\ \boldsymbol{E}_{tail} &= \boldsymbol{W}_t\boldsymbol{E}_o + \boldsymbol{b}_t \end{aligned} \tag{6}$$

where $\boldsymbol{E}_{head}$ and $\boldsymbol{E}_{tail}$ are the d-dimentional representations obtained by v_{sub_k} and v_{obj_k}; $\boldsymbol{W}_h$, $\boldsymbol{W}_t \in \mathbb{R}^{d\times d}$ are two matrices, allowing for the model to extract subject–object characteristics of entity pairs; and b_h, $b_t \in \mathbb{R}^{d\times 1}$ are two biases.

Lastly, we predicted the relations between each pair of entities for each relation r to determine if they can form a valid triple:

$$p^k = \sigma\left(\boldsymbol{E}_{\text{head}}^T\,\boldsymbol{U}_k\boldsymbol{E}_{\text{tail}}\right) \tag{7}$$

where $U_k \in \mathbb{R}^{d+1\times d+1}$ is a relation-specific matrix. If the possibility p_{ij}^k exceeds the predefined threshold, the relational triple (s_i, r_k, o_j) will be considered to exist; otherwise, it will be considered not to exist.

Linking a bipartite graph to judge relations can better encode the directional information of words during decoding and has an excellent classification ability. These two benefits help to increase the precision of the extraction. On the one hand, this keeps track of the matrix for every type of relation, which faithfully represents the traits of the relation; on the other hand, it accurately mines the relations between subjects and objects thanks to its probabilistic computational architecture.

3.3. Joint Training Strategy

Our proposed approach can be seen as three steps with five subtasks. We traind the model jointly, optimized the combined objective function during training, and shared the BTDM encoder parameters. According to our description, our model is a bidirectional extraction model, where all modules on the bidirectional work in multi-task learning, and the extraction of entities in the *s2p* direction and loss of entities on the extracted modules are calculated as follows.

$$\begin{aligned}\mathcal{L}_s &= -\frac{1}{L}\prod_{t\in\{\text{s_start,s_end}\}}\prod_{i=1}^{L} p_i(t)\\ \mathcal{L}_{s2p} &= -\frac{1}{L}\prod_{t\in\{\text{s2p_start,s2p_end}\}}\prod_{i=1}^{L} p_i(t)\end{aligned} \tag{8}$$

where $p_i(t) = ^{\mathbf{I}\{y_i^t=1\}}(1-p_i^t)^{\mathbf{I}\{y_i^t=0\}}$, L is the length of the sentence, $\mathbf{I}\{z\} = 1$ if z is true and 0 otherwise, and $p_i(t)$ is a label of the start or end position of each entity for the i-th token in sentence. Similarly, the same loss functions $\mathcal{L}_o$ and $\mathcal{L}_{o2p}$ were found in the *o2p* direction.

The subtask of relation judgement minimizes the following binary cross-entropy loss function:

$$\mathcal{L}_{rel} = -\frac{1}{K}\sum_{k=1}^{K}(y_t \log p^k + (1-y_t)\log(1-p^k)) \tag{9}$$

where K denotes the number of predefined types of relation, and y_t is the right label of the relational triple (s_i, r_k, o_j).

Finally, the overall loss function definition of BTDM is as follows. The weight of each sub-loss may be carefully adjusted to improve performance; however, for the sake of simplicity, we merely provided identical weights (i.e., $\alpha = \beta = \gamma = \delta = \epsilon = 1$).

$$\mathcal{L}_{total} = \alpha\mathcal{L}_s + \beta\mathcal{L}_o + \gamma\mathcal{L}_{s2o} + \delta\mathcal{L}_{o2s} + \epsilon\mathcal{L}_{rel} \tag{10}$$

In addition, to alleviate the issue of exposure bias, we used a negative sample strategy to randomly generate some negative samples to add to all the training sets. During the inference time, the total sample containing negative samples can simulate the true scenario. This method improves the robustness and recognition accuracy of the model in natural scenes. As a result, the exposure bias problem is greatly alleviated.

4. Experiments

We conducted numerous experiments to evaluate and characterize the proposed BTDM. This section first describes the experimental conditions. Following that, the evaluation findings and subsequent discussion are given.

4.1. Datasets and Evaluation Metrics

For a fair and comprehensive comparison, we followed [24,25] to evaluate our model on two publicly used datasets: *The New York Times* (denoted as NYT) [26] and WebNLG [27]. The NYT dataset was developed using distant supervision, which au-

tomatically aligns the relational facts in Freebase. The NYT dataset consists of 56 k training data and 5 k test data. The WebNLG dataset was initially created for the natural language generation (NLG) task, which tries to generate corresponding descriptions using the given triples. The WebNLG dataset contains 5 k training data and 703 test data. There were two versions of each dataset. The different versions were denoted as NYT*, NYT and WebNLG*, WebNLG, respectively. It is worth noting that NYT* and WebNLG* annotate the last word of entities, whereas NYT and WebNLG annotate the entire entity span. Table 1 describes the statistics of the datasets. Following [17], we further characterized the test set with respect to the overlapping patterns of relational triples and the number of relational triples in each sentence.

Table 1. Statistics of four datasets used in our experiments, where N is the number of relational triples in a sentence.We divided the overlapping patterns into four categories: *Normal*, *Single Entity Overlap (SEO)*, *Entity Pair Overlap (EPO)*, and *Subject Object Overlap (SOO)*.

Dataset	#Sentence			Details of Test Data							
	Train	Valid	Test	Normal	SEO	EPO	SOO	$N = 1$	$N > 1$	#Triples	#Relations
NYT*	56,195	4999	5000	3266	1297	978	45	3244	1756	8110	24
WebNLG*	5019	500	703	245	457	26	84	266	437	1591	171
NYT	56,196	5000	5000	3071	1273	1168	117	3089	1911	8616	24
WebNLG	5019	500	703	239	448	6	85	256	447	1607	216

In keeping with previous studies, a relational triple is only regarded as valid if it precisely fits the ground truth. Specifically, if the subject s, object o, and relation r are identical to the ground truth, the predicted triple (s, r, o) is considered valid. Meanwhile, the standard micro-precision (denoted as Prec.), recall (denoted as Rec.), and F1-score (denoted as F1) for each baseline are presented.

4.2. Implementation Details

We implemented our model using the frame of PyTorch. During training, we adopted the Adam [28] optimizer. The hyperparameters were tuned by the grid search on the validation set. The learning rate was set to $1 \times 10^{-3}/5 \times 10^{-5}$. The batch size was set to eight on NYT* and NYT and six on WebNLG* and WebNLG. We trained the model for 200 epochs and chose the best model. We used the pre-trained Cased BERT base (https://huggingface.co/bert-base-cased) as the sentence encoder and set the max length of a sentence to 100. Our experiments were conducted on the workstation with an Intel Xeon(R) E5 2.20 GHz CPU, 256 GB memory, an NVIDIA Tesla V100GPU, and Ubuntu 20.04.

4.3. Baselines

We compared our model with several baselines. The compared models were as follows:

- NovelTagging [29] treats the extraction problem as a sequence labeling problem by merging entities and relational roles;
- CopyR [30] applies a sequence-to-sentence architecture;
- MultiHead [31] uses a multi-headed selection technique to identify the entity and the relation;
- GraphRel [32] uses graph convolutional neural networks to extract entities and their relations;
- OrderCopyR [33] uses reinforcement learning in seq-to-seq models to extract triples;
- ETL-span [24] proposed a decomposition-based tagging scheme;
- RSAN [34] proposed a relation-specific attention mechanism network to extract entities and relations extraction;
- CasRel [17] applies a cascade binary tagging framework to extract triples;
- TPLinker [25] firstly finds all token pairs and then uses map to tag token links to recognize relations between token pairs;
- TDEER [18] proposes a translating decoding network to relational triples;

- PRGC [19] proposes a framework based on a potential relation and global correspondence;
- R-BPtrNet [35] proposes a unified network to extract explicit and implicit relational triples.

For a fair comparison, all baseline results are taken directly from the original literature.

4.4. Experimental Results

In this section, we provide the overall outcomes as well as the results of the other difficult scenarios.

4.4.1. Main Results

The comparison results of our model BTDM versus thirteen baselines on the NYT and WebNLG datasets are shown in Table 2.

Table 2. Comparison (%) of the proposed BTDM method with the baselines. Note **Bold** marks the highest score and underline marks the second best score.

Model	NYT*			WebNLG*			NYT			WebNLG		
	Prec.	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
NovelTagging [29]	-	-	-	-	-	-	32.8	30.6	31.7	52.5	19.3	28.3
CopyRE [30]	61.0	56.6	58.7	37.7	36.4	37.1	-	-	-	-	-	-
MultiHead [31]	-	-	-	-	-	-	60.7	58.6	59.6	57.5	54.1	55.7
GraphRel [32]	63.9	60.0	61.9	44.7	41.1	42.9	-	-	-	-	-	-
OrderCopyRE [33]	77.9	67.2	72.1	63.3	59.9	61.6	-	-	-	-	-	-
ETL-span [24]	84.9	72.3	78.1	84.0	91.5	87.6	85.5	71.7	78.0	84.3	82.0	83.1
RSAN [34]	-	-	-	-	-	-	85.7	83.6	84.6	80.5	83.8	82.1
CasRel [17]	89.7	89.5	89.6	93.4	90.1	91.8	-	-	-	-	-	-
TPLinker [25]	91.3	**92.5**	91.9	91.8	92.0	91.9	91.4	**92.6**	92.0	88.9	84.5	86.7
TDEER [18]	93.0	92.1	92.5	93.8	92.4	93.1	-	-	-	-	-	-
PRGC [19]	**93.3**	91.9	92.6	94.0	92.1	93.0	**93.5**	91.9	**92.7**	89.9	87.2	88.5
R-BPtrNet [35]	92.7	**92.5**	92.6	93.7	92.8	93.3	-	-	-	-	-	-
BTDM	93.0	**92.5**	**92.7**	**94.1**	**93.5**	**93.8**	93.1	92.4	**92.7**	**90.9**	**90.1**	**90.5**

As shown, our technique BTDM outperforms all thirteen baselines overall. Our model outperforms PRGC [19] on 9 of the 12 criteria and shows the second best performance, if not the best, on the other 3. In particular, BTDM outperforms baseline models over the F1 scores on all datasets. BTDM outperforms the strongest baseline with an 0.5% and 2% absolute gain in F1-score on datasets WebNLG and WebNLG*. Additionally, as shown in Table 1, WebNLG and WebNLG* are more challenging to improve than the NYT dataset because they contain fewer data but a greater variety of relation types (216 and 171, respectively). Nevertheless, not only did we achieve an improvement in both datasets, surpassing the previous best model in every metric, but for the first time ever, our model BTDM boosted the WebNLG dataset's 3 metrics by over 90%.

In addition, we noticed two phenomena. CasRel is a typical one-way extraction framework. Our model BDTM shows a superior performance compared to CasRel, demonstrating the correctness of our idea that bidirectional extraction avoids the propagation of errors due to subject identification failure. Additionally, the PRGC is the latest SOTA model, which first predicts the possible relations in the text. This leads to the omission of some possible relations, which leads to poor generalization of the model. Our model, the BDTM, performs better than the PRGC, especially in datasets with more relation types. This proves our idea that adding relation features can strengthen the interaction between entity pairs and relations so that the model can find the relationship between entity pairs more accurately and effectively. In addition, our bipartite graph-linking approach maintains a relationship matrix, allowing for each relation to make relation judgments and providing powerful classification and relation-mining capabilities.

4.4.2. Detailed Results on Complex Scenarios

To validate the performance of BTDM in some complex scenarios, we used a subset of the NYT* and WebNLG* datasets for complementary experiments, which included

testing the capacity to handle different types of overlapping triples and sentences with varying amounts of triples. Following previous works [17,19,25,34], We divided the overlapping patterns into four categories: *Normal, Single Entity Overlap (SEO), Entity Pair Overlap (EPO)*, and *Subject Object Overlap (SOO)*. The dataset can be divided into five subdatasets: $N = 1, N = 2, N = 3, N = 4, N \geq 5$, where N is the number of relational triples in a sentence. We chose five strong models as our baselines, and Table 3 provides the total outcomes.

Table 3. F1-score (%) of relational triple-extraction with different overlapping patterns. Note that † marks results reported by PRGC, and **Bold** marks the highest score in the table.

Model	NYT*									WebNLG*								
	Normal	**EPO**	**SEO**	**SOO**	***N* = 1**	***N* = 2**	***N* = 3**	***N* = 4**	***N* ≥ 5**	**Normal**	**EPO**	**SEO**	**SOO**	***N* = 1**	***N* = 2**	***N* = 3**	***N* = 4**	***N* ≥ 5**
OrderCopyRE	71.2	69.4	72.8	-	71.7	72.6	72.5	77.9	45.9	65.4	60.1	67.4	-	63.4	62.2	64.4	57.2	55.7
ETL-Span	88.5	87.6	60.3	-	88.5	82.1	74.7	75.6	76.9	87.3	91.5	80.5	-	82.1	86.5	91.4	89.5	91.1
CasRel	87.3	91.4	92.0	77.0 †	88.2	90.3	91.9	94.2	83.7	89.4	92.2	94.7	90.4 †	89.3	90.8	94.2	92.4	90.9
TPLinker	90.1	93.4	94.0	**90.1** †	90.0	92.8	93.1	96.1	90.0	87.9	92.5	95.3	86.0 †	88.0	90.1	94.6	93.3	91.6
PRGC	**91.0**	94.0	94.5	81.8	**91.1**	93.0	93.5	95.5	93.0	90.4	93.6	**95.9**	**94.6**	89.9	91.6	95.0	94.8	**92.8**
BTDM	90.8	**94.7**	**94.9**	87.5	90.7	**93.4**	**94.2**	**96.2**	**94.0**	**91.0**	**94.3**	93.5	**94.6**	**90.8**	**92.5**	**96.1**	**95.4**	92.7

As shown in the Table 3, BTDM has the highest F1-score on thirteen of the eighteen subsets and ranks second on the remaining four subsets. In addition, when there are multiple triples in a sentence, the BTDM obtains more performance gains.

The excellent performance of BTDM could be due to two factors: on the one hand, it effectively alleviates the error propagation problem, ensuring the recall and precision of the extraction of entity pairs and further ensures the precision of the extracted relational triples; on the other hand, the model maintains a relation matrix for each relation and applies a specific relation link between each entity pair, which guarantees the recall of the extracted relational triples. This ultimately means that the model has an excellent composite F1 metric in various complex scenarios.

In general, the aforementioned experimental results show that BTDM outperforms the baseline and is more robust in dealing with a variety of complicated scenarios.

4.4.3. Detailed Results on Different Sub-Tasks

The bidirectional translating decoding structure of our model can effectively mitigate error propagation and align entity pairs, and the linking of bipartite graphs can effectively identify the relations between entity pairs. To further investigate the performance of the model on the alignment of entity pair and identification of relations, we chose the current best SOTA model PRGC (one of the current state-of-the-art triple-extraction models, with an outstanding performance in relation judgement and subject–object alignment) for further comparison experiments on subtasks. The results are shown in Table 4.

In general, BTDM outperforms PRGC in thirteen of the eighteen metrics and achieves a similar performance to PRGC in the remaining metrics. Moreover, BTDM exceeds PRGC in F1 metrics on all subtasks, indicating the overall superiority of model and the stability of performance of model on each subtask. In addition, on the relation extraction subtask, we perform about the same, or even slightly better, than the PRGC method on dataset NYT*. However, on dataset WebNLG with more types of relations, the recall rate is similar to that of PRGC: we achieved a 3.4% performance improvement in terms of precision, indicating the superior performance of the relation embedding and bipartite graph-linking approaches in relation extraction.

Lastly, on the entity pair and relation triplet tasks, BTDM leads PRGC overall. The experiments also demonstrate that capturing the dependencies between entity pairs and relations is crucial, which points us in the direction of designing stronger models in future work.

Table 4. Results on different subtasks. (s, o) denotes the subtask of the entity pair, r means the subtask of relation, and (s, r, o) means the task of relational triples.

Model	Element	NYT*			WebNLG*		
		Prec.	**Rec.**	**F1**	**Prec.**	**Rec.**	**F1**
PRGC	(s, o)	**94.0**	92.3	**93.1**	**96.0**	93.4	94.7
	r	95.3	**96.3**	95.8	92.8	**96.2**	94.5
	(s, r, o)	**93.3**	91.9	92.6	94.0	92.1	93.0
BTDM	(s, o)	93.2	**93.0**	**93.1**	95.8	**95.1**	**95.5**
	r	**96.9**	95.4	**96.1**	**96.2**	95.1	**95.7**
	(s, r, o)	93.0	**92.5**	**92.7**	**94.1**	**93.5**	**93.8**

5. Related Work

The work associated with joint entity and relationship extraction can be divided into three categories.

The first category is tagging-based methods. This type of approach [17–19,24,29,34,36–38] can be used for relational triple-extraction tasks when applied to multiple interrelated sequential labeling problems. It first marks the start and the end positions of the entity and then identifies the relation between them. For instance, NovelTagging [29] is a typical approach, designing a complex labeling scheme that contains information about the start position of the entity, the end position of the entity, and relations. Several studies employ a sequence-tagging network to identify the positions of the entity, then classify the relationship using multiple classification networks. The following endeavors have recently gained much attention and achieved a competitive performance. CasRel [17] treats the relational triple-extraction task as a subject-to-object mapping relation, first identifying all possible entity heads and then applying a relation-based sequence tagging network to each entity head to identify the corresponding entity tails. PRGC [19] proposes an extraction method based on relation prediction and global correspondence, which constrains the relation classification to a subset of predicted relations, greatly reducing the complexity of determining all relations.

The second category is table-filling based methods. This type of approach [25,32,39–41] forms a table of input sentences and identifies the relations between entities by filling the table. Some works keep a table for each relation, with items representing the start and end positions of the two entities with this relation. Graphrel [32] uses words as nodes and identifies the relations between word pairs by discriminating between nodes. TPLinker [25] converts the relational triple-extraction task into a linking problem between entity pairs and introduces a handshake-marking scheme to align the boundaries of the entity pairs.

The third category is seq2seq-based methods. This type of approach [2,30,33,35,42] treats a triple as a sequence of tokens. It transforms the triple-extraction task into a generation task, which uses an encoder–decoder framework to generate the elements of the triple. For instance, CopyRE [30] uses a copy method to extract the hidden relations behind entities and can resolve the triple overlapping issue. R-BPtrNet [35] proposes a unified framework with a binary pointer network, which can extract explicit relational triples and implicit relational triples through a binary pointer network.

6. Conclusions and Future Work

In this study, we revisited the relational triple-extraction task from a fresh bidirectional perspective and proposed a new joint relational triple-extraction framework bidirectional translating decoding model. The bidirectional framework of BTDM greatly mitigates error propagation; the translating decoding approach of BTDM handles the overlapping triple problem and efficiently aligns subject–object. The relation judgement of the bipartite graph of BTDM fully mines the relations between entity pairs to achieve accurate and efficient relation judgement. The experimental results show that BTDM exhibited SOTA performance on several public datasets and performed equally well in different complex scenarios.

For future work, we hope to investigate more effective approaches to the error propagation problem. We would like to investigate the use of triple classification in additional types of information-extraction challenges, such as document-level relation extraction and event extraction.

Author Contributions: Conceptualization, Z.Z. and J.Y.; methodology, Z.Z.; software, Z.Z.; validation, Z.Z., J.Y. and H.L.; formal analysis, H.L. and P.H.; investigation, Z.Z.; resources, J.Y. and H.L.; data curation, Z.Z.; writing—original draft preparation, Z.Z.; writing—review and editing, Z.Z., H.L. and P.H.; visualization, Z.Z. and P.H.; supervision, J.Y.; project administration, Z.Z. and J.Y.; funding acquisition, J.Y. and H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Anhui Provincial Science Foundation (NO.1908085MF202) and Independent Scientific Research Program of National University of Defense Science and Technology (NO.ZK18-03-14).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

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

Conflicts of Interest: The author declare no conflict of interest.

References

1. Dong, X.; Gabrilovich, E.; Heitz, G.; Horn, W.; Lao, N.; Murphy, K.; Strohmann, T.; Sun, S.; Zhang, W. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 601–610.
2. Nayak, T.; Majumder, N.; Goyal, P.; Poria, S. Deep neural approaches to relation triplets extraction: A comprehensive survey. *Cogn. Comput.* **2021**, *13*, 1215–1232. [CrossRef]
3. Fader, A.; Zettlemoyer, L.; Etzioni, O. Open question answering over curated and extracted knowledge bases. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 24–27 August 2014; pp. 1156–1165.
4. Li, X.; Yin, F.; Sun, Z.; Li, X.; Yuan, A.; Chai, D.; Zhou, M.; Li, J. Entity-Relation Extraction as Multi-Turn Question Answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Florence, Italy, 2019; pp. 1340–1350. [CrossRef]
5. Huang, C.C.; Lu, Z. Community challenges in biomedical text mining over 10 years: Success, failure and the future. *Briefings Bioinform.* **2016**, *17*, 132–144. [CrossRef] [PubMed]
6. Lai, T.; Ji, H.; Zhai, C.; Tran, Q.H. Joint Biomedical Entity and Relation Extraction with Knowledge-Enhanced Collective Inference. 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; Association for Computational Linguistics: Seattle, WA, USA, 2021; pp. 6248–6260. [CrossRef]
7. Zelenko, D.; Aone, C.; Richardella, A. Kernel methods for relation extraction. *J. Mach. Learn. Res.* **2003**, *3*, 1083–1106.
8. Zhou, G.; Su, J.; Zhang, J.; Zhang, M. Exploring Various Knowledge in Relation Extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*; Association for Computational Linguistics: Ann Arbor, MI, USA, 2005; pp. 427–434. [CrossRef]
9. 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*; Association for Computational Linguistics: Portland, OR, USA, 2011; pp. 551–560.
10. Zhong, Z.; Chen, D. A Frustratingly Easy Approach for Entity and Relation Extraction. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*; Association for Computational Linguistics: Seattle, WA, USA, 2021; pp. 50–61. [CrossRef]
11. Tjong, E.F.; De Meulder, F. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL*; Association for Computational Linguistics: Seattle, WA, USA, 2003; pp. 142–147.
12. Ratinov, L.; Roth, D. Design Challenges and Misconceptions in Named Entity Recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*; Association for Computational Linguistics: Boulder, CO, USA, 2009; pp. 147–155.
13. Bunescu, R.C.; Mooney, R.J. A Shortest Path Dependency Kernel for Relation Extraction. In *Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP-05)*; Association for Computational Linguistics: Vancouver, BC, Canada, 2005; pp. 724–731.

14. 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*; (Volume 1: Long Papers); Association for Computational Linguistics: Baltimore, MD, USA, 2014; pp. 402–412. [CrossRef]
15. 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, Geneva, Switzerland, 3–7 April 2017; pp. 1015–1024.
16. Xu, B.; Wang, Q.; Lyu, Y.; Shi, Y.; Zhu, Y.; Gao, J.; Mao, Z. EmRel: Joint Representation of Entities and Embedded Relations for Multi-triple Extraction. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*; Association for Computational Linguistics: Seattle, WA, USA, 2022; pp. 659–665. [CrossRef]
17. Wei, Z.; Su, J.; Wang, Y.; Tian, Y.; Chang, Y. 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: Seattle, WA, USA, 2020; pp. 1476–1488. [CrossRef]
18. Li, X.; Luo, X.; Dong, C.; Yang, D.; Luan, B.; He, Z. TDEER: An Efficient Translating Decoding Schema for Joint Extraction of Entities and Relations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*; Association for Computational Linguistics: Punta Cana, Dominican Republic, 2021; pp. 8055–8064. [CrossRef]
19. Zheng, H.; Wen, R.; Chen, X.; Yang, Y.; Zhang, Y.; Zhang, Z.; Zhang, N.; Qin, B.; Ming, X.; Zheng, Y. PRGC: Potential Relation and Global Correspondence Based Joint Relational Triple Extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*; Volume 1: Long Papers; Association for Computational Linguistics: Seattle, WA, USA, 2021; pp. 6225–6235. [CrossRef]
20. 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*; Volume 1: Long and Short Papers; Association for Computational Linguistics: Minneapolis, MN, USA, 2019; pp. 4171–4186. [CrossRef]
21. Pennington, J.; Socher, R.; Manning, C. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*; Association for Computational Linguistics: Doha, Qatar, 2014; pp. 1532–1543. [CrossRef]
22. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
23. Sui, D.; Chen, Y.; Liu, K.; Zhao, J.; Zeng, X.; Liu, S. Joint entity and relation extraction with set prediction networks. *arXiv* **2020**, arXiv:2011.01675.
24. Yu, B.; Zhang, Z.; Shu, X.; Wang, Y.; Liu, T.; Wang, B.; Li, S. Joint extraction of entities and relations based on a novel decomposition strategy. *arXiv* **2019**, arXiv:1909.04273.
25. 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. *arXiv* **2020**, arXiv:2010.13415.
26. Riedel, S.; Yao, L.; McCallum, A. Modeling relations and their mentions without labeled text. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, 20–24 September 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 148–163.
27. 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)*; Association for Computational Linguistics: Vancouver, BC, Canada, 2017.
28. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
29. Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; 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*; Volume 1: Long Papers; Association for Computational Linguistics: Vancouver, BC, Canada, 2017; pp. 1227–1236. [CrossRef]
30. Zeng, X.; Zeng, D.; He, S.; 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*; Volume 1: Long Papers; Association for Computational Linguistics: Melbourne, Australia, 2018; pp. 506–514. [CrossRef]
31. Bekoulis, G.; Deleu, J.; Demeester, T.; Develder, C. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Syst. Appl.* **2018**, *114*, 34–45. [CrossRef]
32. 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, Florence, Italy, 28 July–2 August 2019; pp. 1409–1418.
33. Zeng, X.; He, S.; Zeng, D.; Liu, K.; Liu, S.; 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 the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*; Association for Computational Linguistics: Hong Kong, China, 2019; pp. 367–377. [CrossRef]
34. Yuan, Y.; Zhou, X.; Pan, S.; Zhu, Q.; Song, Z.; Guo, L. A Relation-Specific Attention Network for Joint Entity and Relation Extraction. In Proceedings of the IJCAI, Yokohama, Japan, 11–17 July 2020; Volume 2020, pp. 4054–4060.

35. Chen, Y.; Zhang, Y.; Hu, C.; Huang, Y. 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*; Association for Computational Linguistics: Seattle, WA, USA, 2021; pp. 5694–5703. [CrossRef]
36. Sun, C.; Gong, Y.; Wu, Y.; Gong, M.; Jiang, D.; Lan, M.; Sun, S.; Duan, N. Joint Type Inference on Entities and Relations via Graph Convolutional Networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*; Association for Computational Linguistics: Florence, Italy, 2019; pp. 1361–1370. [CrossRef]
37. Liu, J.; Chen, S.; Wang, B.; Zhang, J.; Li, N.; Xu, T. Attention as relation: Learning supervised multi-head self-attention for relation extraction. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, Yokohama, Japan, 7–15 January 2021; pp. 3787–3793.
38. Tian, X.; Jing, L.; He, L.; Liu, F. 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; Association for Computational Linguistics: Seattle, WA, USA, 2021; pp. 4851–4861. [CrossRef]
39. 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*; Volume 1: Long Papers; Association for Computational Linguistics: Berlin, Germany, 2016; pp. 1105–1116. [CrossRef]
40. 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; Association for Computational Linguistics: Osaka, Japan, 2016; pp. 2537–2547.
41. 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*; Association for Computational Linguistics: Copenhagen, Denmark, 2017; pp. 1730–1740. [CrossRef]
42. Zeng, D.; Zhang, H.; Liu, Q. Copymtl: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 9507–9514.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.