applied sciences

MDPI

Article

Joint Extraction of Entities and Relations via Entity and Relation Heterogeneous Graph Attention Networks

Bo Jiang [1,2] and Jia Cao [1,2,*]

[1] School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China
[2] Engineering Research Center for Forestry-Oriented Intelligent Information Processing, National Forestry and Grassland Administration, Beijing 100083, China
* Correspondence: caojia@bjfu.edu.cn

Abstract: Entity and relation extraction (ERE) is a core task in information extraction. This task has always faced the overlap problem. It was found that heterogeneous graph attention networks could enhance semantic analysis and fusion between entities and relations to improve the ERE performance in our previous work. In this paper, an entity and relation heterogeneous graph attention network (ERHGA) is proposed for joint ERE. A heterogeneous graph attention network with a gate mechanism was constructed containing word nodes, subject nodes, and relation nodes to learn and enhance the embedding of parts for relational triple extraction. The ERHGA was evaluated on the public relation extraction dataset named WebNLG. The experimental results demonstrate that the ERHGA, by taking subjects and relations as a priori information, can effectively handle the relational triple extraction problem and outperform all baselines to 93.3%, especially overlapping relational triples.

Keywords: entity and relation extraction; overlapping triples; heterogeneous graph attention networks

check for updates

Citation: Jiang, B.; Cao, J. Joint Extraction of Entities and Relations via Entity and Relation Heterogeneous Graph Attention Networks. *Appl. Sci.* **2023**, *13*, 842. https://doi.org/10.3390/app13020842

Academic Editor: Yutaka Ishibashi

Received: 17 November 2022
Revised: 2 January 2023
Accepted: 4 January 2023
Published: 7 January 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

Information extraction (IE) aims to structure information contained in text. Entity and relation extraction (ERE) is a core task in information extraction. Entity pairs and the relations between them can be represented by relational triples (i.e., subject, relation, and object), where the subject is the head entity and the object is the tail entity. Thus, an unstructured text is structuralized by several triples, which are basic components of the knowledge graphs.

The field of ERE has always faced the overlap problem. Sentences are divided into two overlapping types: single entity overlap (SEO) and entity pair overlap (EPO) [1]. The SEO means that one entity is overlapping and contained by multiple relational triples in a sentence, while the EPO means that an entity pair is overlapping. When a sentence has no overlap, it belongs to the normal category.

In recent years, many studies have been devoted to this challenge; for example, CopyRE [1], CasRel [2], PRGC [3], etc. More studies and their detailed descriptions are given in Section 2.1.

The aforementioned methods have made great contributions to joint ERE. However, few existing works consider the semantic fusion between subjects and relations when extracting objects. Incorporating subjects and relations as a priori information would detect more semantically relevant objects in the sentence and alleviate the problem of overlapping triples. It is worth noting that subjects and relations are two distinct types of information for objects, so it is necessary to distinguish the importance of different types and different information.

The heterogeneous graph attention network (HGAT) has achieved great success in short text classification [4,5], knowledge graph embedding [6], social event detection [7], etc. These achievements are consistent with the target of ERE, so a model based on an entity

and relation heterogeneous graph attention network was proposed to fuse their semantic information to cope with overlapping relational triples.

In this paper, a novel entity and relation heterogeneous graph attention network is imported to achieve better performance on SEO and EPO. A heterogeneous graph attention network is constructed including three types of nodes—word nodes, relation nodes, and subject nodes—to learn and enhance the embedded representation of parts for relation extraction, combined with a gating mechanism to control the updating of the nodes. The experimental results demonstrate that the ERHGA can effectively solve the relational triple extraction problem and outperform all baselines [1–3,8–15].

The rest of this article is structured as follows: Section 2 summarizes prior research on joint ERE and HGAT. In Section 3, the problem formulation is proposed. In Section 4, we describe the methodology. The experimental details, evaluation results, and discussion are presented in Section 5. Finally, our concluding viewpoints are presented in Section 6.

2. Related Works

2.1. ERE-Related Work

As the core task of information extraction, entity and relation extraction automatically extracts entities and identifies the relations between entity pairs by modeling textual information, and its research results are primarily applied in automatic question answering, machine translation, knowledge graphs, etc.

Early pipeline-based methods [16–20] for ERE can be divided into two subtasks, called entity recognition (NER) and relation extraction (RE), in chronological order. This approach causes error propagation and ignores the correlation that exists between the two steps. Due to the inherent flaws of the pipeline approaches, researchers have shifted their focus to joint entity and relation extraction.

Joint extraction methods integrate ERE and RE, simultaneously extracting entities and classifying the relations of entity pairs. Many joint methods apply parameter sharing or table filling to achieve joint extraction [21–24], and one of them was proposed by Katiyar and Cardie as the first true neural-network-based joint extraction model [22]. A novel tagging strategy called NovelTagging for assigning tags to words was created by Zheng et al. [8], which can transform the joint ERE task into a sequence labeling problem. All of the foregoing works overlooked the fact that an entity may contain multiple relational triples. To assign multiple relations to an entity, the task is considered as a multiple head selection problem [25]. The subject or object of one relational triple can also be the subject or object of another relational triple (SEO), or there may be more than one relation between the entity pairs (EPO) [1]. NovelTagging [8] cannot resolve overlapping relational triples, as a word can only be assigned one tag. A sequence-to-sequence model with a copy mechanism named CopyRE was proposed by Zeng et al. [1], in which one entity can be copied several times when it exists in different triples. This method was very novel at that time because it used a fresh perspective to revisit the characteristics of datasets. To consider the interactions between entities and relations, and to overcome the limited number of relational triples generated by CopyRE, a relation extraction model called GraphRel using graph convolutional networks was presented by Fu et al. [9]. Zeng et al. [10] found that CopyRE suffers from inaccurate entity copying and the generated entities are not complete, so a multitask learning framework called CopyMTL was introduced to solve these problems. To make up for GraphRel ignoring the relational features, a framework called Relation-Aware improving the relation-aware attention mechanism and building the complete interaction between sentences and relations was designed by Hong et al. [11]. A cascade binary tagging framework called CasRel was proposed by Wei et al. [2], modeling relations as functions that map subjects to objects. It handles the overlapping triple problem by design and provides a general algorithmic framework. TPLinker, a handshaking tagging scheme designed by Wang et al. [12], eliminates the influence of complicated entities in extracting overlapping entity relations by answering questions. In light of the relations implicit in the relation trigger words of the sentence, RIFRE, proposed by Zhao et al. [13],

takes relations as a priori knowledge in the framework. To reduce computational complexity and avoid sparse labels, Zheng et al. [3], based on PRGC (potential relation and global correspondence), decomposed the relational triple extraction into three subtasks: relation judgement, entity extraction, and subject–object alignment. A dual-decoder model was reported by Ma et al. [26], first identifying relations using the text semantics and then considering these relations as additional mapping features to guide entity pair extraction to address the overlapping triple problem. To make full use of syntactical information and consider the interaction between relations, an attentive and transformer-based weighted GCN was reported by El-allaly et al. [27]. LDPs combined with active learning were used to solve the dilemma that relation extraction requires a lot of training data [28]. RMAN [14] applied two multi-head attention layers to add extra semantic information. A Gaussian graph generator and an attention-guiding layer were utilized to address the lack of information, and the GlobalPointer decoder was used to tackle the extraction of problematic overlapping triples by Liang et al. [15].

2.2. Heterogeneous Graph Attention Network (HGAT)-Related Work

Recently, the heterogeneous graph attention network (HGAT) has achieved appealing performance in a variety of tasks, especially natural language processing. A heterogeneous graph attention network was applied to short text classification by Hu et al. [4] and Yang et al. [5]. A flexible heterogeneous information network (HIN) framework was used for modeling short texts to address the semantic sparsity of short texts, which can fuse additional information, and their heterogeneous graph attention network includes a dual-level attention mechanism at multiple granularity levels to capture key information. Because a user's profile can be inferred from other dependent users, a heterogeneous graph attention network was launched by Chen et al. [29] to solve a single type of input data to address the task of user profiling. A heterogeneous graph attention network framework was proposed by Huang et al. [30] to capture the global semantic relations and the global structural information for rumor detection. Knowledge graphs have an intrinsic property of heterogeneity, so a heterogeneous relation attention network was proposed by Li et al. [6] to learn knowledge graph embedding. A heterogeneous graph neural network containing sentence nodes and entity nodes was suggested by Xie et al. [7] for few-shot relation classification. To acquire more knowledge, a model based on heterogeneous graph attention networks was built by Cao et al. [31] to incorporate the rich semantics and structural information in social messages for incremental social event detection. Heterogeneous global graph neural networks were utilized by Pang et al. [32] to consider item transition patterns from other users' historical sessions for deducing user preferences from the current and historical sessions. Lu et al. [33] applied heterogeneous graph attention networks for aspect sentiment analysis. Liang et al. [34] designed a heterogeneous graph-based model for emotional conversation generation.

3. Problem Formulation

Given a sentence h with annotated relational triples and a predefined set of relation types $R(R = \{r_1, r_2, \cdots, r_i \cdots, r_n\})$, n is the number of relations in the predefined relation dataset. The task is to extract all relational triples in the sentence accurately to improve the extraction of overlapping relational triples. Based on previous works [2,13], the same training goal was set.

4. Methodology

In this section, the entity and relation heterogeneous graph attention network (ERHGA) is described. The framework is shown in Figure 1, which consists of four main modules: encoder, subject extraction, update node representations with heterogeneous graph attention network, and relation-specific object extraction.

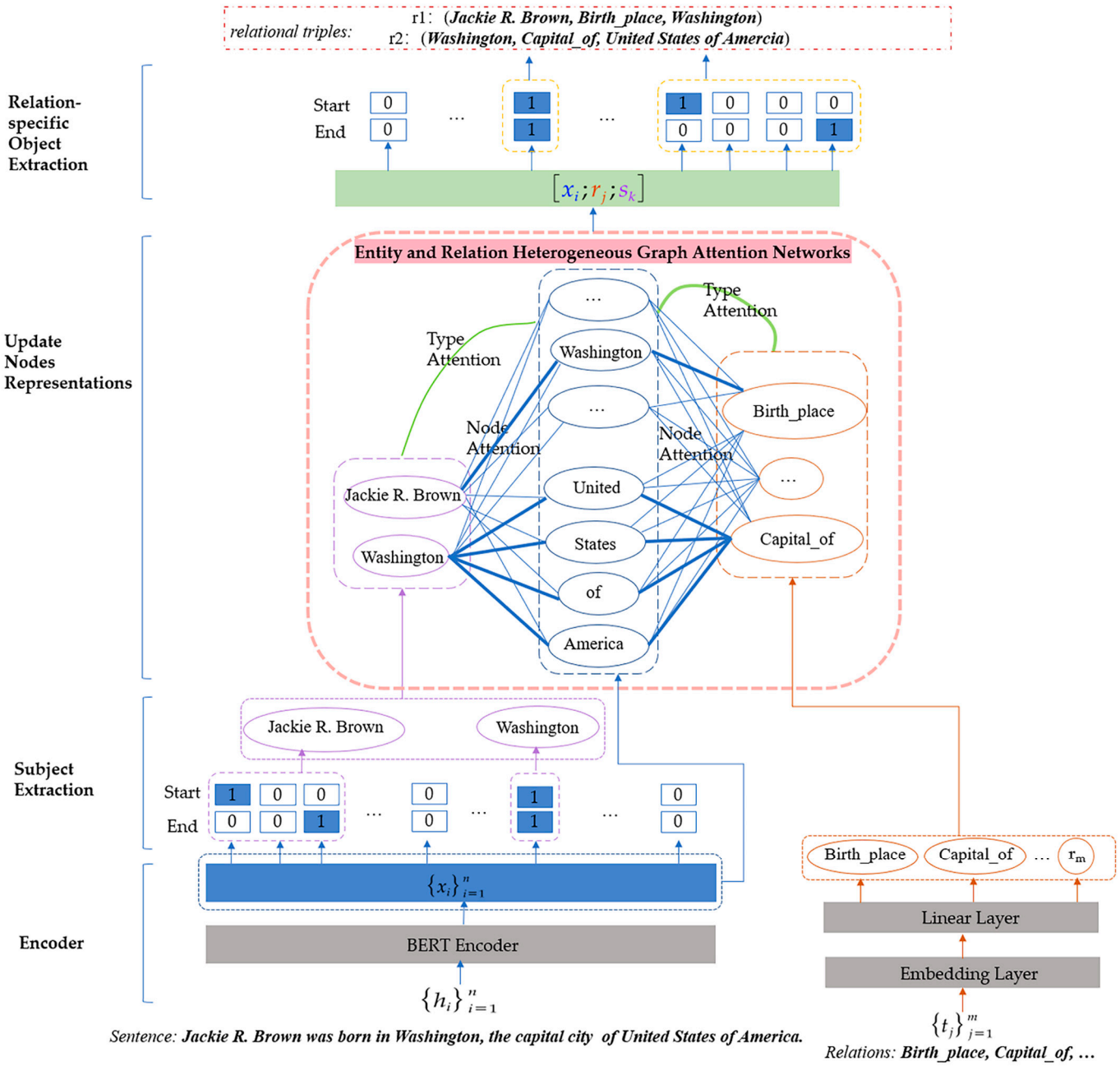


Figure 1. Architecture of our proposed ERHGA for joint entity and relation extraction tasks.

4.1. Encoder

The BERT [35] model was employed to encode sentence contextual information. Given a sentence $h(h = \{h_1, h_2, \cdots, h_i, \cdots, h_n\})$, x_i is the vector corresponding to h_i after BERT encoding.

$$[x_1, x_2, \cdots, x_i, \cdots, x_n] = BERT[h_1, h_2, \cdots, h_i, \cdots, h_n] \tag{1}$$

Each predefined relation label is mapped to a word embedded by an embedding layer and a linear layer.

$$[r_1, r_2, \cdots, r_j, \cdots, r_m] = W_r E[t_1, t_2, \cdots, t_j, \cdots, t_m] + b_r \tag{2}$$

where t_j is the one-hot embedding of a relation label, r_j denotes the vector corresponding to t_j, E is the relation embedding matrix in the shape of (m, d), d is the dimension of

word-embedding layer, and W_r and b_r are trainable parameters. The relation embedding has the same dimensions as the $\{x_i\}_{i=1}^{n}$ after a linear layer.

4.2. Subject Extraction

This module aims to detect all possible subjects $\{s_k\}_{k=1}^{q}$ in sentences. In this module, two identical binary classifiers are used to calculate the possibility of $\{x_i\}_{i=1}^{n}$ being the start position or end position of the subject [2]. As mentioned earlier, there are overlapping triples in the sentence. "Washington" is both the subject of r_2 and the object of r_1, all extracted in this module are regarded as subjects.

The span of the current subject is determined by the principle that the end position necessarily coincides with the start position or is the first end position after the start position. To define the span of subject s_k, a similar likelihood function is optimized [2].

4.3. Update Node Representations with Heterogeneous Graph Attention Network

The inputs of this module are $\{x_i\}_{i=1}^{n}$, $\{r_j\}_{j=1}^{m}$, and $\{s_k\}_{k=1}^{q}$, which learn and update representations of the three types of information through the heterogeneous graph attention network and use them for object extraction.

The module models the sentence's word tokens, relations, and subject tokens as nodes and then updates the nodes' representations. A node of any one type is considered as a neighbor of each node of the other two types, and the embedding of each node of each type is updated by a mechanism similar to a heterogeneous graph attention network [4,5].

In order to distinguish the importance of the types of the current node's neighbors to the current node, it is necessary to calculate the current node's attention scores to the other two types. First, the embedding of each type is represented as the average of the features of the nodes of that type, and then the attention score of the current node to the type is calculated based on the embedding of the current node and the embedding of the type.

$$h_{type} = \frac{1}{count \in \{n, m, q\}} \sum_{1}^{count} node \in \{x_i, r_j, s_k\} \tag{3}$$

where h_{type} is the embedded representation of one type, while n, m, and q represent the number of nodes of type x, r, and s, respectively.

The following is an example of calculating the attention score to types r and s when the current node is x_i:

$$\begin{aligned} a_{x_i type} &= W_a\left[W_x x_i; W_{type} h_{type}\right] \\ \alpha_{x_i type} &= \frac{\exp\left(a_{x_i type}\right)}{\sum_{type' \in type} \exp\left(a_{x_i type'}\right)} \end{aligned} \tag{4}$$

where W_a, W_x, and W_{type} are trainable parameters, and $[\cdot;\cdot]$ denotes the concatenation of vectors.

After calculating the attention scores of the current node to the other two types, it is also necessary to calculate the attention scores of the current node to each neighbor node of both types. The following is an example of computing the attention score to each neighbor node $\{r_j\}_{j=1}^{m}$ or $\{s_k\}_{k=1}^{q}$ when the current node is x_i:

$$\begin{aligned} a_{x_i node} &= \alpha_{x_i type} \odot W_w\left[W_p x_i; W_{node} node\right] \\ \alpha_{x_i node} &= \frac{\exp\left(a_{x_i node}\right)}{\sum_{node' \in node} \exp\left(a_{x_i node'}\right)} \end{aligned} \tag{5}$$

where *node* represents a neighbor node of x_i whose type is *type*; W_w, W_p, and W_{node} are trainable parameters; and $\odot$ is the element-wise production.

After the dual-level attention mechanism, the node x_i's representation of the current layer is as follows:

$$x_i\prime = x_i + \sum_{j=1}^{m} \alpha_{x_i r_j} W_r r_j + \sum_{k=1}^{q} \alpha_{x_i s_k} W_s s_k \tag{6}$$

where x_i is the initial embedded representation of node x_i, $x_i\prime$ is the updated node representation, and W_r and W_s are trainable parameters.

A gate mechanism is used to control the updates to the nodes of the previous layer.

$$o_i = sigmoid(W_o[x_i; x_{i\prime}]) \tag{7}$$

where *sigmoid* is the activation function and W_o is a trainable parameter.

Thus, the representation output of the node x_i at the current layer is as follows:

$$\widetilde{x_i} = o_i \odot x_i\prime + (1 - o_i) \odot x_i \tag{8}$$

The same process is used for updating the node representations of types r and s.

4.4. Relation-Specific Object Extraction

The relation-specific object extraction module is applied to each relation in the predefined set of relations. In this part, all possible objects corresponding to the subject and the specific relation are extracted at the same time. The object for the specific relation is extracted after obtaining the updated representation of the sentence's word tokens, relations, and subject tokens. The average vector of all of the subject's tokens serves as the representation of the subject. Rather than only considering words [2], two binary classifiers with words, relations, and subjects are employed to determine the possible object. To define the span of object, a similar likelihood function is optimized [2].

$$p_i^{obj_{start}} = sigmoid\left(W_{obj_{start}} Tanh\left(W_h\left[x_i; r_j; s_k\right] + b_h\right) + b_{obj_{start}}\right) \tag{9}$$

$$p_i^{obj_{end}} = sigmoid\left(W_{obj_{end}} Tanh\left(W_h\left[x_i; r_j; s_k\right] + b_h\right) + b_{obj_{end}}\right) \tag{10}$$

where $p_i^{obj_{start}}$ and $p_i^{obj_{end}}$ represent the chance that the $i^{-}th$ token in the sentence serves as the object's start position or end position, respectively; x_i, r_j, and s_k signify the vector representation of the $i^{-}th$ token of the sentence, relation, and subject after updating, respectively; *Tanh* is the activation function; and $W_{obj_{start}}$, $W_{obj_{end}}$, W_h, $b_{obj_{start}}$, $W_{obj_{end}}$, and b_h are trainable parameters.

5. Experimental Section

In this section, we describe the experiments carried out to confirm and assess the effectiveness of the suggested method, and we contrast the extraction performance on WebNLG to earlier works. The subdivision and comparison experiments on the different types of overlapping are used to demonstrate how well this method performs in solving overlapping relational triples.

5.1. Experimental Setting

5.1.1. Datasets and Evaluation Metrics

To ensure the fairness of experimental comparison, we used the most commonly used benchmark dataset WebNLG [36], as used in [1–3,8–15,37]. The WebNLG dataset was obtained via natural language processing. Although it is large-scale, inexpensive, and portable, the data quality is inferior to that of manually labeled datasets due to the presence of numerous disturbances. Moreover, it is divided into normal, EPO, and SEO, so that people can evaluate the performance of methods in identifying the overlapping. The statistical information of the dataset is shown in Table 1.

Table 1. Statistics of the WebNLG dataset.

Class	WebNLG	
	Train	**Test**
Normal	1596	246
EPO	227	26
SEO	3406	457
All	5019	703
#Relation	171	

As described in prior works mentioned above, the extracted triples were regarded as true if and only if the relation and two entities were exactly matched. We report the standard micro-precision (Prec.), recall (Rec.), and F1 score (F1) in line with all baselines.

5.1.2. Implementation Details

To fairly compare with the baselines, a pretrained BERT model [35] (base-cased (Available at https://huggingface.co/bert-base-cased, accessed on 1 March 2022), with a 12-layer transformer) was employed and its hyperparameters were set to default. The number of parameters of the BERT model is shown in Table 2, where 28,996 is the number of words in vocab, 768 is the hidden size, 512 is the max positional embeddings, and 3072 is the feedforward/filter size. PyTorch was used to construct this model, and SGD was used to optimize the parameters. Specifically, the threshold of the possibility of a word becoming the start position or end position of the entity was set to 0.5, and the max length of input sentences was set to 100. The training batch size was 6, the learning rate was set to 0.1 and the weight decay was set to 2×10^{-4}. Moreover, the momentum was set to 0.8 on the dataset. The parameter settings are shown in Table 3.

Table 2. The number of parameters of the BERT model.

BERT Model	The Number of Parameters
BertEmbeddings	
word_embeddings	$28{,}996 \times 768$
position_embeddings	512×768
token_type_embeddings	2×768
LayerNorm	$768 + 768$
BertEncoder	
$12 \times$ BertLayer	
$12 \times$ attention	$12 \times [(768 \times 768) + 768 + (768 \times 768) + 768 + (768 \times 768) + 768 + (768 \times 768) + 768 + 768 + 768)]$
$12 \times$ intermediate	$12 \times [768 + (3072 \times 768) + 3072]$
$12 \times$ output	$12 \times [(768 \times 3072) + 768 + 768 + 768]$
BertPooler	
Linear	$(768 \times 768) + 768$
#Total	108,310,272

Table 3. Hyperparameters of the ERHGA.

Hyperparameters	Values
Tags threshold	0.5
Max length of input sentences	100
Training batch size	6
Learning rate	0.1
Weight decay	2×10^{-4}
Momentum	0.8

5.2. Experimental Results

5.2.1. Main Results

Table 4 shows the performance comparison results of the joint entity and relation extraction between our model and the above baselines on the WebNLG dataset. It can be observed that ERHGA is effective and advanced, with superior performance to all baseline models.

Table 4. Results (%) of the proposed ERHGA method with the prior works.

Method	WebNLG		
	Prec.	**Rec.**	**F1**
NovelTagging	52.5	19.3	28.3
CopyRE	37.7	36.4	37.1
GraphRel	44.7	41.1	42.9
CopyMTL	58.0	54.9	56.4
Relation-Aware	66.4	62.7	64.5
CasRel	93.4	90.1	91.8
TPLinker	91.8	92.0	91.9
RIFRE	93.3	92.0	92.6
PRGC	<u>94.0</u>	<u>92.1</u>	<u>93.0</u>
RMAN	83.6	85.3	84.5
SGNet	91.8	91.9	91.9
ERHGA$_{Ø}$	93.1	90.5	91.8
ERHGA	**94.3**	**92.3**	**93.3**

Bold marks the highest score, while underline marks the second best score. The method marked with Ø means unused entity and relation heterogeneous graph attention networks.

Specifically, for the WebNLG dataset, the precision, recall, and F1 score exceeded all baselines and achieved state-of-the-art performance according to our comparisons. The F1 score of the ERHGA model was 0.3% higher than that the of the next best method PRGC, 0.7% higher than that of RIFRE, and 1.5% higher than that of CasRel. Although these baselines have made great contributions, they all ignored the semantic connection and fusion inherent between entities and relations. Our ERHGA method not only considers this aspect but also achieves encouraging results. RIFRE [13] considers the interaction between words and relations. PRGC [3] predicts potential relations in advance, improving the efficiency of the model, but it is undeniable that even if the recall is as high as 96.3% in this step, the overall extraction result will be affected. Moreover, the precision and recall of the ERHGA are 0.3% and 0.2% higher than those of PRGC, respectively.

Compared with ERHGA$_{Ø}$, the experimental results show that ERHGA, which is based on entity and relation heterogeneous graph attention networks, can validly enhance the performance of relational triple extraction. In detail, ERHGA is 1.2%, 1.8%, 1.5% higher than ERHGA$_{Ø}$ in terms of precision, recall, and F1, respectively. All of the above results indicate that ERHGA is effective and advanced for extracting relational triples.

5.2.2. Detailed Results on Different Types of Sentences

To further examine the capability of ERHGA in extracting overlapping relational triples, detailed experiments were carried out on different types of sentences and the experimental results were compared with prior works. Table 5 shows the experimental results of our model and the baseline models on distinct types of overlapping subsets of the dataset. The experimental results highlight that our model achieves the best F1 score on all subsets. The above experimental results demonstrate the ability of our method to recognize overlapping relational triples and the superiority of our method in the identification of relational triples under complex scenarios.

Table 5. F1 (%) on sentences with different overlapping types and different triple numbers.

Method	WebNLG		
	Normal	SEO	EPO
CopyRE	59.2	33.0	36.6
GraphRel	65.8	38.2	40.6
CasRel	89.4	92.2	94.7
TPLinker	87.9	92.5	95.3
RIFRE	90.1	93.1	94.7
PRGC	<u>90.4</u>	<u>93.6</u>	<u>95.9</u>
SGNet	89.2	92.5	95.0
ERHGA	**91.2**	**93.7**	**95.9**

Bold marks the highest score, while underline marks the second best score.

5.3. Analysis and Discussion

5.3.1. Ablation Study

In this section, ablation experiments were conducted to demonstrate the necessity of the entity and relation heterogeneous graph attention network in ERHGA. Table 6 shows the results of different numbers of entity and relation heterogeneous graph attention network layers. As the number of entity and relation heterogeneous graph attention network layers increases, the F1 score first increases and then decreases. For the WebNLG dataset, when the number of layers is 2, the F1 is the best. This illustrates that setting the number of layers to 2 is sufficient and most suitable for the WebNLG dataset. The joint entity and relation extraction method based on entity and relation heterogeneous graph attention networks, where each node learns the semantic information associated with it and updates its own representation, resulted in better performance of the model. However, with the increasing number of layers, the nodes exchanged too much information with one another, absorbing some irrelevant information and impairing the model's performance. Table 7 displays the results of the other hyperparameter settings. Here, 16 groups of hyperparameter settings are taken as examples.

Table 6. F1 (%) of different numbers of entity and relation heterogeneous graph attention network layers.

Number	WebNLG
0	91.8
1	92.5
2	**93.3**
3	91.9

Bold marks the highest score.

Table 7. F1 (%) of different hyperparameter settings.

Different Hyperparameters.	WebNLG
Training batch size = 6, Learning rate = 0.1, Weight decay = 1×10^{-4}, Momentum = 0.8	91.8
Training batch size = 6, Learning rate = 0.1, Weight decay = 1×10^{-4}, Momentum = 0.9	87.2
Training batch size = 6, Learning rate = 0.1, Weight decay =2 $\times$ 10^{-4}, Momentum = 0.8	**93.3**
Training batch size = 6, Learning rate = 0.1, Weight decay = 2×10^{-4}, Momentum = 0.9	91.6
Training batch size = 6, Learning rate = 0.01, Weight decay = 1×10^{-4}, Momentum = 0.8	91.7
Training batch size = 6, Learning rate = 0.01, Weight decay = 1×10^{-4}, Momentum = 0.9	92.3
Training batch size = 6, Learning rate = 0.01, Weight decay = 2×10^{-4}, Momentum = 0.8	91.5
Training batch size = 6, Learning rate = 0.01, Weight decay = 2×10^{-4}, Momentum = 0.9	90.9
Training batch size = 8, Learning rate = 0.1, Weight decay = 1×10^{-4}, Momentum = 0.8	92.9
Training batch size = 8, Learning rate = 0.1, Weight decay = 1×10^{-4}, Momentum = 0.9	92.4
Training batch size = 8, Learning rate = 0.1, Weight decay = 2×10^{-4}, Momentum = 0.8	92.7
Training batch size = 8, Learning rate = 0.1, Weight decay = 2×10^{-4}, Momentum = 0.9	92.0
Training batch size = 8, Learning rate =0.01, Weight decay = 1×10^{-4}, Momentum = 0.8	91.8

Table 7. *Cont.*

Different Hyperparameters.	WebNLG
Training batch size = 8, Learning rate =0.01, Weight decay = 1×10^{-4}, Momentum = 0.9	91.0
Training batch size = 8, Learning rate =0.01, Weight decay = 2×10^{-4}, Momentum = 0.8	91.2
Training batch size = 8, Learning rate =0.01, Weight decay = 2×10^{-4}, Momentum = 0.9	92.1

Bold marks the highest score.

5.3.2. Results on Different Numbers of Relations

As mentioned in previous works [2,15,37], the more relations in the predefined relation set, the greater the challenges faced by the model. Several relations were randomly removed to evaluate the performance of our model under different numbers of relations. The results are shown in Figure 2. The findings indicate that our model performs better when more relations are retained. As the number of relations rises, the F1 score shows an overall growing trend, which reflects the stability of our model and can be applied to complex scenarios.

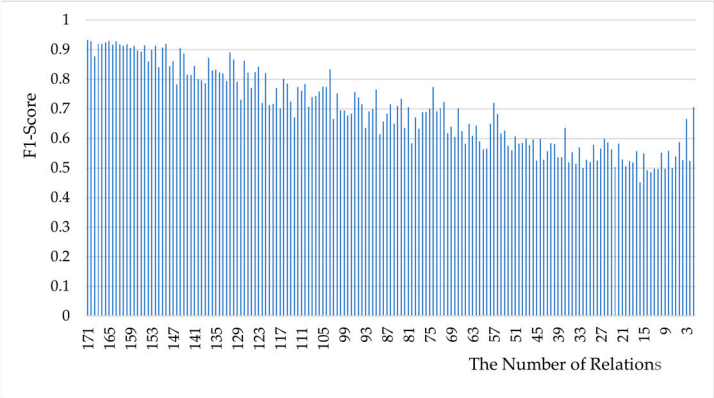


Figure 2. F1 scores with different numbers of relations.

5.3.3. Error Analysis

To investigate the variables that influence the effectiveness of ERHGA, the extraction performance of triple elements was analyzed. (E1, R, E2) was used to represent three elements of the triple. When there was one element to be tested, as long as this element was correct, it was considered to be correct, regardless of the other two. When there were two elements to be tested, as long as the two elements were correct at the same time, it was considered to be correct, regardless of the remaining one.

Table 8 shows the results of the relational triple elements on ERHGA and PRGC. From the results of E1, E2, and R, it can be seen that ERHGA has excellent extraction performance for each element, especially the extraction precision of subjects (reaching 98.9%) and the extraction precision of objects (reaching 98.2%). ERHGA and PRGC mark subjects and objects in different ways; the BIO tag scheme was used to identify the tag of each token in PRGC. ERHGA learns and fuses representations of parts for the extraction of relational triples before object extraction, and the F1 of ERHGA in object extraction was 14.1% higher than that of PRGC. The extraction performance of the subjects and the extraction performance of the objects were better than that of the relations. The reason for this was that the WebNLG dataset has 171 relations; this large number and the complex situation bring some difficulties to the identification of relations. The relation-specific object extraction module is applied to each relation in ERHGA, whereas PRGC predicts potential relations in advance. Even though PRGC believes that predicting potential relations in advance improves the efficiency of the model, and the recall rate is more important in this step, it still affects the overall performance. The performance of (R, E2) and (E1, E2) demonstrates the role of our proposed method in identifying object entity

mentions. Our method incorporating subjects and relations as a priori information lessens the detection of semantically irrelevant objects in the sentence. From the results of (E1, R), (R, E2), (E1, E2), and (E1, R, E2), the F1 of (E1, R, E2) is lower than that of the other three; making the alignment between entities and relations explicit is the crux for the extraction of relational triples.

Table 8. Results (%) of relational triple elements on ERHGA and PRGC.

Element	ERHGA			PRGC		
	Prec.	Rec.	F1	Prec.	Rec.	F1
E1	98.9	95.1	96.9	69.4	96.3	80.7
E2	98.2	94.4	96.3	72.1	95.7	82.2
R	97.1	93.4	95.2	92.8	96.2	94.5
(E1, R)	95.4	92.6	94.0	-	-	-
(R, E2)	96.1	93.0	94.5	-	-	-
(E1, E2)	96.1	93.6	94.8	96.0	92.1	94.7
(E1, R, E2)	94.3	92.3	93.3	94.0	92.1	93.0

5.3.4. Case Study

Table 9 shows the case study of our proposed ERHGA. It can be seen from the first and second examples that our proposed model has an outstanding extraction of SEO and EPO. In the third example, our model identified the relation between "Azerbaijan" and "Rasizade" as "leaderName". According to the annotation of the original dataset, the relation between them should be "leader". However, in the second example, our model can accurately distinguish "leader" and "leaderName". We found that "leaderName" contains more samples than "leader" by revisiting the original dataset. The class imbalance in the dataset itself will affect the performance of our model. In the fourth example, our model recognized the relational triples (Lanka, currency, Rupee) and (Hall, location, Haputale). Although these triples are not marked in the dataset, through reading this example, the semantics of the sentence actually express these two relational triples. As analyzed in Section 5.1.1., the annotated triples of the dataset are not completely accurate. This shows that our model can understand semantics better, is more in line with human cognition, and has a certain generalization ability.

Table 9. Case study for ERHGA.

Sentences	Ground Truth	Predictive Value
Sri Jayawardenepura Kotte is the capital of Sri Lanka, where Ampara Hospital is located.	(Hospital, country, Lanka), (Lanka, capital, Kotte)	(Hospital, country, Lanka), (Lanka, capital, Kotte)
Ampara Hospital is in Eastern Province, Sri Lanka. Ranil Wickremesinghe is a leader there.	(Hospital, country, Lanka), (Lanka, leaderName, Wickremesinghe), (Hospital, state, Lanka)	(Hospital, country, Lanka), (Lanka, leaderName, Wickremesinghe), (Hospital, state, Lanka)
Baku is the capital of Azerbaijan and the leader of the legislature (the National Assembly) is Artur Rasizade. Azerbaijan is the location of the Baku Turkish Martyrs memorial.	(Azerbaijan, capital, Baku) (Azerbaijan, leader, Rasizade)	(Azerbaijan, capital, Baku) (Azerbaijan, leaderName, Rasizade)
Adisham Hall is located in Haputale, Sri Lanka. The capital of Sri Lanka is Sri Jayawardenepura Kotte, the language used in the country is Tamil and the currency is the Sri Lankan Rupee.	(Lanka, capital, Kotte) (Hall, country, Lanka) (Lanka, language, language)	(Lanka, capital, Kotte) (Lanka, currency, Rupee) (Hall, country, Lanka) (Lanka, language, language) (Hall, location, Haputale)

6. Conclusions

In this paper, a novel entity and relation heterogeneous graph attention network ERHGA is proposed for joint entity and relation extraction. Unlike existing approaches, ERHGA incorporates subjects and relations as a priori information when extracting the objects by paying greater attention to the subjects and the relations. At the same time, since the subjects and relations are two types of information for the objects and play different roles in extracting objects, the subjects and relations are modeled as nodes on the heterogeneous graph. To verify the effectiveness of ERHGA, extensive experiments were conducted on the widely used open dataset WebNLG. The experimental results demonstrate that our method achieved outstanding performance and exceeded all baselines. The experimental results of subdivision show that it can powerfully extract overlapping triples. In the future, ERHGA can be applied to extract relational triples in different fields and construct the knowledge graphs of related tasks.

Author Contributions: Conceptualization, B.J.; methodology, B.J.; software, B.J.; validation, B.J.; formal analysis, B.J.; investigation, B.J.; resources, J.C.; writing—original draft preparation, B.J.; writing—review and editing, B.J.; visualization, B.J.; funding acquisition, J.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, grant number 61602042.

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 authors declare no conflict of interest.

References

1. 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, Melbourne, Australia, 15–20 July 2018; pp. 506–514.
2. Wei, Z.P.; 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, Seattle, WA, USA, 5–10 July 2020; pp. 1476–1488.
3. Zheng, H.; Wen, R.; Chen, X.; Yang, Y.; Zhang, Y.; Zhang, Z.; Zhang, N.; Qin, B.; Xu, M.; Zheng, Y. PRGC: Potential Relation and Global Correspondence Based Joint Relational Triple Extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Bangkok, Thailand, 1–6 August 2021; pp. 6225–6235.
4. Hu, L.; Yang, T.; Shi, C.; Ji, H.; Li, X. Heterogeneous Graph Attention Networks for Semi-Supervised Short Text Classification. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019; pp. 4823–4832.
5. Yang, T.; Hu, L.; Shi, C.; Ji, H.; Li, X.; Nie, L. HGAT: Heterogeneous Graph Attention Networks for Semi-Supervised Short Text Classification. *ACM Trans. Inf. Syst* **2021**, *39*, 32. [CrossRef]
6. Li, Z.; Liu, H.; Zhang, Z.; Liu, T.; Xiong, N.N. Learning Knowledge Graph Embedding with Heterogeneous Relation Attention Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *33*, 3961–3973. [CrossRef] [PubMed]
7. Xie, Y.; Xu, H.; Li, J.; Yang, C.; Gao, K. Heterogeneous Graph Neural Networks for Noisy Few-Shot Relation Classification. *Knowl.-Based Syst.* **2020**, *194*, 105548. [CrossRef]
8. 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, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 1227–1236.
9. 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.
10. 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; pp. 9507–9514.
11. Hong, Y.; Liu, Y.; Yang, S.; Zhang, K.; Wen, A.; Hu, J. Improving Graph Convolutional Networks Based on Relation-Aware Attention for End-to-End Relation Extraction. *IEEE Access* **2020**, *8*, 51315–51323. [CrossRef]
12. 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, 13–18 September 2020; pp. 1572–1582.

13. Zhao, K.; Xu, H.; Cheng, Y.; Li, X.; Gao, K. Representation Iterative Fusion Based on Heterogeneous Graph Neural Network for Joint Entity and Relation Extraction. *Knowl.-Based Syst.* **2021**, *219*, 106888. [CrossRef]
14. 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]
15. Liang, J.; He, Q.; Zhang, D.; Fan, S. Extraction of Joint Entity and Relationships with Soft Pruning and GlobalPointer. *Appl. Sci.* **2022**, *12*, 6361. [CrossRef]
16. Nadeau, D.; Sekine, S. A Survey of Named Entity Recognition and Classification. *Lingvisticæ Investig.* **2007**, *30*, 3–26. [CrossRef]
17. Gormley, M.R.; Yu, M.; Dredze, M. Improved Relation Extraction with Feature-Rich Compositional Embedding Models. *arXiv* **2015**, arXiv:1505.02419.
18. Cai, R.; Zhang, X.; Wang, H. Bidirectional Recurrent Convolutional Neural Network for Relation Classification. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Berlin, Germany, 7–12 August 2016; pp. 756–765.
19. Christopoulou, F.; Miwa, M.; Ananiadou, S. A Walk-Based Model on Entity Graphs for Relation Extraction. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; pp. 81–88.
20. Qin, P.; Xu, W.; Wang, W.Y. Robust Distant Supervision Relation Extraction via Deep Reinforcement Learning. *arXiv* **2018**, arXiv:1805.09927.
21. 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, Berlin, Germany, 7–12 August 2016; pp. 1105–1116.
22. 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, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 917–928.
23. 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, Doha, Qatar, 25–29 October 2014; pp. 1858–1869.
24. 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).
25. 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]
26. Ma, L.; Ren, H.; Zhang, X. Effective Cascade Dual-Decoder Model for Joint Entity and Relation Extraction. *arXiv* **2021**, arXiv:2106.14163.
27. El-allaly, E.; Sarrouti, M.; En-Nahnahi, N.; Ouatik El Alaoui, S. An Attentive Joint Model with Transformer-Based Weighted Graph Convolutional Network for Extracting Adverse Drug Event Relation. *J. Biomed. Inform.* **2022**, *125*, 103968. [CrossRef] [PubMed]
28. Sun, H.; Grishman, R. Employing Lexicalized Dependency Paths for Active Learning of Relation Extraction. *Intell. Autom. Soft Comput.* **2022**, *34*, 1415–1423. [CrossRef]
29. Chen, W.; Gu, Y.; Ren, Z.; He, X.; Xie, H.; Guo, T.; Yin, D.; Zhang, Y. Semi-Supervised User Profiling with Heterogeneous Graph Attention Networks. In Proceedings of the Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, Macao, China, 10–16 August 2019; pp. 2116–2122.
30. Huang, Q.; Yu, J.; Wu, J.; Wang, B. Heterogeneous Graph Attention Networks for Early Detection of Rumors on Twitter. In Proceedings of the 2020 International Joint Conference on Neural Networks, Glasgow, UK, 19–24 July 2020; pp. 1–8.
31. Cao, Y.; Peng, H.; Wu, J.; Dou, Y.; Li, J.; Yu, P.S. Knowledge-Preserving Incremental Social Event Detection via Heterogeneous GNNs. In Proceedings of the Web Conference 2021, Ljubljana Slovenia, 19–23 April 2021; ACM: New York, NY, USA, 2021; pp. 3383–3395. [CrossRef]
32. Pang, Y.; Wu, L.; Shen, Q.; Zhang, Y.; Wei, Z.; Xu, F.; Chang, E.; Long, B.; Pei, J. Heterogeneous Global Graph Neural Networks for Personalized Session-Based Recommendation. In Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining, held virtually, Tempe, AZ, USA, 15 February 2022; pp. 775–783. [CrossRef]
33. Lu, G.; Li, J.; Wei, J. Aspect Sentiment Analysis with Heterogeneous Graph Neural Networks. *Inf. Process. Manag.* **2022**, *59*, 102953. [CrossRef]
34. Liang, Y.; Meng, F.; Zhang, Y.; Chen, Y.; Xu, J.; Zhou, J. Emotional Conversation Generation with Heterogeneous Graph Neural Network. *Artif. Intell.* **2022**, *308*, 103714. [CrossRef]
35. 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, Minneapolis, MN, USA, 2–7 June 2019; pp. 4171–4186.
36. 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, Vancouver, BC, Canada, 30 July–4 August 2017; pp. 179–188.
37. Chen, T.; Zhou, L.; Wang, N.; Chen, X. Joint Entity and Relation Extraction with Position-Aware Attention and Relation Embedding. *Appl. Soft Comput.* **2022**, *119*, 108604. [CrossRef]

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.