

Article

Multi-Level Attention with 2D Table-Filling for Joint Entity-Relation Extraction

Zhenyu Zhang, Lin Shi [†], Yang Yuan [†], Huanyue Zhou [†] and Shoukun Xu ^{*,†}

Computer and Artificial Intelligence, Alibaba Cloud Big Data College, Changzhou University, Changzhou 213164, China

* Correspondence: xsk@cczu.edu.cn

† These authors contributed equally to this work.

Abstract: Joint entity-relation extraction is a fundamental task in the construction of large-scale knowledge graphs. This task relies not only on the semantics of the text span but also on its intricate connections, including classification and structural details that most previous models overlook. In this paper, we propose the incorporation of this information into the learning process. Specifically, we design a novel two-dimensional word-pair tagging method to define the task of entity and relation extraction. This allows type markers to focus on text tokens, gathering information for their corresponding spans. Additionally, we introduce a multi-level attention neural network to enhance its capacity to perceive structure-aware features. Our experiments show that our approach can overcome the limitations of earlier tagging methods and yield more accurate results. We evaluate our model using three different datasets: SciERC, ADE, and CoNLL04. Our model demonstrates competitive performance compared to the state-of-the-art, surpassing other approaches across the majority of evaluated metrics.

Keywords: named entity recognition; relation extraction; word-pair tagging; multi-level attention neural network



Citation: Zhang, Z.; Shi, L.; Yuan, Y.; Zhou, H.; Xu, S. Multi-Level Attention with 2D Table-Filling for Joint Entity-Relation Extraction. *Information* **2024**, *15*, 407. <https://doi.org/10.3390/info15070407>

Academic Editor: David Martins de Matos

Received: 20 June 2024

Revised: 8 July 2024

Accepted: 12 July 2024

Published: 14 July 2024



Copyright: © 2024 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

Named Entity Recognition (NER) and Relation Extraction (RE) aim to extract structured information from plain texts. They are long-standing research topics in the field of Natural Language Processing (NLP). We present Figure 1 as an example of the NER and RE problem: NER aims to identify entities in text and classify them into pre-defined entity types, for example, “Reagan” should be recognized as a person (Peop) and “U.S.” as a location (Loc), respectively. On the other hand, RE is usually based on the entities that have been identified by NER, combined with contextual semantic information, to assign a relation type to these entities. For instance, a “Live_In” relation exists between “Reagan” and “U.S.”.

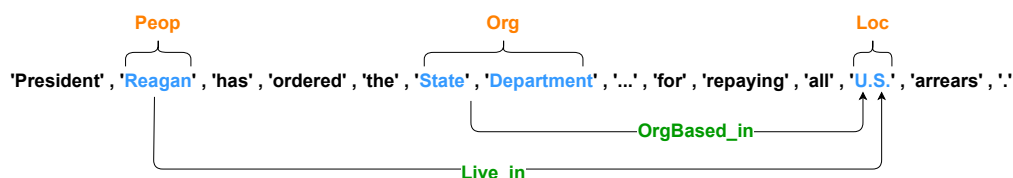


Figure 1. An illustrative example of the entities and relations extraction task.

Methods for entities and relations extraction can be categorized into pipeline or joint models. In the traditional pipeline approach [1–4], NER and RE are considered as two independent tasks: first, entities are recognized in the input sentence, and then relations are classified as pairs of extracted entities. Joint works [5–10] extract entities and relations

in parallel, then combine them into triples and avoid the error propagation caused by the pipeline framework.

Many joint methods focus on learning a unified representation of these two tasks to explore the correlation between NER and RE. Given the exceptional performance of Pre-trained Language Models (PLMs) like BERT [11], which can help mitigate problems, such as limited semantic elements within a sentence, researchers can maximize the utility of BERT to extract more complex features. Some works [3,12,13] have focused on exploring methods to obtain improved span representations from pre-trained encoders. For example, ref. [13] proposes a simple and effective way to capture span representations through BERT for lightweight reasoning. Ref. [4] introduces a novel span representation approach to consider the interrelation between the spans (pairs) by strategically packing the markers in the encoder. These approaches often heavily rely on predefined features (span features), causing the model to overlook the intricate interconnections among the entities and relations, thereby impeding the recognition of semantic relations between entity pairs.

To explore the common structure of the two tasks, table-filling methods have been proposed, wherein unit features are defined as the basic semantic properties of the target word pair [8,14–17]. In this approach, the (i, j, r) -th cell is assigned a label that represents the relationship between tokens at positions (i, j) in the sentence. To this end, for an input sentence, the output of the method is usually a three-dimensional (3D) matrix with each entry corresponding to the classification result. These approaches built upon the table structure operate on the idea that cell labels are dependent on features or predictions derived from preceding or adjacent cells. Ref. [8] formulates joint extraction as a token pair linking problem and introduces an innovative handshaking tagging scheme that aligns the boundary tokens of entity pairs for each relation type. Ref. [14] proposes to eliminate the different treatments on the two sub-tasks' label spaces and applies a unified classifier to predict each cell's label. In their approach, entities and relations are represented by squares and rectangles in the table. Ref. [15] employs a scoring-based classifier and a relation-specific horn tagging strategy. However, the information from type markers is not utilized in these methods.

In our study, we propose leveraging a pre-trained encoder to enhance the model's semantic information with features linked to the target information. This encompasses entity and relation type markers, along with structural details. Specifically, inspired by the works above and the interaction map proposed by [16], we design a new word-pair tagging method to extract all results in one step. The input of our model is a two-dimensional (2D) table, with each entry corresponding to a word pair in sentences. A detailed description of our word-pair tagging can be found in Figure 2. Furthermore, we design a multi-level attention network joint extraction model: First, we facilitate multi-head biaffine auxiliary alignment between objects to discern correlations between units. Then, we combine table structure-aware features with sequence-aware features, thereby capturing connections between unit features while providing the model with both textual semantic information and task-related details. Our model predicts the most probable results from the word-pair tagging table by calculating the attention score. In general, our main contributions are as follows:

- We incorporate the type markers alongside text tokens in the same encoder, thus preserving task relevance rather than treating them as isolated components. Building upon a novel word-pair tagging approach, we condense our table into two dimensions.
- We propose a multi-level attention mechanism that models interactions around unit features, capturing dependencies between table structure-aware and sequence-aware features. This mechanism effectively integrates the inherent relationships between feature sequences relevant to entities or relations, while maintaining the efficiency advantage of the model.

	President	Reagan	has	ordered	the	State	Department	...	U.S.	arrears	Peop	Org	Loc	Live_In	OrgBased_In
President	1														
Reagan		1							1		1			1	
has			1												
ordered				1											
the					1										
State						1			1			1			1
Department							1								
...								1							
U.S.		1				1			1				1		
arrears										1					
Peop		1													
Org							1								
Loc									1						
Live_In									1						
OrgBased_In									1						

Figure 2. The marks in the table are represented as entities and relations in the sentence of Figure 1. The model outputs individual scores for each table element, which represent the relationships between word pairs.

2. Related Work

In recent years, many works [18–20] have considered the joint modeling of entity recognition and relation extraction tasks and largely focused on developing effective prediction models. Joint extraction of entity and relation mitigates the error propagation issue associated with the traditional pipeline approach and leverages the interaction between tasks, resulting in improved performance. Furthermore, some problems attract much attention from researchers:

- **Overlapping:** Based on the different overlapping patterns of triples [21], sentences can be divided into three categories, as suggested by [22]: Normal, Entity Pair Overlap (EPO) and Single Entity Overlap (SEO). A sentence is classified as Normal if none of its triples have overlapping entities. It is categorized as EPO if some of its triples have overlapping entity pairs. Meanwhile, a sentence falls into the SEO class if some of its triples have an overlapping entity but do not have overlapping entity pairs. Note that a sentence can belong to both the EPO and SEO classes.
- **Interaction:** Since these tasks are closely interconnected, joint models capable of simultaneously extracting entities and their relations within a single framework have the potential to leverage inter-task correlations and dependencies, leading to potential performance improvements. Several recent efforts have aimed to exploit such inter-task correlations by jointly modeling both NER and RE tasks.

Some approaches like token-level models [23,24] using the BIO tagging scheme face challenges in modeling overlapping entity mentions and often encounter cascading errors due to sequential decoding. The span-based approach [25] identifies overlapping entities by determining the boundaries of objects and then categorizing them based on these boundaries. However, span-based models are affected by maximal span lengths, and a sentence including n words may consist of $n(n + 1)/2$ numbers of entity possibilities. In previous works, ref. [13] width embeddings were set and learned through backpropagation, while [3] the process span pairs with levitated markers independently, which is time-consuming and overlooks the interrelation between the span pairs.

Earlier work [26] in this area commonly reduces the task to a table-filling problem to be useful in addressing overlapping and interaction problems. However, these methods usually required an additional expensive decoding step to obtain globally consistent cell labels. In the work by [27], a novel neural architecture was introduced, which utilized the table structure and involved repeated applications of 2D convolutions for pooling local dependency and metric-based features. Another work [28] proposed a global feature-oriented triple extraction model that fully leveraged the global associations. Each relation's table is filled based on its refined table feature, and all triples linked to this relation are extracted based on its filled table.

This paper introduces a two-dimensional table to represent interactions between individual words in a sentence. Our method leverages both the table structure within the 2D table representation and the sequence structure information within the text. We facilitate interaction between these elements with our multi-level attention architecture, especially considering the context of neighboring entries in the table.

3. Methods

In this section, we first detail the joint extraction of entities and relations tasks and our word-pair tagging method (Sections 3.1 and 3.2). Then, we describe our contextualized word representations based on pre-trained language models (Section 3.3) and introduce our multi-level attention for table-filling tasks (Section 3.4). Finally, we introduce the training methods to extract entities and relations (Section 3.5). Figure 2 shows a detailed description of our word-pair tagging, and Figure 3 shows an overview of our model architecture.

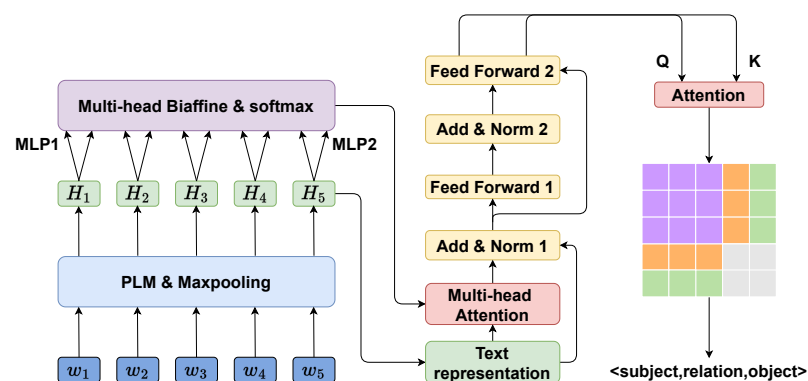


Figure 3. An overview of our model architecture, consisting of four main modules: 1. Max-pooling aggregation module: uses a pre-trained language model (PLM) and max-pooling for contextualized representations. 2. Table structural-aware module: derives head and tail representations with MLPs and computes word-pair representations using a multi-head biaffine model. 3. Context-table fusion module: applies multi-head attention to combine the weighted and original sequences. 4. Sequence-aware module: encodes the sequence with FFNN layers and residual structures, followed by a non-linear transformation for relationship prediction.

3.1. Task Description

Given a sentence S of words w_1, w_2, \dots, w_n as input, the model is required to extract related entities and to identify the relation types between entities to form a set of triplets identifying pairs in the form of $(e_1^{t_1}, r, e_2^{t_2})$, where e_1 is not equal to e_2 . An entity e_1/e_2 is a span with the pre-defined entity types t_1/t_2 . The r represents the relation between the entities e_1 and e_2 . The task requires the model to correctly predict the boundaries of the subject entity and the object entity, and the entity relation.

3.2. Word-Pair Tagging

We propose a new word-pair tagging method, thereby transforming the task into one that extracts the predicted results between each word-pair (w_i, w_j) . By concatenating text and task label types into a natural language sequence, our model can exploit their

contextualized correlations and leverage the semantic knowledge learned from the pre-trained language model. These markers will be explained further below:

- Diagonal markers in the purple part indicate entity-head and entity-tail. The orange part on the right represents the connection between an entity-head and an entity type. Similarly, the orange part below the table represents the connection between an entity-tail and an entity type. When both the entity-head and entity-tail have the same entity type, they can form an entity. The table exactly expresses how to detect the correct span boundary of the spans, as shown in Figure 2, where (“Reagan”, Peop), (“State Department”, Org) and (“U.S.”, Loc) can be extracted.
- Out-of-diagonal markers in the purple part indicate subjects and objects. The green part on the right represents the connection between a subject and a relation type, while the green part below the table represents the connection between an object and a relation type. If a subject and object share the same relation type, they can form relational triples. Therefrom, the table can exactly express overlapped relations, e.g., the location entity “U.S.” participates in two relations, (“Reagan”, “U.S.”, Live_In) and (“State Department”, “U.S.”, OrgBased_In).

By combining the extraction of entity and relation parts, we successfully extract complete relational triples ($Reagan_{Peop}$, Live_In, $U.S._{Org}$).

3.3. Text Representation

In our approach, we enhance the input sequence by appending entity and relation type markers, which distinguishes our method from standard BERT models that process only raw text augmented with [CLS] and [SEP] tokens. Specifically, given an input sentence with n words (e.g., $S = \{w_1, w_2, \dots, w_i, w_n\}$, where the sentence length is n , and entity types (e.g., $E = \{t_{e1}, t_{e2}, \dots, t_{en}\}$) and relation types (e.g., $R = \{t_{r1}, t_{r2}, \dots, t_{rn}\}$), we provide the combined sequence of the text and the inserted type markers to the PLM (e.g., BERT) to obtain the contextualized representations, and the sequence length becomes $L = tn + 2 + en + rn$ (including [CLS] and [SEP], two special start and end markers):

$$H' = BERT([CLS], t_{t1}, t_{t2}, \dots, t_{tn}, [SEP], t_{e1}, t_{e2}, \dots, t_{en}, t_{r1}, t_{r2}, \dots, t_{rn}) \quad (1)$$

where $H' \in R^{L \times d}$ is the context-aware embedding of tokens, where tn is the sum of word pieces in the sentence after the segmentation (e.g., Mondrian \rightarrow Mon, ##dr, ##ian), en is the number of entity types, rn is the number of relation types, and d is the dimension of hidden units in the BERT model. These markers are integrated into the input sequence, providing contextual cues that are absent in traditional BERT inputs, thereby enabling the PLM to leverage semantic and relational metadata along with textual information. After that, we compute the embedding of each word by max-pooling its composing tokens to aggregate information for their associated spans. If a word is split into multiple word pieces, we use the max-pooling of all piece vectors as its word representation. Finally, the length of sequence representation H becomes $n + 2 + en + rn$.

3.4. Multi-Level Attention Encoder

Our multi-level attention encoder consists of a table structure-aware module, context-table fusion modules, a and sequence-aware module. Our model takes the sequence representation H obtained in Section 3.2 as input, and its output is used to predict both entities and relations in sentences.

To ensure that text representations are shared between the entity and relation types, we adopt a table structure-aware module. Initially, we apply two multi-layer perceptrons (MLPs) on the pre-trained feature vector H to obtain separate representations for head-and-tail parts of an entity or relation. We split the representations H_i and H_j obtained from the MLPs into multiple heads. Then, a multi-head biaffine model is leveraged to obtain representations of word pairs (h_i, h_j) . Next, we concatenate the representations from all heads to obtain H_T and apply a softmax activation function to H_T . The resulting H_T serves

as the weight information for the sequence, containing both context information and table structure. The calculation formula for this process is as follows:

$$H_i, H_j = MLP_1(H), MLP_2(H) \tag{2}$$

$$h_i^{(k)}, h_j^{(k)} = Split(H_i), Split(H_j) \tag{3}$$

$$h^k[i, j] = (h_i^{(k)})^T U h_j^{(k)} \tag{4}$$

$$H_T = Concat(h^{(1)}, \dots, h^{(n)}) \tag{5}$$

$$H_T = Softmax(H_T) \tag{6}$$

where $H_i, H_j \in R^{n \times h}$, n is the length of a sentence, h is the hidden size, $Split(\cdot)$ equally splits a matrix in the last dimension, $h_i^{(k)}, h_j^{(k)} \in R^{n \times h_k}$, h_k is the hidden size for each head, U is a $n \times r \times n$ trainable parameter, r is the number of heads, and $H_T \in R^{n \times n \times r}$.

We then perform multi-head attention calculations using the weight information and sequence information as our context-table fusion modules, obtaining the new sequence representation S :

$$S = H_T \times H \tag{7}$$

where $S \in R^{n \times h}$. In the final sequence-aware module, we use two separate feed-forward neural network (FFNN) layers with the residual structure to encode representations S . The interaction function is defined as follows:

$$S = Layernorm(Relu(Linear(S)) + H) \tag{8}$$

$$S = Layernorm(FFNN(S) + S) \tag{9}$$

Finally, we transform the features S through a non-linear transformation Q and K and calculate the attention score to generate a predicted score for each relationship of the 2D word-pair:

$$Q = Linear_1(S) \tag{10}$$

$$K = Linear_2(S) \tag{11}$$

$$P = \sigma(QK^T) \tag{12}$$

where $p \in R^{n \times n}$ is the interaction matrix for prediction results, n means the length of sentence, each entry corresponds to a word-pair, σ is a sigmoid function, and we consider $P(\cdot)$ valid when the value of $P(\cdot)$ exceeds threshold $\sigma(\sigma > 0.5)$. The representation P_{ij} of the word-pair (x_i, x_j) can be considered as a combination of the representation h_i of x_i and h_j of x_j .

3.5. Training

Given the input and its gold label y' (0 or 1), the binary cross entropy loss is used for training:

$$\mathcal{L} = \sum_{i=1}^{n \times n} BCELoss(y, y') \tag{13}$$

where y is the predicted results, and n is the length of the sentence.

4. Results

In this section, we present the experimental part, including the datasets, evaluation metrics, and experiment settings to evaluate the performance of our proposed model for entity and relation extraction. Additionally, we conduct exhaustive ablation studies to further investigate the effectiveness of the model.

4.1. Datasets

To evaluate the performance of our proposed method, we tested it across three datasets from different domains, namely SciERC, ADE and CoNLL04:

SciERC: ref. [29] is derived from 500 AI paper abstracts and defines scientific terms and relations specifically for scientific knowledge graph construction. This dataset includes six scientific entities, including task, method, metric, material, other-scientific-term, generic and seven relation types, including compare, conjunction, evaluate-for, used-for, feature-of, part-of, hyponym-of, and includes 2687 sentences. We adopt the official training (1861 sentences)/validation (275 sentences)/testing (551 sentences) splits.

ADE: ref. [30] propose the Adverse Drug Events (ADE) dataset for extracting drug-related adverse effects from medical text, which focuses on one relation category and two entity categories, including drug and adverse-effect. ADE consists of 4272 sentences and 6821 relations, these sentences describe the adverse effects arising from drug use. Given there are no official train-test splits, we report the mean performance based on 10-fold cross-validation, where results are based on averaging performance across the ten folds, as in prior work.

CoNLL04: ref. [31] contains 1441 sentences with annotated named entities and relations extracted from news articles. It has four entity categories, including person, location, organization, and other, and five relation categories, including Live_In, Located_In, OrgBased_In, Work_For, and kill. We employ the training (1153 sentences) and test set (288 sentences), where 20% of the training set is used as a held-out development part, which is consistent with [13,32]. This dataset contains no overlapping entities.

4.2. Evaluation Metrics

We evaluate these models on both entity recognition and relation extraction tasks, following the approach of prior work. For the NER task, an entity is considered correct if its predicted boundary and type match the ground-truth. For the RE task, previous works have used different metrics: (1) boundaries evaluation (Re), where a relation is considered correct if its relation type, as well as the two related entities, are both correct, without considering the correctness of the entity type; (2) strict evaluation (Re+), where a predicted relation is treated as a true positive if it is exactly matched to a relation in the ground truth based on boundaries and type of subject/object entities and relation type. For the convenience of comparison, we report multiple evaluation metrics consistent with them. In our experiments on these datasets, we report a micro-F1 score for the ADE and CoNLL04 datasets, and we also report the macro-F1 score.

4.3. Experiment Settings

For fair comparison, we used bert-base-cased as the encoder on most datasets and replaced with scibert-scivocab-uncased for the SciERC dataset. We fixed the length of the input sentence to 100. We employed multi-head biaffine decoding with heads = 4 and embedding size = 300. The Adam Optimizer [33] is used with a linear warmup-decay learning rate schedule. We trained the entity model for 100 epochs with a learning rate of 1×10^{-5} for all experiments. To mitigate overfitting, we applied a dropout strategy with a rate set between 0.2 and 0.4. We used a batch size of 4/20 for SciERC/other datasets, respectively. In our experiments, we ran all experiments with five different seeds and reported the average score.

4.4. Results

Tables 1–3 present the test set evaluation results for the SciERC, ADE, and CoNLL04 datasets.

Regarding entity recognition, our model achieves an absolute F1-score improvement of +0.1% on the SciERC dataset and +0.52% on the ADE dataset, using the ALBERT PLM. In our experiments on the CoNLL04 dataset, our model demonstrates notable improvements and competitive performance across various metrics. Notably, under the macro metric, our

model exhibits a precision advantage over the best-reported model [34] and achieves an F1-score enhancement of 0.68% compared to the second-best model [35]. Additionally, our approach yields competitive results in terms of Micro-F1 values. This demonstrates that entity-type information is useful for the entity model, and pre-trained transformer encoders are able to capture long-range dependencies from context.

For relation extraction, our approach outperforms the best previous methods by an absolute F1 of +0.7% and +1.2% on the SciERC dataset for RE and RE+ tasks, respectively. Additionally, we achieve +1.12% and +2.27% F1-score improvements on the ADE dataset when using bert and albert PLM, respectively. On the CoNLL04 dataset, our model achieved the highest precision and recall across both macro and micro metrics. Our model is competitive without using additional data. Notably, under the micro metric, our model surpassed the second-best performing model [20] with a competitive F1-score improvement of 0.4%.

By comparing the results presented in recent papers, our proposed model attains consistently strong performance over all three datasets, from which we can observe that our word-pair tagging method and learned multi-level features are effective for entity and relation extraction.

Table 1. Overall precision (%), recall (%) and F1-scores (%) on the SciERC dataset, calculated using micro-averages. All methods employ a pre-trained scibert-scivocab-uncased model for feature extraction. Scores highlighted in bold represent the highest values.

Method	Metric	NER	RE	RE+
Luan et al. (2018) [29]	Prec.	67.2	47.6	-
	Rec.	61.5	33.5	-
	F1	64.2	39.3	-
Wadden et al. (2019) [36]	Prec.	-	-	-
	Rec.	-	-	-
	F1	67.5	48.4	-
Eberts and Ulges (2020) [13]	Prec.	70.9	53.4	40.5
	Rec.	69.8	48.5	36.8
	F1	70.3	50.8	38.6
Shen et al. (2021) [37]	Prec.	70.2	52.6	-
	Rec.	70.2	52.3	-
	F1	70.2	52.4	-
Zhong and Chen (2021) [3]	Prec.	-	-	-
	Rec.	-	-	-
	F1	68.9	50.1	36.8
Wang et al. (2021) [14]	Prec.	65.8	-	37.3
	Rec.	71.1	-	36.6
	F1	68.4	-	36.9
Yan et al. (2021) [38]	Prec.	-	-	-
	Rec.	-	-	-
	F1	66.8	-	38.4
Santosh et al. (2021) [39]	Prec.	69.8	51.9	39.9
	Rec.	71.3	50.6	39.0
	F1	70.5	51.3	39.4
Ye et al. (2022) [4]	Prec.	-	-	-
	Rec.	-	-	-
	F1	69.9	53.2	41.6

Table 1. Cont.

Method	Metric	NER	RE	RE+
Jeong et al. (2022) [40]	Prec.	-	-	-
	Rec.	-	-	-
	F1	70.8	45.5	-
Ours	Prec.	71.4	58.3	47.1
	Rec.	69.2	50.1	39.2
	F1	70.9	53.9	42.8

Table 2. Overall precision (%), recall (%) and F1-scores (%) on the ADE dataset. These methods employ the macro metric. Bold marks the highest score.

Method	Model	Score	NER	RE+
Giannis et al. (2018) [41]	-	Prec.	84.72	72.10
		Rec.	88.16	77.24
		F1	86.40	74.58
Eberts and Ulges (2020) [13]	BERT	Prec.	89.26	78.09
		Rec.	89.26	80.43
		F1	89.25	79.24
Wang and Lu (2020) [42]	ALBERT	Prec.	-	-
		Rec.	-	-
		F1	89.70	80.10
Cabot and Navigli (2021) [43]	BART	Prec.	-	81.50
		Rec.	-	83.10
		F1	-	82.20
Yan et al. (2021) [38]	ALBERT	Prec.	-	-
		Rec.	-	-
		F1	91.30	83.20
Crone (2020) [35]	BERT	Prec.	89.06	80.51
		Rec.	89.63	86.81
		F1	89.48	83.74
Zhao et al. (2021) [44]	BERT	Prec.	-	-
		Rec.	-	-
		F1	89.40	81.14
Wan et al. (2023) [20]	BERT	Prec.	-	-
		Rec.	-	-
		F1	91.30	83.07
Wang et al. (2022) [34]	GLM	Prec.	-	-
		Rec.	-	-
		F1	91.1	83.8
Ours	BERT	Prec.	88.39	82.09
		Rec.	93.31	87.82
		F1	90.78	84.86
	ALBERT	Prec.	90.88	83.90
		Rec.	92.78	87.10
		F1	91.82	85.47

Table 3. Overall precision (%), recall(%) and F1-scores (%) on the CoNLL04 dataset. These methods employ a pre-trained bert-base-cased model to obtain feature representations. Bold marks the highest score.

Method	Model	Metric	Score	NER	RE+
Li et al. (2019) [45]	-	Micro	Prec.	89.00	69.20
			Rec.	86.60	68.20
			F1	87.80	68.90
Eberts and Ulges (2020) [13]	BERT	Macro	Prec.	85.78	74.75
			Rec.	86.84	71.52
			F1	86.25	72.87
Crone (2020) [35]	BERT	Macro	Prec.	87.92	77.73
			Rec.	86.42	68.38
			F1	87.00	72.63
Wang and Lu (2020) [42]	ALBERT	Macro	Prec.	-	-
			Rec.	-	-
			F1	86.90	75.40
Cabot and Navigli (2021) [43]	BERT	Macro	Prec.	-	-
			Rec.	-	-
			F1	-	76.65
		Micro	Prec.	-	-
			Rec.	-	-
			F1	-	75.40
Shen et al. (2021) [37]	-	Micro	Prec.	90.30	73.00
			Rec.	90.30	71.60
			F1	90.30	73.60
Zhao et al. (2021) [44]	-	Micro	Prec.	-	-
			Rec.	-	-
			F1	90.62	72.97
Wan et al. (2023) [20]	BERT	Micro	Prec.	-	-
			Rec.	-	-
			F1	91.43	74.39
Wang et al. (2022) [34]	GLM	Macro	Prec.	-	-
			Rec.	-	-
			F1	90.70	78.30
Ours	BERT	Macro	Prec.	90.72	77.56
			Rec.	84.84	75.03
			F1	87.68	76.27
		Micro	Prec.	91.01	75.13
			Rec.	89.58	74.46
			F1	90.28	74.79

5. Ablation Study

Our model basically consists of four modules: max-pooling aggregation module (A), table structural-aware module (B), context-table fusion module (C) and sequence-aware module (D). We report the ablation analysis results for the ADE and SciERC datasets, focusing on the RE+ from Table 4, and the layers of the encoded block are all set to one:

Table 4. Ablation study for ADE and SciERC datasets, focusing on the RE+. Each row after the first indicates the removal of a particular component.

Method	Metric	ADE	SciERC
Our model	Prec.	82.14	47.88
	Rec.	85.88	37.51
	F1	83.97	42.07
A	Prec.	81.98	43.62
	Rec.	85.30	35.75
	F1	83.61	39.29
B&C	Prec.	82.17	47.78
	Rec.	83.77	37.87
	F1	82.96	40.07
D	Prec.	80.78	45.34
	Rec.	83.92	37.51
	F1	82.32	41.26

While the max-pooling aggregation module had a positive effect for F1-score on ADE and SciERC datasets, it also helped the model improve the precision to a certain extent. When removing table structural-aware and context-table fusion modules, for the ADE and SciERC datasets, we find that recall has a large negative impact and approximately decreased by 2.11–3%. When removing the sequence-aware module, the system shows a decrease of 1.65% and 0.81% in F1-scores for the ADE and SciERC datasets, respectively. These results indicate that the BERT encoder itself can capture type-specific dependencies among tokens and labels within its architecture, the joint addition of table structure-aware, context-table fusion and sequence-aware modules have a significant effect on NER and RE improvement.

5.1. Effect of Encode Layers

To investigate whether a deeper module can further model dense interactions over label spaces, we stack multi-level attention units in depth from 0 to 5 on the ADE and SciERC datasets and analyze the performance. The results are presented in Figure 4:

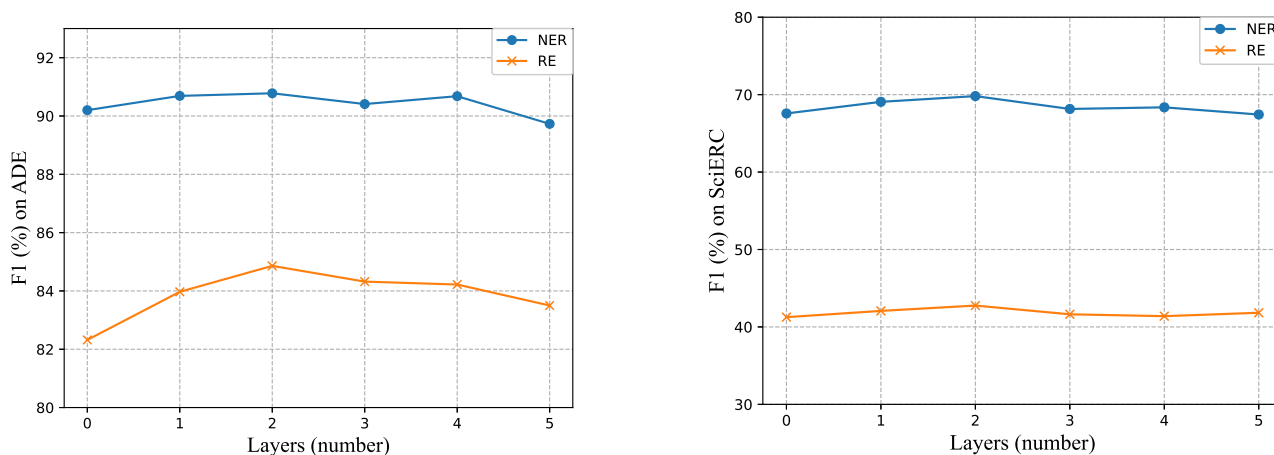


Figure 4. Performances with respect to the number of layers setting on the ADE and SciERC test sets.

In Figure 4, we demonstrate improvements of model performance through adjustments to the model’s layer settings and explore the effect of the superposition of different layers. We found that increasing the number of layers from 0 to 2 leads to a significant improvement in the F1 scores for both tasks. However, we found that the F1 score did not improve further by continuing to increase the number of layers. Therefore, in our final model, we use two layers as the optimum configuration.

5.2. Effect of Table Encoding

In this section, we conducted numerous experiments to explore the performance impact of several different table encoding strategies on entity and relation extraction. Each model utilized in these experiments was structured with two layers. We conducted a study using the ADE dataset, and the experimental results are shown in Table 5:

Table 5. Study on the ADE dataset. The separate-type method employs two tables within the same model. Bold marks the highest score.

Method	NER	RE
Our model	90.78	84.86
Concat	90.09 (−0.69)	83.46 (−1.40)
Multi-heads CNN	90.54 (−0.24)	84.40 (−0.46)
CLN	88.95 (−1.83)	83.92 (−0.94)
Separate-type	87.91 (−2.87)	83.25 (−1.61)

- **Concat:** the concat method represents each word-pair representation via concatenating the corresponding distinct tokens features. While this method collects information at the token level, it overlooks the connections between tokens, leading to coarse-grained formative features. Consequently, using the Concat model leads to a drop in NER and RE+ F1-score performance by 0.69% and 1.4%, respectively.
- **Multi-head CNN:** the convolutional approach is a natural method to merge all of the features, and it might be necessary to utilize all local features and predict scores on a global scale. Fusion features that are composed of correlations between unit features can help the model in capturing local sentence features and in learning connections between features, thus learning semantic structural information in sentences. When constructing the CNN structure, we still employed a two-layer CNN with convolutional kernels of 3, and we set its output dimension number of the decoder to be the same as the number of heads. CNN-based models are effective in capturing local features of adjacent cells, but make it difficult to capture long-distance dependencies. As shown in Table 5, using the multi-head CNN has a small negative impact, with performance declining by 0.24% and 0.46% for NER and RE.
- **CLN:** we use the Conditional Layer Normalization (CLN) proposed in [46], which generates a high-quality representation of the word-pair grid. The layer normalization is conducted in the feature dimension. The results, as displayed in Table 5, show a decrease of 1.83% in NER and a decrease of 0.96% in RE.

The experiments demonstrate that it is necessary to fuse the representations of table structure to predict the entity and relations. Furthermore, the application of multi-head biaffine can enhance the learning of table structural information.

5.3. Effect of Type Information

To examine the influence of the interaction of information on the types of entities and relations, we separate the entity or relation-type sequences from the input sentence to model the two tables independently, denoted as the separate-type model. Specifically, we obtain the sequence embeddings of the input sentence, including the natural language texts of entity type, and the input sentence, including the natural language texts of relations with the same BERT encoder. From these, we generated two tables to jointly decode the predicted two tables. As this method takes entity and relation types as separate inputs, the network can only independently model the correlations of the entity part and relation part, without capturing the interdependencies between task interactions. As shown in Table 5, the separate-type model has marked performance degradation on both tasks compared to a joint-type model, with F1-scores dropping by 2.87% for NER and 1.61% for RE. Experimental results prove the interdependencies between the type's information of

entities and relations, and our model benefits from unifying these elements in the modeling process. The integration of type information improves the performance of all sub-tasks.

6. Conclusions

In this paper, we present an effective approach for joint entity and relation extraction. Our method is able to simultaneously and efficiently recognize boundaries and types of entities, as well as the relations among them. By utilizing our novel word-pair tagging method, we overcome the spatial and semantic limitations of previous methods, thereby effectively generating more accurate triplets through the fusion of structural information. Our experiments demonstrate that our method is competitive with the previous state-of-the-art results on three standard benchmarks and consistently delivers significant enhancements over the runner-up models in a majority of the evaluated scenarios. We illustrate the feasibility of integrating entity and relation type information within the pre-trained language model, which enriches the final contextual representation of the model. Simultaneously, the extraction model will be relieved of insufficient interaction of two tasks. In future work, we plan to further study the effect of fusion representation in our framework and expand the model framework to support a wider array of information extraction tasks.

Author Contributions: All authors contributed to the study's conception and design. Z.Z.: Conceptualization; Methodology; Writing—Review & editing. H.Z.: conceptualization, methodology, writing—reviewing and editing. L.S.: data curation, writing—original draft preparation. Y.Y.: visualization, investigation. S.X.: supervision, validation. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the China National Petroleum Corporation (CNPC) Science and Technology Program (Project No. 2021DQ06). This work was supported by The Basic Science (Natural Science) Research Projects of Universities in Jiangsu Province under Grant Agreement (Grant No. 22KJB110009). This work was supported by Xinjiang Uygur Autonomous Region Natural Science Foundation (Project No. 2022D01F67).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all individual participants included in the study.

Data Availability Statement: The datasets that support the findings of this study are available from SciERC <http://nlp.cs.washington.edu/sciIE/> (accessed on 11 July 2024), ADE <https://github.com/lavis-nlp/spert/tree/master/scripts> (accessed on 11 July 2024), and CoNLL04 <https://github.com/lavis-nlp/spert/tree/master/scripts> (accessed on 11 July 2024).

Conflicts of Interest: The authors declare that they have no conflicts of interests.

Abbreviations

The following abbreviations are used in this manuscript:

MDPI	Multidisciplinary Digital Publishing Institute
DOAJ	Directory of open access journals
TLA	Three-letter acronym
LD	Linear dichroism

References

1. 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, 2011, HLT'11, Portland, ON, USA, 19–24 June 2011; pp. 551–560.
2. Gormley, M.R.; Yu, M.; Dredze, M. Improved relation extraction with feature-rich compositional embedding models. *arXiv* **2015**, arXiv:1505.02419. [[CrossRef](#)]
3. 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, Online, 6–11 June 2021; pp. 50–61. [[CrossRef](#)]

4. Ye, D.; Lin, Y.; Li, P.; Sun, M. Packed Levitated Marker for Entity and Relation Extraction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Dublin, Ireland, 22–27 May 2022; pp. 4904–4917. [[CrossRef](#)]
5. 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), Baltimore, MD, USA, 22–27 June 2014; pp. 402–412. [[CrossRef](#)]
6. Wang, S.; Zhang, Y.; Che, W.; Liu, T. Joint extraction of entities and relations based on a novel graph scheme. In Proceedings of the Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 13–19 July 2018; pp. 1227–1236.
7. Verga, P.; Strubell, E.; McCallum, A. Simultaneously self-attending to all mentions for full-abstract biological relation extraction. *arXiv* **2018**, arXiv:1802.10569. [[CrossRef](#)]
8. Wang, Y.; Yu, B.; Zhang, Y.; Liu, T.; Sun, L. TPLinker: Single-stage Joint Extraction of Entities and Relations Through Token Pair Linking. *arXiv* **2020**, arXiv:2010.13415.
9. 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 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), Virtual, 1–6 August 2021. [[CrossRef](#)]
10. Sui, D.; Zeng, X.; Chen, Y.; Liu, K.; Zhao, J. Joint entity and relation extraction with set prediction networks. In *IEEE Transactions on Neural Networks and Learning Systems*; IEEE: New York, NY, USA, 2023; pp. 1–12. [[CrossRef](#)]
11. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *arXiv* **2018**, arXiv:1810.04805.
12. Wu, S.; He, Y. Enriching pre-trained language model with entity information for relation classification. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, Beijing, China, 3–7 November 2019; pp. 2361–2364. [[CrossRef](#)]
13. Eberts, M.; Ulges, A. Span-Based Joint Entity and Relation Extraction with Transformer Pre-Training. *arXiv* **2020**, arXiv:1909.07755.
14. Wang, Y.; Sun, C.; Wu, Y.; Zhou, H.; Yan, J. UniRE: A Unified Label Space for Entity Relation 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), Virtual, 1–6 August 2021; pp. 220–231. [[CrossRef](#)]
15. Shang, Y.M.; Huang, H.; Mao, X.L. OneRel: Joint Entity and Relation Extraction with One Module in One Step. *Proc. Aaai Conf. Artif. Intell.* **2022**, *36*, 11285–11293. [[CrossRef](#)]
16. Tang, W.; Xu, B.; Zhao, Y.; Mao, Z.; Liu, Y.; Liao, Y.; Xie, H. UniRel: Unified Representation and Interaction for Joint Relational Triple Extraction. *arXiv* **2022**, arXiv:2211.09039.
17. Liang, S.; Wei, W.; Mao, X.L.; Fu, Y.; Fang, R.; Chen, D. STAGE: Span Tagging and Greedy Inference Scheme for Aspect Sentiment Triplet Extraction. *Aaai Conf. Artif. Intell.* **2023**, *37*, 13174–13182. [[CrossRef](#)]
18. Ren, F.; Zhang, L.; Yin, S.; Zhao, X.; Liu, S.; Li, B.; Liu, Y. A novel global feature-oriented relational triple extraction model based on table filling. *arXiv* **2021**, arXiv:2109.06705.
19. Ma, Y.; Hiraoka, T.; Okazaki, N. Named entity recognition and relation extraction using enhanced table filling by contextualized representations. *J. Nat. Lang. Process.* **2022**, *29*, 187–223. [[CrossRef](#)]
20. Wan, Q.; Wei, L.; Zhao, S.; Liu, J. A Span-based Multi-Modal Attention Network for joint entity-relation extraction. *Knowl.-Based Syst.* **2023**, *262*, 110228. [[CrossRef](#)]
21. Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; Weld, D.S. Knowledge-based weak supervision for information extraction of overlapping relations. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, ON, USA, 19–24 June 2011; pp. 541–550.
22. 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), Melbourne, Australia, 15–20 July 2018; pp. 506–514. [[CrossRef](#)]
23. Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; Xu, B. Joint extraction of entities and relations based on a novel tagging scheme. *arXiv* **2017**, arXiv:1706.05075.
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. Dixit, K.; Al-Onaizan, Y. Span-level model for relation extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 5308–5314. [[CrossRef](#)]
26. Miwa, M.; Sasaki, Y. Modeling joint entity and relation extraction with table representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014; pp. 1858–1869. [[CrossRef](#)]
27. Tran, T.; Kavuluru, R. Neural metric learning for fast end-to-end relation extraction. *arXiv* **2019**, arXiv:1905.07458.
28. Ren, F.; Zhang, L.; Yin, S.; Zhao, X.; Liu, S.; Li, B.; Liu, Y. A Novel Global Feature-Oriented Relational Triple Extraction Model based on Table Filling. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, Virtual, 7–11 November 2021; pp. 2646–2656. [[CrossRef](#)]

29. Luan, Y.; He, L.; Ostendorf, M.; Hajishirzi, H. Multi-Task Identification of Entities, Relations, and Coreference for Scientific Knowledge Graph Construction. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, 31 October–4 November 2018. [\[CrossRef\]](#)
30. Gurulingappa, H.; Rajput, A.M.; Roberts, A.; Fluck, J.; Hofmann-Apitius, M.; Toldo, L. Development of a benchmark corpus to support the automatic extraction of drug-related adverse effects from medical case reports. *J. Biomed. Inform.* **2012**, *45*, 885–892. [\[CrossRef\]](#) [\[PubMed\]](#)
31. Roth, D.; Yih, W.t. A linear programming formulation for global inference in natural language tasks. In Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004, Boston, MA, USA, 6–7 May 2004; pp. 1–8.
32. 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, 26th International Conference on Computational Linguistics: Technical Papers, Osaka, Japan, 11–16 December 2016; pp. 2537–2547.
33. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
34. Wang, C.; Liu, X.; Chen, Z.; Hong, H.; Tang, J.; Song, D. DeepStruct: Pretraining of language models for structure prediction. *arXiv* **2022**, arXiv:2205.10475.
35. Crone, P. Deeper task-specificity improves joint entity and relation extraction. *arXiv* **2020**, arXiv:2002.06424.
36. Wadden, D.; Wennberg, U.; Luan, Y.; Hajishirzi, H. Entity, Relation, and Event Extraction with Contextualized Span Representations. *arXiv* **2019**, arXiv:1909.03546.
37. Shen, Y.; Ma, X.; Tang, Y.; Lu, W. A Trigger-Sense Memory Flow Framework for Joint Entity and Relation Extraction. In Proceedings of the Web Conference 2021, Ljubljana, Slovenia, 19–23 April 2021; pp. 1704–1715. [\[CrossRef\]](#)
38. Yan, Z.; Zhang, C.; Fu, J.; Zhang, Q.; Wei, Z. A partition filter network for joint entity and relation extraction. *arXiv* **2021**, arXiv:2108.12202.
39. Santosh, T.; Chakraborty, P.; Dutta, S.; Sanyal, D.K.; Das, P.P. Joint entity and relation extraction from scientific documents: Role of linguistic information and entity types. In Proceedings of the 2nd Workshop on Extraction and Evaluation of Knowledge Entities from Scientific Documents (EEKE 2021), Online, 27–30 September 2021.
40. Jeong, Y.; Kim, E. Scideberta: Learning deberta for science technology documents and fine-tuning information extraction tasks. *IEEE Access* **2022**, *10*, 60805–60813. [\[CrossRef\]](#)
41. Giannis, B.; Johannes, D.; Thomas, D.; Chris, D. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Syst. Appl.* **2018**, *114*, 34–45. [\[CrossRef\]](#)
42. Wang, J.; Lu, W. Two are Better than One: Joint Entity and Relation Extraction with Table-Sequence Encoders. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 1706–1721. [\[CrossRef\]](#)
43. Cabot, P.L.H.; Navigli, R. REBEL: Relation extraction by end-to-end language generation. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual, 16–20 November 2021; pp. 2370–2381. [\[CrossRef\]](#)
44. Zhao, S.; Hu, M.; Cai, Z.; Liu, F. Modeling dense cross-modal interactions for joint entity-relation extraction. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, Online, 7–15 January 2021; pp. 4032–4038.
45. Li, X.; Yin, F.; Sun, Z.; Li, X.; Yuan, A.; Chai, D.; Zhou, M.; Li, J. Entity-relation extraction as multi-turn question answering. *arXiv* **2019**, arXiv:1905.05529.
46. Li, J.; Fei, H.; Liu, J.; Wu, S.; Zhang, M.; Teng, C.; Ji, D.; Li, F. Unified Named Entity Recognition as Word-Word Relation Classification. *Proc. AAAI Conf. Artif. Intell.* **2022**, *36*, 10965–10973. [\[CrossRef\]](#)

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.