

## Article

# A Joint Extraction Model for Entity Relationships Based on Span and Cascaded Dual Decoding

Tao Liao, Haojie Sun \*  and Shunxiang Zhang

College of Computer Science and Engineering, Anhui University of Science and Technology,  
Huainan 232001, China

\* Correspondence: sun01175724@gmail.com

**Abstract:** The entity–relationship joint extraction model plays a significant role in entity relationship extraction. The existing entity–relationship joint extraction model cannot effectively identify entity–relationship triples in overlapping relationships. This paper proposes a new joint entity–relationship extraction model based on the span and a cascaded dual decoding. The model includes a Bidirectional Encoder Representations from Transformers (BERT) encoding layer, a relational decoding layer, and an entity decoding layer. The model first converts the text input into the BERT pretrained language model into word vectors. Then, it divides the word vectors based on the span to form a span sequence and decodes the relationship between the span sequence to obtain the relationship type in the span sequence. Finally, the entity decoding layer fuses the span sequences and the relationship type obtained by relation decoding and uses a bi-directional long short-term memory (Bi-LSTM) neural network to obtain the head entity and tail entity in the span sequence. Using the combination of span division and cascaded double decoding, the overlapping relations existing in the text can be effectively identified. Experiments show that compared with other baseline models, the F1 value of the model is effectively improved on the NYT dataset and WebNLG dataset.

**Keywords:** entity relation extraction; span; decode; cascade; neural network



**Citation:** Liao, T.; Sun, H.; Zhang, S. A Joint Extraction Model for Entity Relationships Based on Span and Cascaded Dual Decoding. *Entropy* **2023**, *25*, 1217. <https://doi.org/10.3390/e25081217>

Academic Editor: Wray Buntine

Received: 7 May 2023

Revised: 8 August 2023

Accepted: 12 August 2023

Published: 16 August 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/>).

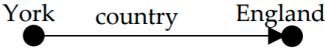
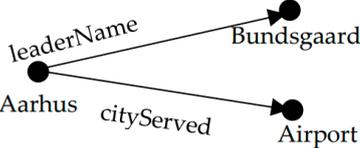
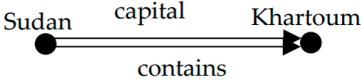
## 1. Introduction

Entity–relationship extraction means to extract entity relationships from unstructured text [1,2] and convert them into structured data by analyzing unstructured text. Entity–relationship extraction is very important for building knowledge graphs and question–answering systems [3], and information retrieval tasks [4,5] play a crucial role [6]. The entity–relationship triple is one of the basic representation methods of entity relationships. A triple in the form of <head entity, relation, tail entity> consists of two entities and the relationship between them, which represents the semantic relationship between entities in the text.

The two primary kinds of existing entity–relationship extraction methods are the pipeline extraction method and the joint model extraction method [7]. Among them, the pipeline extraction method divides the relationship extraction task into two independent subtasks, firstly, identifying the entities in the given text, and then identifying the relationship between entities [8–10]. The joint model extraction method recognizes the entities and the relationship between entities at the same time. Although the above two methods have achieved good results, the existing models still face problems such as entity nesting, overlapping relationships, and data noise. In the overlapping relationship problem, the overlapping relationship includes single entity overlap (SEO) and entity pair overlap (EPO) [11]. In this paper, this category is also used for the division, as shown in Table 1. In the sentence ‘The city of Aarhus, whose mayor is Jacob Bundsgaard, is served by Aarhus Airport’, there are two entity–relationship triples ‘< Aarhus, leaderName, Bundsgaard >’ and ‘< Aarhus airport, cityServed, Aarhus >’. There is a ‘leaderName’ relationship between

the entity ‘Aarhus’ and the entity ‘Bundsgaard’, at the same time, and there is a ‘cityServed’ relationship between the entity ‘Aarhus’ and the entity ‘Aarhus airport’. We call this type of overlapping relationship as single entity overlap. In the sentence ‘News of the list’s existence unnerved officials in Khartoum, Sudan ‘s capital’, there are two entity–relationship triples ‘< Sudan, capital, Khartoum > and < Sudan, contains, Khartoum >’. There are two kinds of relationships between entity ‘Sudan’ and entity ‘Khartoum’, which are the ‘capital’ relationship and the ‘contains’ relationship. We call this type of overlapping relationship as entity pair overlap.

**Table 1.** The examples of normal, single entity overlap, and entity pair overlap. Single entity overlap means that two or more entities have a relationship with a certain entity; entity pair overlap means that there is more than one relationship between two entity pairs.

Type	Example Sentences and Entity Relationships	Graphic Representation
Normal	York is located in the England. {<York, country, England>}	
Single entity overlap	The city of Aarhus, whose mayor is Jacob Bundsgaard, is served by Aarhus Airport. {<Aarhus, leaderName, Bundsgaard >, <Aarhus airport, cityServed, Aarhus>}	
Entity pair overlap	News of the list’s existence unnerved officials in Khartoum, Sudan’s capital. {<Sudan, capital, Khartoum >, <Sudan, contains, Khartoum>}	

In this paper, we propose a novel joint entity–relationship extraction model based on a span and cascaded dual-decoding method (SCDM) to solve the problem of overlapping relationship. First, we use the Bidirectional Encoder Representations from Transformers (BERT) pretrained language model to process the text, convert them into the tokens, and then use the random span division mechanism to divide the tokens to form span sequences. Second, we perform relation decoding on span sequences in combination with a predefined set of relation types to obtain the relation types that exist between span sequence entities. Third, integrating the relationship type obtained by the relationship decoding layer for entity decoding, in the entity decoding part, we first obtain the head entity and then obtain the tail entity. The head entity is decoded by fusing the relation type feature, and the tail entity is obtained by fusing the relation type feature and the head entity. Through the above steps, the model proposed can effectively solve the overlapping relationship, and use the dual-decoding mechanism, which can improve the training speed and reduce the error transmission.

Overall, the significant contributions of this paper are as follows:

- A novel joint entity–relationship extraction model, SCDM, based on the span and a cascaded dual decoding are proposed. Different from existing methods, this paper performs the span division after the BERT preprocessing to form span sequences, and subsequent decoding tasks are performed within the span sequences. The issue of overlapping relations in the text is successfully resolved.
- This method uses a cascaded decoding mechanism, which has the advantages of a fast learning speed and a reduced error transmission. First, it decodes the relationship between entities in the span sequences, and then it decodes the entities in the span sequences to obtain the head entity and the tail entity. The relationship type information obtained by the relation decoding layer is fused when decoding the head entity, and

the relationship type information and the acquired head entity information are fused when decoding the tail entity.

- Through the analysis of the NYT dataset and the WebNLG dataset, when compared to the current models, the methodology suggested in this research produced the best results.

## 2. Related Work

An essential task in the field of natural language processing (NLP) is extracting entity connections from unstructured text, which is also a necessary step in building a knowledge graph to provide support for downstream related tasks [12,13]. At present, the mainstream entity–relationship extraction methods can be roughly divided into two types: pipeline entity–relationship extraction methods and joint entity–relationship extraction methods.

### 2.1. The Pipeline Extraction Model

The relationship extraction task is divided into two separate subtasks using the pipeline extraction approach. Nayak T. et al. [14] introduced the use of the Bi-LSTM neural network model and attention mechanism to obtain long-distance dependent information, effectively solving the problem of facing sentences that are longer and have long distances between entities in the sentence. Zeng D. et al. [15] introduced the use of deep convolutional neural networks to extract vocabulary and sentence-level features in the text, which effectively solved the problem of classifying the relationship between entity pairs that required complex preprocessing. Guo X. et al. [16] introduced the use of a neural network combining recurrent neural network (RNN) and convolutional neural network (CNN) and adding an attention mechanism to complete the relationship classification, extract higher-level text information and obtain the feature information of sentences. Guo Z. et al. [17] introduced adding entity type and relationship alias information and inputting them into the graph convolutional neural network to improve the effect of the entity–relationship extraction.

### 2.2. The Joint Extraction Model

The pipeline method does not need to manually construct features and has high accuracy, so it is widely used, but because the error in the entity recognition task may be transmitted to the relationship classification task, it causes the problem of error propagation [1]. Therefore, in recent years, researchers have gradually focused on the research of joint entity–relationship extraction methods. Zeng X. et al. [11] introduced an end-to-end model based on the copy mechanism. That model used the copy mechanism to solve the problem of overlapping relationships and achieved excellent results. Copying text information is likely to cause information redundancy and relatively complex calculations. Luo L. et al. [18] introduced a tagging scheme for solving overlapping problems and proposed an Att-Bi-LSTM-CRF model to solve the problem of entity–relationship extraction in the biological domain. Ma Y. et al. [19] introduced a self-training entity–relationship extraction model in document-level text, which mainly solved the problems of high memory consumption and manual annotation in document-level entity–relationship extraction. Bhartiya A. et al. [20] first analyzed the problems existing in the original dataset and introduced a dataset, DiS-ReX, for the entity–relationship extraction, and then provided a benchmark. Hwang W. et al. [21] introduced an end-to-end extraction system applied to the legal field, which could bring convenience to lawyers' statistics and analysis of legal data. Xie Y. et al. [22] improved the performance of the entity–relationship extraction model by enhancing evidence in document-level entity–relationship extraction.

In recent years, Shang Y. et al. [23] introduced a method using one step and one module to extract entity and relations. They showed an annotation strategy and a decoding strategy to solve the overlapping relations. Ye D. et al. [24] introduced a new method to obtain the representation of the relationship between span sequences. The packaging strategy oriented by similar spans effectively solved the problem of overlapping relationships in

the text. Wei Z. et al. [25] introduced a cascaded binary tagging framework for entity–relationship extraction and analyzed the framework theoretically. It not only significantly improved the effect in the case of BERT pretraining but also significantly improved the extraction effect even without pretraining. Ma L. et al. [26] conducted research on the basis of Wei Z. et al. [25], which introduced the use of cascaded double-decoding methods for entity–relationship extraction. First, the relationships existing in the text were decoded, and then the head and tail entities in the text were decoded. Although this method achieved good results, it only partially solved the problem of overlapping relationships.

This paper is also aimed at the problem of overlapping relationships in the text. Different from Ma L. et al. [26], the method proposed in this paper uses random spans to form span sequences after the BERT encoding layer. A span partitioning of the tokens can divide entities that may have overlapping relationships into the same span sequence to solve the overlapping relationship problem. The length of the span sequence is effectively set to take advantage of the long-distance relationship between entities in the text.

### 3. Methodology

#### 3.1. Overview Network Architecture

This section describes the joint entity–relationship extraction model proposed in this paper based on the span and a cascaded decoding. Compared with the original joint model of entity–relationship extraction, if there are overlapping relationships in the text, the effect of the entity–relationship extraction will not be ideal. On the whole, the model is divided into three layers, a BERT decoding layer, a relationship decoding layer, and an entity decoding layer. The entity decoding layer includes two parts: the head entity extraction and the tail entity extraction. In this model, we first input the text into the BERT encoding layer, use the BERT-BASE-CASED pretrained language model to convert the text into vectors, and randomly divide the vectors to form span sequences. Then, we use linear layers to get the type of relationship between entities within the span sequences. Finally, the span sequences information is combined with the relationship type feature to extract the head entity and tail entity in the span sequence. The model is shown in Figure 1.

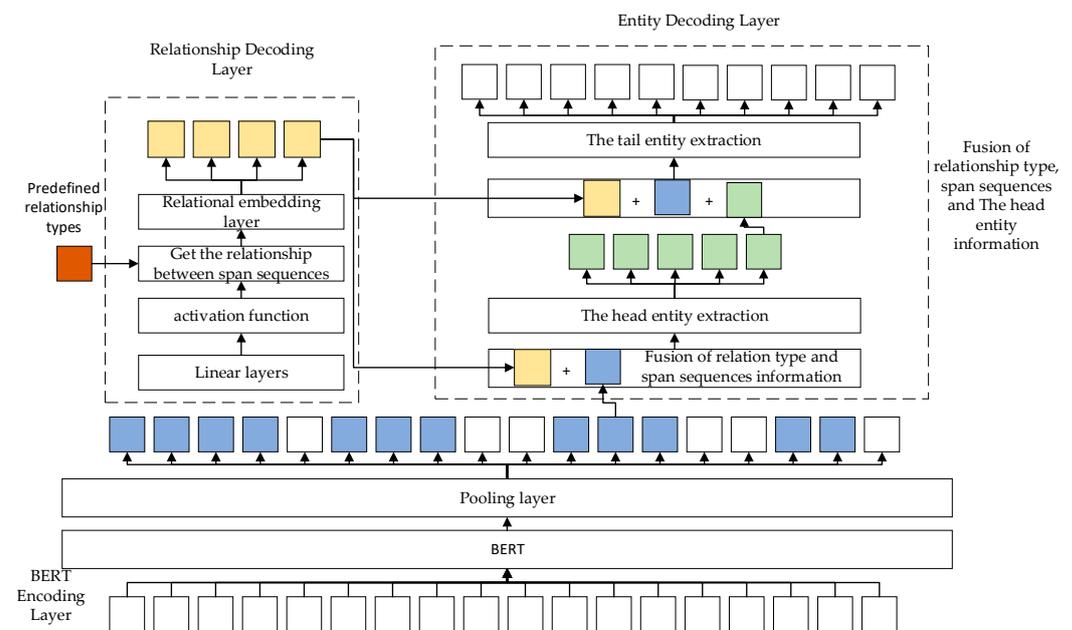


Figure 1. The framework of the model.

Where blue represents the span sequences, and the number of blue-colored blocks represents the size of the span sequence; yellow represents the relation type obtained by the relationship decoding layer; orange represents the predefined relation type, which contains

24 relation types in the NYT dataset, and the WebNLG dataset contains 246 relationship types; green indicates the head entity obtained by the entity decoding layer. In this model, we use the BERT encoder to generate tokens and divide them into span sequences; we use the relation decoder to get the relations in the span sequence; and we use the entity decoder to get the head and tail entities.

### 3.2. BERT Encoder Layer

We use the BERT pretrained language model to encode texts to capture semantic information between texts. The main task of the encoding layer is to encode the text information into matrix vectors. The encoding layer uses the BERT pretrained language model to obtain the semantic features of the sentence. The semantic features of the sentence are expressed as  $X = [x_1, x_2, x_3, x_4 \dots x_n]$ , which contains the prior knowledge obtained by BERT (BERT-BASE-CASED) in the pretraining stage.

Firstly, the input text sequence is represented as word vectors through the embedding layer, and the  $i$ -th token in the processed vectors is represented as formula (1):

$$e_i = W_{token}(t_i) \quad (1)$$

where  $W_{token}(t_i)$  is represented as a token embedding. Then, we input it into the BERT (BERT-BASE-CASED) pretraining model for encoding. The BERT pretraining model contains 12 hidden layers, and the size of each hidden layer is 768. The encoded result in the sentence is shown in formula (2):

$$H_b = BERT(E) \quad (2)$$

where  $E$  represents the sequence of tokens formed after token embedding, and the semantic representation  $H_b$  of the sentence is obtained from the text through the BERT model, and it is input into the pooling layer to generate the input of the relational decoding layer, as shown in formula (3):

$$H'_b = Pooling(H_b) \quad (3)$$

### 3.3. Relationship Decoding Layer

Since there are many kinds of relationships between entities in the span sequences, decoding the relationship type in the text is the main task of this layer. Determining the relationships that exist in span sequences is similar to a multilabel classification problem [26]. Span division is performed on the vectors generated by the BERT decoding layer to form a span sequence, as shown in formula (4):

$$S^i = (s_i, s_{i+1}, \dots, s_{i+n}) \quad (4)$$

where  $(i, i + 1, \dots, i + n)$  represents the size of the span sequence formed after the division. The relationship decoding layer decodes the relationship type between entity pairs within span sequences. Given a predefined set of relationship types, as shown in formula (5):

$$R^t = (r_1, r_2, \dots, r_n) \quad (5)$$

where  $n$  is the size of the relation type  $R^t$ , first, we input the span sequences into the linear layers to obtain the information in the span sequences, and then we use the sigmoid activation function to calculate the probability of the relationship type between entities, as shown in formula (6):

$$P^r = \varphi(W^r h + b) \quad (6)$$

We set the relationship filtering threshold. If the value of  $P^r$  is greater than the value of the relationship filtering threshold, it is considered to be a valid relationship type, otherwise, it is considered to be an invalid relationship type. We embed the relationship type in the

span sequences into the vector for the subsequent entity decoding layer, as shown in formula (7):

$$V^r = (v_1, v_2, \dots, v_n) \quad (7)$$

### 3.4. Entity Decoding Layer

#### 3.4.1. Head Entity Extraction

The main task of the head entity extraction is to extract head entities in relation types, which are obtained from the relation decoding layer. First, the span sequences and relation type information are fused; then, they are input into the Bi-LSTM neural network to obtain the information in the span sequences, where the output representation of the hidden layer of the Bi-LSTM neural network is shown in formula (8):

$$H^h = Bi - LSTM(S^i; V^r) \quad (8)$$

The probability of the head entity in the relation type to which the entity belongs in the span sequences is determined using the softmax activation function. As shown in formulas (9) and (10):

$$P_{head}^{start} = Softmax(W_h^s * X + b_h^s) \quad (9)$$

$$P_{head}^{tail} = Softmax(W_h^t * X + b_h^t) \quad (10)$$

where  $W_h^s$  represents the weight matrix for obtaining the starting position,  $W_h^t$  represents the weight matrix for obtaining the end position,  $X$  represents the matrix formed by fusing the final span sequences and relationship type, and  $b_h^s$  and  $b_h^t$  represent obtaining different position offsets.

#### 3.4.2. Tail Entity Extraction

The main task of the tail entity extraction is to extract tail entities in relation types. First, the span sequences, head entity information, and relationship type information are fused; then they are input into the Bi-LSTM neural network to obtain the information in the span sequences, where the output representation of the hidden layer of the Bi-LSTM neural network is shown in formula (11):

$$H^h = Bi - LSTM(S^i; V^r; E^h) \quad (11)$$

where  $E^h$  represents the head entity obtained by the head-entity extraction process. The probability of the tail entity in the relation type to which the entity belongs in the span sequences is determined using the softmax activation function, as shown in formulas (12) and (13):

$$P_{tail}^{start} = Softmax(W_t^s * X + b_t^s) \quad (12)$$

$$P_{tail}^{tail} = Softmax(W_t^t * X + b_t^t) \quad (13)$$

where  $W_t^s$  represents the weight matrix for obtaining the starting position,  $W_t^t$  represents the weight matrix for obtaining the end position,  $X$  represents the matrix formed by fusing the final span sequences and relationship type, and  $b_t^s$  and  $b_t^t$  represent obtaining different positions offset.

### 3.5. Loss Function

Relationship categorization and entity recognition are both multiclassification issues, so the cross-entropy loss function is used in the SCDM model. The relationship decoding

layer, the head entity extraction, and the tail entity extraction's loss functions are added to create the overall model's loss function.

$$L = c_1l^r + c_2l^h + c_3l^t \quad (14)$$

where  $l^r$  represents the loss function of the relational decoding layer,  $l^h$  represents the loss function of extracting the head entity in the entity decoding layer, and  $l^t$  represents the loss function of extracting the tail entity in the entity decoding layer.  $c_1$  represents the influence of the relational decoding layer in our model, and  $c_2$  and  $c_3$ , respectively, represent the influence of the head entity extraction and tail entity extraction in the entity decoding layer in our model. The loss values of the relational decoding layer and the entity decoding layer are averaged to calculate the total loss of the model.

## 4. Experiments

### 4.1. Datasets

We evaluated the method proposed in this paper on two datasets, the NYT dataset [26] and the WebNLG dataset [27,28]. The training set in the NYT dataset has a total of 56,214 sentences, and the test set has a total of 5000 data and a total of 24 relationship types. The training set in the WebNLG dataset has a total of 5017 sentences, and the test set has a total of 703 data and a total of 246 relationship types. The division of the dataset is shown in Table 2.

**Table 2.** Size of datasets.

Datasets	NYT		WebNLG	
	Training Set	Test Set	Training Set	Test Set
Normal	37,015	3264	1599	246
EPO	9781	979	224	26
SEO	14,737	1295	3407	457
Total	56,214	5000	5017	703
Relation type	24		246	

### 4.2. Experimental Environment

The model proposed in this paper used the PyTorch framework and NVIDIA GeForce RTX 3090 (24 G) graphic processing unit (GPU), and its hardware and software are shown in Table 3.

**Table 3.** Software and hardware environment.

Name	Environment
System	Windows
GPU	NVIDIA GeForce RTX 3090 (24 G)
Memory	56 G
Hard disk	2 T
Python version	Python 3.8
PyTorch version	1.8.0

### 4.3. Experimental Parameters

The experiment in this paper used the BERT-BASE-CASED pretrained language model. The word embeddings had a dimension of 300 and were randomly initialized. The location embedding dimension was 20. The training parameter settings are shown in Table 4.

**Table 4.** Training parameter settings.

Parameter	Value
BERT model	BERT-BASE-CASED
Learning rate	$2e^{-5}$
Batch size	16
Epochs	100
Dropout	0.4
Optim	Adam
Position embedding size	20
Max span	10

#### 4.4. Evaluation Methods

This paper used two methods to evaluate model performance: partial match and extract match. The partial match method [29] states that an extracted triple (h, r, t) is only considered right if its relation and the last letters in the names of the head entity and tail entity are accurate. A predicted triple is only considered right if its relation and the entire names of its head and tail entities are both correct, according to the exact match method [30,31]. Take note that only the final word of an entity name was annotated in both the training and test sets.

The evaluation indicators used in this experiment were precision, recall, and F1-score. The latter is the most important evaluation parameter in entity–relationship extraction.

#### 4.5. Comparative Experiments

A comparison between the model in this research and the model from more recent years was conducted. Here are the comparative models:

1. CopyRe [19] proposed a method based on the replication mechanism and used one combined decoder or several independent decoders as two alternative approaches to the decoding process.
2. GraphRel [32] proposed a method using graph neural networks (GCN) and a copy mechanism to extract entity relation. It used GCN to make better use of spatial information and improve the extraction effect of models.
3. CopyRL [33] proposed a sequence-to-sequence model based on a copy mechanism and took into account the order of relationships; the model was able to provide superior outcomes thanks to the use of reinforcement learning.
4. CopyMTL [34] proposed a multitask learning method based on the replication mechanism, which enhanced the robustness of the model and could predict multitoken entities.
5. WDec [30] proposed a relation strategy for representation and a decoding method based on a pointer network to realize entity relation extraction.
6. AttentionRE [31] developed a supervised multihead self-attention technique to learn the token-level correlation for each individual connection type.
7. CasRel [25] proposed a binary cascaded tagging framework to eliminate the issue of text's overlapping relations, and a related theoretical analysis was carried out.
8. DualDec [26] proposed a dual-decoding mechanism to extract entity–relationship triple, which first decoded the relationship in the text, and then decoded the entity pairs of a specific relationship.

The comparison between the model used in this paper and the baseline model on the NYT (N) dataset and the WebNLG (W) dataset yielded the findings shown in Table 5. In the table, we conducted comparative experiments on two datasets (NYT dataset and WebNLG dataset), taking the average of multiple experiments to obtain the data values. We strove to ensure the accuracy of the results. The data showed that the model proposed in this paper had improved accuracy, recall, and F1 score in the two datasets. In the N dataset, the F1 score of the partial match evaluation method was increased by 0.7%, and the F1 score of the exact match evaluation method was increased by 0.8%. In the W dataset, the F1 score of the partial match evaluation method was increased by 1%, and the F1 score of

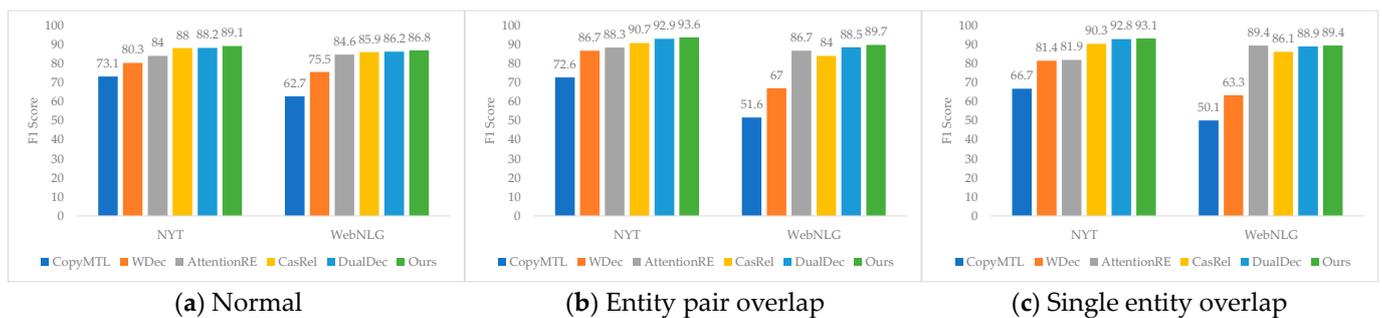
the exact match evaluation method was increased by 0.2%. This improvement was due to the use of the span to divide the text, so that the entity pairs that may have overlapping relationships were divided into the same span sequence, which effectively avoided the omission of the relationships between entity pairs, thereby improving the performance of the entity–relationship extraction.

**Table 5.** Compared with the baseline model. Experiments were carried out on the NYT (N) dataset and the WebNLG (W) dataset, respectively, on the partial match and exact match evaluation methods.

Methods	Partial Match						Exact Match					
	NYT(N)			WebNLG(W)			NYT(N)			WebNLG(W)		
	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1	Pre.	Rec.	F1
CopyRe	61.0	56.6	58.7	37.7	36.4	37.1	-	-	-	-	-	-
GraphRel	63.9	60.0	61.9	44.7	41.1	42.9	-	-	-	-	-	-
CopyRL	77.9	67.2	72.1	63.3	59.9	61.6	-	-	-	-	-	-
CopyMTL	-	-	-	-	-	-	75.7	68.7	72.0	58.0	54.9	56.4
WDec	-	-	-	-	-	-	88.1	76.1	81.7	88.6	51.3	65.0
AttentionRE	-	-	-	-	-	-	88.1	78.5	83.0	89.5	86.0	87.7
CasRel	89.7	89.5	89.6	93.4	90.1	91.8	89.1	89.4	89.2	87.7	85.0	86.3
DualDec	90.2	90.9	90.5	90.3	91.5	90.9	89.9	90.3	90.1	88.0	88.9	88.4
Ours	89.8	92.7	91.2	91.6	92.2	91.9	91.0	90.8	90.9	88.9	88.4	88.6

#### 4.6. Experimental Results on Different Relationships

To investigate how the model in this research affected various relationship types, we also conducted comparative experiments. Since there were three relationship types in the text, normal, entity pair overlap and single entity overlap, comparative experiments were conducted on different relationship types to test the precision, recall, and F1 score, respectively, as shown in Figure 2. An entity–relationship triple  $\langle h, r, t \rangle$  was only considered right if its relation and the last letters in the names of the head entity and tail entity were accurate.



**Figure 2.** Results for various phrase kinds based on how closely they overlap when using the exact match method.

Our model was compared with existing models in the N dataset and the W dataset, which were CopyMTL [34], WDec [30], AttentionRE [31], CasRel [25], and DualDec [26]. From Figure 2, it can be seen that the model in this paper achieved the best results.

As shown in Figure 2, in the normal relationship, the F1 score of the model proposed in this paper was 0.9% and 0.6% higher than that of the DualDec model on the NYT dataset and WebNLG dataset. In the EPO relationship, the proposed model's performance improved by 0.7% and 1.2% in the F1 score over the DualDec model on the NYT dataset and the WebNLG dataset, respectively. In the SEO relationship, the proposed model's performance improved by 0.3% and 0.5% in the F1 score over the DualDec model on the NYT dataset and the WebNLG dataset, respectively.

## 5. Conclusions

According to the analysis of existing entity–relationship extraction methods, this paper proposed a joint entity–relationship extraction model based on the span and a cascaded dual decoding. The model first divided the word vectors based on the span after the BERT pretraining model to form new span sequences and then decoded the relationship type between entity pairs in the span sequence. Finally, the span sequences information was combined with the relationship type feature, and the head entity and tail entity in the span sequences were extracted using the Bi-LSTM neural network. Experimental results showed that the model achieved certain results. The next step will be to study the span sequences formed after the division and to study the relationship between entities in the span sequence and the possible relationship between span sequences, so as to achieve better extraction results when the text contains complex overlapping relationships.

**Author Contributions:** Data curation, H.S.; methodology, H.S.; writing—original draft, H.S.; writing—review and editing, T.L. and S.Z.; visualization, H.S. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by National Natural Science Foundation of China (grant number 62076006); 2019 Anhui Provincial Natural Science Foundation Project (grant number 1908085MF189); University Synergy Innovation Program of Anhui Province (grant number GXXT-2021-008).

**Institutional Review Board Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## References

1. Zhong, Z.; Chen, D. A Frustratingly Easy Approach for Entity and Relation Extraction. In Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021. Association for Computational Linguistics (ACL), Online, 6–11 June 2021; pp. 50–61.
2. Sun, W.; Liu, S.; Liu, Y.; Kong, L.; Jian, Z. Information Extraction Network Based on Multi-Granularity Attention and Multi-Scale Self-Learning. *Sensors* **2023**, *23*, 4250. [[CrossRef](#)]
3. Liao, T.; Huang, R.; Zhang, S.; Duan, S.; Chen, Y.; Ma, W.; Chen, X. Nested named entity recognition based on dual stream feature complementation. *Entropy* **2022**, *24*, 1454. [[CrossRef](#)] [[PubMed](#)]
4. Zhang, Q.; Chen, M.; Liu, L. A Review on Entity Relation Extraction. In Proceedings of the 2017 Second International Conference on Mechanical, Control and Computer Engineering (ICMCCE), Harbin, China, 8–10 December 2017; pp. 178–183.
5. Tian, H.; Zhang, X.; Wang, Y.; Zeng, D. Multi-task learning and improved textrank for knowledge graph completion. *Entropy* **2022**, *24*, 1495. [[CrossRef](#)] [[PubMed](#)]
6. Liu, Z.; Li, H.; Wang, H.; Liao, Y.; Liu, X.; Wu, G. A novel pipelined end-to-end relation extraction framework with entity mentions and contextual semantic representation. *Expert Syst. Appl.* **2023**, *228*, 120435. [[CrossRef](#)]
7. Zhou, G.; Su, J.; Zhang, J.; Zhang, M. Exploring Various Knowledge in Relation Extraction. In Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05), Ann Arbor, MI, USA, 25–30 June 2005; pp. 427–434.
8. Mintz, M.; Bills, S.; Snow, R.; Jurafsky, D. Distant Supervision for Relation Extraction without Labeled Data. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP, Singapore, 2–7 August 2009; pp. 1003–1011.
9. Chan, Y.S.; Roth, D. Exploiting Syntactico-Semantic Structures for Relation Extraction. In Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Portland, OR, USA, 19–24 June 2011; pp. 551–560.
10. Gao, C.; Zhang, X.; Liu, H.; Yun, W.; Jiang, J. A joint extraction model of entities and relations based on relation decomposition. *Int. J. Mach. Learn. Cybern.* **2022**, *13*, 1833–1845. [[CrossRef](#)]
11. 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.
12. Li, Y.; Liu, C.; Du, N.; Fan, W.; Li, Q.; Gao, J.; Zhang, C.; Wu, H. Extracting medical knowledge from crowdsourced question answering website. *IEEE Trans. Big Data* **2016**, *6*, 309–321. [[CrossRef](#)]
13. Sennrich, R.; Haddow, B.; Birch, A. Neural Machine Translation of Rare Words with Sub-word Units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics (ACL), Berlin, Germany, 7–12 August 2016; pp. 1715–1725.

14. Nayak, T.; Ng, H.T. Effective Attention Modeling for Neural Relation Extraction. In Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL), Hong Kong, China, 3–4 December 2019; pp. 603–612.
15. Zeng, D.; Liu, K.; Lai, S.; Zhou, G.; Zhao, J. Relation Classification via Convolutional Deep Neural Network. In Proceedings of the COLING2014, the 25th International Conference on Computational Linguistics: Technical Papers, Dublin, Ireland, 23–29 August 2014; pp. 2335–2344.
16. Guo, X.; Zhang, H.; Yang, H.; Xu, L.; Ye, Z. A single attention-based combination of CNN and RNN for relation classification. *IEEE Access* **2019**, *7*, 12467–12475. [[CrossRef](#)]
17. Guo, Z.; Zhang, Y.; Lu, W. Attention Guided Graph Convolutional Networks for Relation Extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 241–251.
18. Luo, L.; Yang, Z.; Cao, M.; Wang, L.; Zhang, Y.; Lin, H. A neural network-based joint learning approach for biomedical entity and relation extraction from biomedical literature. *J. Biomed. Inform.* **2020**, *103*, 103384. [[CrossRef](#)] [[PubMed](#)]
19. Ma, Y.; Wang, A.; Okazaki, N. DREEAM: Guiding attention with evidence for improving document-level relation extraction. *arXiv* **2023**, arXiv:2302.08675.
20. Bhartiya, A.; Badola, K. Dis-Rex: A Multilingual Dataset for Distantly Supervised Relation Extraction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Dublin, Germany, 22–27 May 2022; pp. 849–863.
21. Hwang, W.; Eom, S.; Lee, H.; Park, H.J.; Seo, M. Data-efficient end-to-end information extraction for statistical legal analysis. In Proceedings of the Natural Legal Language Processing Workshop, Abu Dhabi, United Arab Emirates, 8 December 2022; pp. 143–152.
22. Xie, Y.; Shen, J.; Li, S.; Mao, Y.; Han, J. Eider: Empowering Document-Level Relation Extraction with Efficient Evidence Extraction and Inference-Stage Fusion. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, 22–27 May 2022; pp. 257–268.
23. Shang, Y.M.; Huang, H.; Mao, X. Onerel: Joint Entity and Relation Extraction with One Module in One Step. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 11285–11293.
24. 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, Germany, 22–27 May 2022; pp. 4904–4917.
25. Wei, Z.; Su, J.; Wang, Y.; Tian, Y.; Chang, Y. A Novel Cascade Binary Tagging Framework for Relational Triple Extraction. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 5–10 July 2020; pp. 1476–1488.
26. Ma, L.; Ren, H.; Zhang, X. Effective cascade dual-decoder model for joint entity and relation extraction. *arXiv* **2021**, arXiv:2106.14163.
27. Surdeanu, M.; Tibshirani, J.; Nallapati, R.; Manning, C.D. Multi-Instance Multi-Label Learning for Relation Extraction. In Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Jeju Island, Korea, 12–14 July 2012; pp. 455–465.
28. Riedel, S.; Yao, L.; McCallum, A. Modeling Relations and Their Mentions without Labeled Text. In Proceedings of the Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2010, Barcelona, Spain, 20–24 September 2010; pp. 148–163.
29. Gardent, C.; Shimorina, A.; Narayan, S.; Perez-Beltrachini, L. Creating Training Corpora for NLG Micro-Planning. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (ACL), Vancouver, BC, Canada, 30 July–4 August 2017.
30. Nayak, T.; Ng, H.T. Effective Modeling of Encoder-Decoder Architecture for Joint Entity and Relation Extraction. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 8528–8535.
31. Liu, J.; Chen, S.; Wang, B.; Zhang, J.; Li, N.; Xu, T. Attention as Relation: Learning Supervised Multi-Head Self-Attention for Relation Extraction. In Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, Online, 7–15 January 2021; pp. 3787–3793.
32. Fu, T.J.; Li, P.H.; Ma, W.Y. Graphrel: Modeling Text as Relational Graphs for Joint Entity and Relation Extraction. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 1409–1418.
33. Zeng, X.; He, S.; Zeng, D.; Liu, K.; Liu, S.; Zhao, J. Learning the Extraction Order of Multiple Relational Facts in A Sentence with Reinforcement Learning. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), Hong Kong, China, 3–7 November 2019; pp. 367–377.
34. Zeng, D.; Zhang, H.; Liu, Q. Copymtl: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning. In Proceedings of the AAAI Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020; Volume 34, pp. 9507–9514.

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