applied sciences

MDPI

Article

REACT: Relation Extraction Method Based on Entity Attention Network and Cascade Binary Tagging Framework

Lingqi Kong and Shengquau Liu *

School of Information Science and Engineering, Xinjiang University, Urumqi 830017, China; 107552101311@stu.xju.edu.cn
* Correspondence: liu@xju.edu.cn

Abstract: With the development of the Internet, vast amounts of text information are being generated constantly. Methods for extracting the valuable parts from this information have become an important research field. Relation extraction aims to identify entities and the relations between them from text, helping computers better understand textual information. Currently, the field of relation extraction faces various challenges, particularly in addressing the relation overlapping problem. The main difficulties are as follows: (1) Traditional methods of relation extraction have limitations and lack the ability to handle the relation overlapping problem, requiring a redesign. (2) Relation extraction models are easily disturbed by noise from words with weak relevance to the relation extraction task, leading to difficulties in correctly identifying entities and their relations. In this paper, we propose the Relation extraction method based on the Entity Attention network and Cascade binary Tagging framework (REACT). We decompose the relation extraction task into two subtasks: head entity identification and tail entity and relation identification. REACT first identifies the head entity and then identifies all possible tail entities that can be paired with the head entity, as well as all possible relations. With this architecture, the model can handle the relation overlapping problem. In order to reduce the interference of words in the text that are not related to the head entity or relation extraction task and improve the accuracy of identifying the tail entities and relations, we designed an entity attention network. To demonstrate the effectiveness of REACT, we construct a high-quality Chinese dataset and conduct a large number of experiments on this dataset. The experimental results fully confirm the effectiveness of REACT, showing its significant advantages in handling the relation overlapping problem compared to current other methods.

Keywords: relation extraction; relation overlapping problem; attention mechanism; gate mechanism

Citation: Kong, L.; Liu, S. REACT: Relation Extraction Method Based on Entity Attention Network and Cascade Binary Tagging Framework. *Appl. Sci.* **2024**, *14*, 2981. https://doi.org/10.3390/app14072981

Academic Editors: Enrique Costa-Montenegro, Francisco De Arriba-Pérez and Silvia García-Méndez

Received: 23 February 2024
Revised: 28 March 2024
Accepted: 29 March 2024
Published: 2 April 2024

1. Introduction

Relation extraction, an important research direction [1,2] in the field of information extraction [3], aims to automatically extract entities and their relations from massive text data, providing support for downstream tasks such as intelligent recommendation, semantic search, and deep question answering [4,5]. Relation extraction is usually divided into two subtasks: entity identification [6,7] and relation identification [8,9]. The main goal of entity identification is to identify entities with specific meanings from text, such as names of people, places, dates, organizations, and so on. Entity identification technology has a wide range of applications in the field of natural language processing, including the tasks of question-and-answer systems, public opinion analysis, entity linking, and so on. The main goal of relation identification is to identify the relations between entities from text. By automatically extracting relations between entities from large amounts of textual data, relation identification can help build applications such as knowledge graphs, recommendation systems, and sentiment analysis.

With the increasing complexity of information, the relation overlapping problem has begun to emerge [10]. The relation overlapping problem refers to the situation where there are shared entities among the entity-relation triad in the text. According to the types of overlapping, the relation overlapping problem can be further divided into three subcategories: Normal, Single Entity Overlap (SEO), and Entity Pair Overlap (EPO), as illustrated in the examples in Figure 1.

	Texts	Triplets
Normal	The **US** president **Trump** will visit **Beijing**, **China**	(Trump, *President_of*, US) (China, *Contains*, Beijing)
SEO	The **US** president **Trump** was born in **New York City**.	(**Trump**, *President_of*, US) (**Trump**, *Born_in*, New York City)
EPO	Trump will visit **Beijing**, which is the capital of **China**.	(**China**, *Contains*, **Beijing**) (**China**, *Capital*, **Beijing**)

Figure 1. Examples of cases of Normal, Single Entity Overlap (SEO), and Entity Pair Overlap (EPO). The overlapping entities are marked in bold. The first example belongs to the Normal class, which has no overlapping entities. The second one, with triplets sharing one single head entity "Trump", belongs to the SEO class. The last one with triplets, with the overlapping entity pair "{China, Beijing}", belongs to the EPO class.

Most existing relation extraction methods cannot effectively deal with the relation overlapping problem for two main reasons: (1) The model design is flawed and does not consider the situation where one entity in the text may have relations with multiple entities (Zheng et al. [4]), or it does not consider the existence of multiple relations between a particular head and tail entity situation (Miwa et al. [11]). (2) When the relation overlapping problem arises, there are always numerous entities in the text, and most of these entities do not have relations with each other. The noise interference from these irrelevant entities may mislead the model, causing it to incorrectly identify these irrelevant entities as part of the tail entity or relation. Additionally, the model needs to accurately understand the context around the head entity in order to correctly identify the tail entity and relation. The presence of irrelevant entities increases the difficulty of understanding the context. As shown in Figure 2, the head entity "Zhang Fansheng" in the text is only able to form the entity-relation triad "{Zhang Fansheng, Founder, Jinghai Enterprises}" with the tail entity "Jinghai Enterprises", and there are no relations with other entities. Due to the interference of entities unrelated to "Zhang Fansheng", it is difficult for the model to identify the tail entities related to the head entity "Zhang Fansheng" and the relations between them.

Text: In 1994, Zhang Fansheng established Jinghai Enterprises in Shanghai, which has been in operation for 25 years. As a renowned local enterprise, Jinghai Enterprises comprises two major subsidiaries: Jinghai Entertainment and Jinghai Football, both of which generate substantial profits

Figure 2. "Zhang Fansheng" is only able to form an entity-relation triad with "Jinghai Enterprises" and has no relations with any other entities.

In order to solve the above problems, we have designed the Entity Attention network, and we propose the Relation extraction method based on the Entity Attention network and Cascade binary Tagging framework (REACT). To address problem (1), we divide the relation extraction task into two subtasks: head entity identification and tail entity and relation identification. REACT first identifies the head entities present in the text and then identifies the tail entities that can be paired with the head entities and all possible relations between them. With this architecture, REACT is able to handle the relation overlapping problem. Please note that in this paper, entity pairs specifically refer to combinations of head and tail entities that have at least one type of relation. To address problem (2), we introduce the Entity Attention network. The Entity Attention network includes two parts: an Entity Attention Mechanism and an Entity Gated Mechanism. Words in the text have different degrees of importance for different head entities, and by introducing head entity information into the Entity Attention Mechanism, REACT is able to assign attention weights to words according to the head entities and reduce the word noise accordingly. After reducing the word noise according to the head entity, it is also necessary to consider the relevance of the words to the relation extraction task. The Entity Gated Mechanism calculates the degree of association of the words with the relation extraction task and diminishes the noise from words noise less associated with the task. By reducing the word noise through the above operations, REACT is able to focus on words with higher relevance to the head entity and the relation extraction task, thus increasing the accuracy of identifying the tail entity and relation and improving the performance of relation extraction.

Our main contributions are as follows:

1. Due to the current lack of Chinese datasets, we constructed a high-quality Chinese dataset with a high number of data with relation overlapping problems by optimizing the public Duie 2.0 entity-relation dataset;
2. For the relation overlapping problem, we propose the Relation extraction method based on the Entity Attention network and Cascade binary Tagging framework;
3. We conducted extensive experiments on a high-quality Chinese dataset to evaluate REACT and compared it with other baselines. The results demonstrate that REACT outperforms other baselines in handling relation overlapping problems.

2. Related Works

Early works on relation extraction adopted a pipeline approach (Zelenko et al. [8], Zhou et al. [12], Chan et al. [13], Mintz et al. [14], and Gormley et al. [15]). It first identifies all the entities in a sentence and then performs relation classification for each entity pair. This approach often faces the problem of error propagation because errors in the early stages cannot be corrected in later stages. To address this problem, subsequent works proposed joint learning of entities and relations, which includes feature-based models (Yu et al. [16]; Li et al. [17]; and Ren et al. [18]), as well as more recent neural network models (Gupta et al. [19]; Katiyar et al. [20]; Zhou et al. [21]; and Fu et al. [22]). By replacing manually constructed features with learned representations, neural network-based models have achieved considerable success in the relation extraction task.

As research has progressed, the relation overlapping problem has been increasingly emphasized. The relation overlapping problem refers to the situation where there are shared entities among the entity-relation triad in the text. To address this problem, many neural network-based joint models have been proposed [23].

Zeng et al. [24] were among the first to consider the relation overlapping problem. They first divided the relation overlapping problem into Normal, SEO, and EPO, and proposed a sequence-to-sequence (Seq2Seq) model with a copying mechanism. To address the relation overlapping problem, they allowed entities to participate freely in multiple triples. Building upon the Seq2Seq model, Zeng et al. [25] further investigated the impact of triple extraction order on relation extraction performance, transforming the relation extraction task into a reinforcement learning process, resulting in significant improvements.

Yuan et al. [26] redesigned the relation extraction method and successfully enabled the model to handle the relation overlapping problem. They first identified all entity pairs in the text and then individually determined whether each relation existed, rather than extracting only the most probable relation. Additionally, they pointed out that previous relation extraction methods ignored the connections between relations and independently predicted each possible relation. For example, if an entity pair has the "liveIn" relation, the "dieIn" relation is almost impossible to establish. Therefore, when determining the existence of a certain relation between entities, it is necessary to not only consider the target relation but also calculate the probability scores with respect to other relations. Inspired by the aforementioned research, Yuan et al. [27] assumed that the importance of words in text varies across different relations. They constructed a joint model based on a specific attention mechanism, which first identifies the relation existing in the text, then incorporates the relation into the attention mechanism, and finally identifies entity pairs containing the relation based on attention scores.

Although the above research was able to address the relation overlapping problem, they still regarded the identification of head and tail entities as independent processes, ignoring the semantic and logical connections between them. For example, organizational entities usually have relations with human entities (leaders, founders, members, etc.), but do not always have relations with entities such as songs, animals, movies, etc. Yu et al. [28] re-designed the method to first identify the head entity, then identify the potential tail entities based on the head entity, and finally determine all possible relations that may exist between the head entity and the tail entities. This approach not only addressed the relation overlapping problem but also achieved outstanding performance by utilizing head entity information in the tail entity identification process. Building upon this research, Li et al. [29] proposed an HBT framework, where they regarded the relation extraction task as a multi-turn question-answering task, incorporating external knowledge to introduce entities and relations. However, generating appropriate questions remains a challenge and is not suitable for most complex scenarios [30]. Inspired by the work of Yuan et al. [26] and Yu et al. [28], Fu et al. [22] introduced graphical convolutional networks, which are widely used in the field of logical reasoning [31], into the task of relation extraction. After identifying the relation between entity pairs, graph convolutional networks are used to infer the possibility of the existence of other relations. For example, "{Trump, LiveIn, United States}" can be inferred from "{Trump, LiveIn, Florida}". Prior to this, Zheng et al. [4] proposed a strong neural end-to-end joint model based on an LSTM sequence tagger for entities and relations, which helped infer unidentified relations based on identified relations. However, they were unable to address the relation overlapping problem.

Although current research has achieved joint extraction of entities and relations and developed models to address the relation overlapping problem, relations are still treated as discrete labels for entity pairs. This makes it difficult for models to correctly extract overlapping triples. Wei et al. [32] proposed a new cascaded binary labeling framework (CasRel), which uses BERT [33] as the feature extraction method and maps the tail entities through the head entities conditioned on the relations. As shown in the equation, r stands for relation, S stands for head entity, and O stands for tail entity. Relations are modeled as functions that map the head entity to the tail entity:

$$F_r(S) \to O \tag{1}$$

In natural language, the relations between entities are often closely related to their context. By introducing information about the head entity, the model can better understand the semantic structure of the sentence, thus predicting the tail entity and relations more accurately. Additionally, entities in a sentence may have multiple roles or meanings, but when paired with a specific head entity, their roles or meanings become clearer, reducing ambiguity in the tail entity and relations and improving the accuracy of identifying them. Therefore, CasRel has achieved good results. However, Wei et al. [32] were still unable to effectively utilize head entity information. Specifically, they only used head entity information as a parametric feature when identifying tail entities and relations, ignoring the potential role of head entity information in reducing word noise. When the problem of overlapping relations occurs, especially in cases of multiple overlapping relations, numerous entities typically emerge in the text. However, most of these entities do not participate in forming effective relations. This situation leads to the generation of a large number of potential invalid entity pairs, making it difficult for the model to accurately identify truly effective entity pairs among numerous possibilities, ultimately affecting the overall accuracy of relation extraction tasks.

In order to utilize the head entity information to reduce text noise and minimize interference from other irrelevant entities, we propose the Relation extraction method based on the Entity Attention network and Cascade binary Tagging framework (REACT). We utilize Entity Attention networks to help the model focus on words that are highly relevant to the head entity and the relation extraction task, thereby reducing the likelihood of identifying errors in tail entities and relations and improving the accuracy of extracting entity relation triplets.

3. Methodology

In this section, we propose the Relation extraction method based on the Entity Attention network and Cascade binary Tagging framework (REACT). A diagram outlining REACT is shown in Figure 3, and its detailed structure is shown in Figure 4. REACT consists of four main parts: the Encoding layer, the Head Entity Identification layer, the Entity Attention network layer (feature fusion layer), and the Tail Entity and Relation Identification layer. Our main contribution is in the Entity Attention network layer. In the following sections, each of these four parts is introduced.

3.1. Formalization of the Task

The input is a text $X = \{x_1, x_2, \ldots, x_n\}$ with n words. The desired output is an entity-relation triad, such as $T(X) = \{(h, r, t) | h, t \in E, r \in R\}$, where E and R are the set of entities and the set of relations, respectively. For example, in Figure 5, given a text containing 35 words, the goal is to extract the entity-relation triad "{United States, Contains, Los Angeles}", where "United States" is h, "Los Angeles" is t, and "Contains" is r in the entity-relation triad.

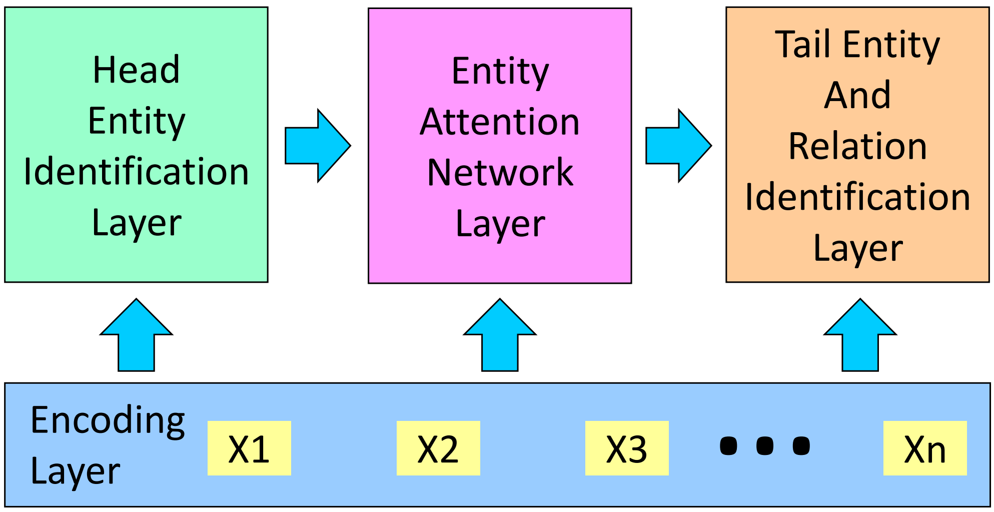

Figure 3. Diagram outlining REACT.

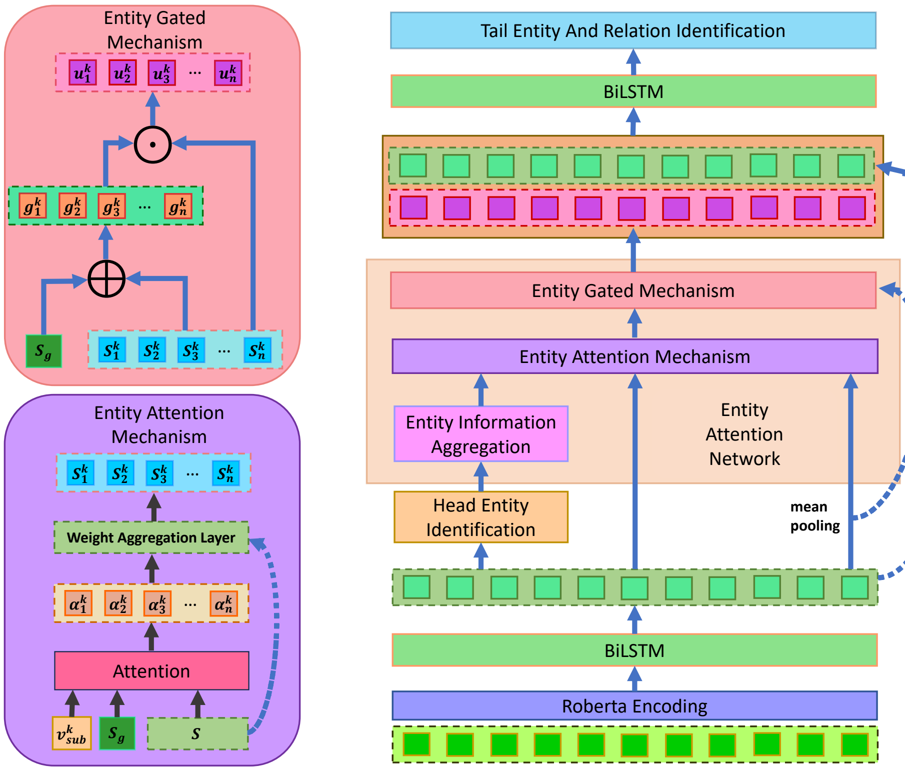

Figure 4. Detailed structure of REACT.

```
Los Angeles, the second largest city in the United States, is also
the largest city in the western United States and is often referred
to as the "City of Angels".
```

Figure 5. Example of a relation extraction task. Blue part represents the head entity, and the green part represents the tail entity.

3.2. Encoding Layer

3.2.1. Roberta Layer

In order to convert text into information that can be understood by computers and capture the semantic features of the text to improve the model's understanding of the text, we use *Roberta* [34] to encode the input text. Roberta is an improved version of Bert [34], and it is currently the mainstream method for encoding text features [35].

Here, we briefly review *Roberta*, a multi-layer bidirectional Transformer-based language representation model. It is designed to learn deep representations by jointly conditioning on both the left and right contexts of each word. Specifically, it is composed of a stack of N identical Transformer blocks. We denote the Transformer block as $Trans(h)$, in which h represents the input vector. The detailed operations are as follows:

$$h_0 = DW_d + W_p \tag{2}$$

$$h_\alpha = Trans(h_{\alpha-1}), \alpha \in [1, N] \tag{3}$$

where D is the matrix of one-hot vectors of subword indices in the input sentence. W_d is the subword embedding matrix. W_p is the positional embedding matrix, where p represents the position index in the input sequence. h_α is the hidden state vector, i.e., the context representation of the input sentence at the α_{th} layer, and N is the number of Transformer blocks. For a more comprehensive description of the Transformer structure, we refer readers to Liu et al. [34] and Vaswani et al. [36].

Using *Roberta*, we obtain the text encoding $H = \{h_1, h_2, \ldots, h_n\}$ for the text X.

3.2.2. BiLSTM Layer

Due to the complexity of the relation overlapping problem, we desire to exploit the interaction information among the words in the textual data. *BiLSTM* consists of two independent *LSTM* networks: one responsible for the forward sequence and the other responsible for the backward sequence. During the forward pass, the first *LSTM* gradually processes the input data from the beginning to the end of the sequence. During the backward pass, the second *LSTM* processes the input data from the end of the sequence to the beginning. By combining the hidden states of these two *LSTM* networks, a comprehensive sequence representation can be obtained that considers the contextual interaction information throughout the entire sequence.

The detailed network architecture is shown in Figure 6, which mainly consists of an update gate and a forget gate [11].

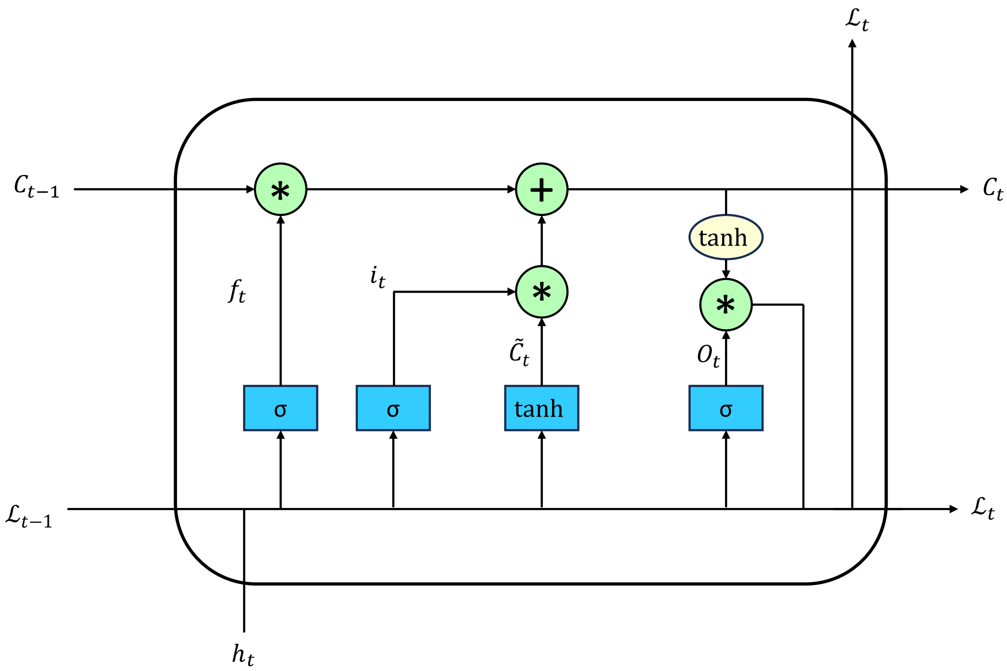

Figure 6. Detailed structure of LSTM. "*" denotes the dot product operation.

The update gate mainly controls the retention or deletion of some information from the forward state. The forget gate is used to control whether the calculation of the candidate state depends on the previous state.

The forget gate is calculated using the following formula:

$$f_t = \sigma(W_f \cdot [\mathcal{L}_{t-1}, h_t] + b_f) \tag{4}$$

The update gate is calculated using the following formula:

$$i_t = \sigma(W_i \cdot [\mathcal{L}_{t-1}, h_t] + b_i) \tag{5}$$

$$\widetilde{C}_t = tanh(W_c \cdot [\mathcal{L}_{t-1}, h_t] + b_c) \tag{6}$$

The output state is calculated using the following formula:

$$O_t = \sigma(W_o\ [\mathcal{L}_{t-1}, h_t] + b_o) \tag{7}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{8}$$

$$\mathcal{L}_t = O_t * tanh(C_t) \tag{9}$$

where h_t represents the current input data, $\mathcal{L}_{t-1}$ represents the input at the previous time, and $\mathcal{L}_t$ represents the current output. σ is the sigmoid function with a value ranging from 0 to 1; W_f, W_i, W_c, W_o, b_f, b_i, and b_c represent the weight matrix; and $tanh$ is the activation function.

Use text encoding H as the input for the $BiLSTM$, and obtain the output s_i.

$$s_i = [\overrightarrow{LSTM}(h_i); \overleftarrow{LSTM}(h_i)], i \in [1, n] \tag{10}$$

$S = \{s_1, s_2, \ldots, s_n\}$ denotes the contextual features.

3.3. Head Entity Identification

Since the problem of entity nesting may occur in the text, the decoder separately predicts all possible head entity start and end positions. In Figure 7, the start and end positions are predicted for the head entity "Stephen King". The specific operation is shown below:

$$P_i^{H-start} = W_{start} s_i + b_{start} \tag{11}$$

$$P_i^{H-end} = W_{end} s_i + b_{end} \tag{12}$$

where $P_i^{H-start}$ and P_i^{H-end} represent the probabilities that the i_{th} word in the text denotes the start and end positions of the head entity, respectively, which are set to 1 if they exceed a set threshold during the experiment, and 0 otherwise. W_{start} and W_{end} denote trainable weights.

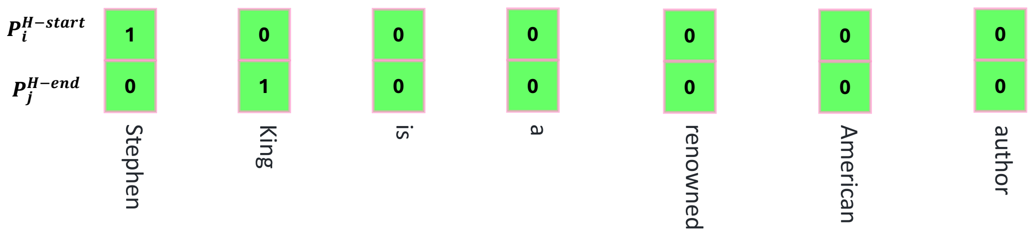

Figure 7. Head entity prediction example.

3.4. Entity Attention Network

3.4.1. Entity Information Aggregation

Assuming that the head entity k in the text has already been predicted, we cannot directly input the words comprising the head entity k into the Entity Attention Mechanism because the computer cannot understand them. Therefore, it is necessary to convert them into an appropriate form.

We take the average of the word feature s_i from the contextual features S that comprise the head entity k, obtaining the head entity feature v_{sub}^k. The specific calculation process is as follows:

$$v_{sub}^k = \sum_{i=start}^{end} \frac{s_i}{end - start + 1} \tag{13}$$

where s_i denotes the contextual features of the i_{th} word in the text and $start$ and end indicate the start and end positions of the head entity, respectively. For example, the head entity "United States" in Figure 5 has a start position of 10 and an end position of 11.

3.4.2. Entity Attention Mechanism

The main reason for the occurrence of the relation overlapping problem is the presence of a large number of entities in the text. Even worse, most of these entities do not have relations with each other, which poses a significant challenge for the model in extracting entity pairs.

Therefore, in order to allow the model to focus on words and entities related to the head entity and improve the accuracy of entity pair extraction, we have designed the Entity Attention Mechanism. By introducing the head entity information into the Entity Attention Mechanism, the model can allocate attention weights to words based on the head entity, reducing the interference of words unrelated to the head entity. Using the head entity k as an example, the specific calculation method is as follows:

$$S_g = avg\{s_1, s_2, \ldots, s_n\} \tag{14}$$

$$e_i^k = V^T tanh(W_v v_{sub}^k + W_g S_g + W_s s_i) \tag{15}$$

$$\alpha_i^k = \frac{\exp\left(e_i^k\right)}{\sum_{j=1}^n \exp\left(e_j^k\right)} \tag{16}$$

where V, W_v, W_g, and W_s are trainable weights; avg refers to the averaging operation applied to vectors in the context features S; S_g denotes the global features; $tanh$ represents the activation function; and $\sum_{j=1}^n \exp\left(e_j^k\right)$ denotes the summation of attention weights for each word. After calculating the attention weight e_i^k for each word based on the head entity k using Equation (15), we then use Equation (16) to calculate the proportion α_i^k of each word's attention weight relative to the total weight.

Using the above operation, the attention score α_i^k not only measures the importance of each word to the head entity but also quantifies the contribution of each word to the entire sentence.

After obtaining the attention scores, we need to reduce the features of words with low relevance to the head entity k. This helps reduce text noise. The specific calculation process is as follows:

$$s_i^k = \alpha_i^k s_i \tag{17}$$

$$S^k = (s_1^k, s_2^k, \ldots, s_n^k) \tag{18}$$

where s_i^k represents the word feature of the i_{th} word based on the head entity k.

3.4.3. Entity Gated Mechanism

The Entity Attention Mechanism is primarily used to attenuate the noise interference of words with weak associations to the head entity. However, in the text, there may exist some words that have strong associations with the head entity but are almost irrelevant to the relation extraction task. This is also a form of noise.

In order to attenuate the text word noise that is weakly associated with the relation extraction task, we propose the Entity Gated Mechanism. Still using the head entity k as an example, the operation is as follows:

$$g_i^k = \sigma((W_1 S_g + b_1) \oplus (W_2 s_i^k + b_2)) \tag{19}$$

$$u_i^k = g_i^k \odot tanh(W_3 s_i^k + b_3) \tag{20}$$

$$U^k = \{u_1^k, u_2^k, \ldots, u_n^k\} \tag{21}$$

where W_1, W_2, W_3, b_1, b_2, and b_3 are trainable weights, $\oplus$ is the cascade operation, $\odot$ is the dot-product operation, and σ denotes the shape activation function, which returns a value from 0 to 1.

The above formula compares S_g and s_i^k to calculate the degree of association g_i^k between the i_{th} word and the relation extraction task. Subsequently, based on the degree of association, the text noise is attenuated to obtain the associated feature u_i^k.

3.5. Tail Entity and Relation Identification

After deep noise attenuation, there may be a loss of fine-grained information [37]. We connect the noise-weakened U^k and the non-noise-weakened S to provide fine-grained features. After that, in order to learn the interaction information of the two granularity features, we feed them into *BiLSTM* to obtain the final representation O^k. The specific calculation is as follows:

$$h_i^k = u_i^k \oplus s_i \tag{22}$$

$$o_i^k = [\overrightarrow{LSTM}(h_i^k); \overleftarrow{LSTM}(h_i^k)], i \in [1, n] \tag{23}$$

$$O^k = \{o_1^k, o_2^k, \ldots, o_n^k\} \tag{24}$$

Finally, we extract the tail entities for each relation based on the head entity k. An example of *SEO* is given in Figure 8. For the head entity "Tim Robbins", we first identify the tail entity "United States" based on the relation "LiveIn" and then identify the tail entity "The Shawshank Redemption" based on the relation "leadActor". Through the above steps, two entity-relation triples "{Tim Robbins, LiveIn, United States}" and "{Tim Robbins, leadActor, The Shawshank Redemption}" are extracted from the text.

```
Tim Robbins currently lives in the United States and played the
leading role in "The Shawshank Redemption".
```

Figure 8. Specific case of identifying the tail entity and relation based on the head entity.

The specific calculation process is as follows:

$$p_i^{T-start} = W_r^{start} o_i^k + b_r^{start} \tag{25}$$

$$p_i^{T-end} = W_r^{end} o_i^k + b_r^{end} \tag{26}$$

$p_i^{T-start}$ and p_i^{T-end} represent the probabilities that the i_{th} word denotes the start and end positions of the tail entity, respectively. W_r^{start} and W_r^{end} denote the weight matrices for relation r. b_r^{start} and b_r^{end} denote the biases for relation r.

4. Experimental Section

4.1. Balanced Chinese Dataset Construction

DuIE2.0 [38] is the largest Chinese entity-relation extraction dataset, with 173,109 rows in the training set and 15,475 rows in the test set, including 49 entity-relation types. However, the number of occurrences of each relation in DuIE2.0 is uneven and varies greatly, making it unsuitable for relation extraction experiments [39]. At the same time, the two special scenarios of SEO and EPO also account for a small percentage, making it unsuitable for demonstrating the advantages of REACT. For this reason, We optimized DuIE2.0 by setting the threshold for the maximum number of occurrences of each relation in the training set to 230 and the threshold for each relation in the test set to 46. Table 1 shows some examples of sampling results where a few relation types fail to reach the threshold. To increase the number of data in the dataset containing EPO or SEO issues, we carefully collected over 1000 pieces of data, with each piece containing at least one type of overlapping problem. Figure 9 shows some of the data.

Table 1. Comparison between the optimized DuIE2.0 dataset and the original version.

Relation Number	DuIE2.0 Training Set	Uniform Sampling Training Set	DuIE2.0 Test Set	Uniform Sampling Test Set
1	984	230	101	46
2	3159	230	302	46
3	1807	230	170	46
4	7188	230	639	46
5	8345	230	718	46
6	593	230	46	46
7	937	230	87	46
8	1849	230	101	46
...	...	...	...	...
31	22	51	1	29
...	...	...	...	46
49	395	230	32	46
Total number	173,109	10,423	15,475	2012

	Texts	Triplets
Normal	Princess **Pingyang Zhao** was born during the late Sui Dynasty and early Tang Dynasty. Her **father** was none other than Emperor Gaozu of Tang, **Li Yuan**, and she was his third daughter.	(**Pingyang Zhao**, *Father*, Li Yuan)
SEO	According to reports from Japanese media citing Deadline, renowned Japanese horror film director **Hideo Nakata** will serve as the producer of the movie "**The Suicide Forest**." Nakata is best known for his direction of the classic horror film "**The Grudge**."	(**Hideo Nakata**, *Producer*, The Suicide Forest) (**Hideo Nakata**, *Director,* The Grudge)
EPO	**Jay Chou** finally returned to the stage of the Spring Festival Gala this year, performing a magic trick and singing the song "**Love Confession Balloon**" with magician Cai Weize.	(**Jay Chou**, *Singer*, **Love Confession Balloon**) (**Jay Chou**, *Composer*, **Love Confession Balloon**)

Figure 9. Data of Normal, SEO, and EPO classes in the high-quality Chinese dataset.

Finally, the high-quality Chinese dataset we constructed consists of 10,423 rows of training data, 2012 rows of test data, and 49 relation types. After optimization, the frequencies of various relations in the dataset were relatively uniform, covering multiple fields such as education, healthcare, finance, and law. The average length of all sentences is 56.28, with a total of 8043 different Chinese characters. Among the sentences in the dataset, 67% come from the Baidu Baike corpus, whereas 33% come from the Baidu News Push corpus. The distribution of different entity types is shown in Figure 10, with the most common types in the dataset being person, audiovisual work, song, and book. In the Chinese high-quality dataset we constructed, data with relation overlapping problems account for 32% of the total data.

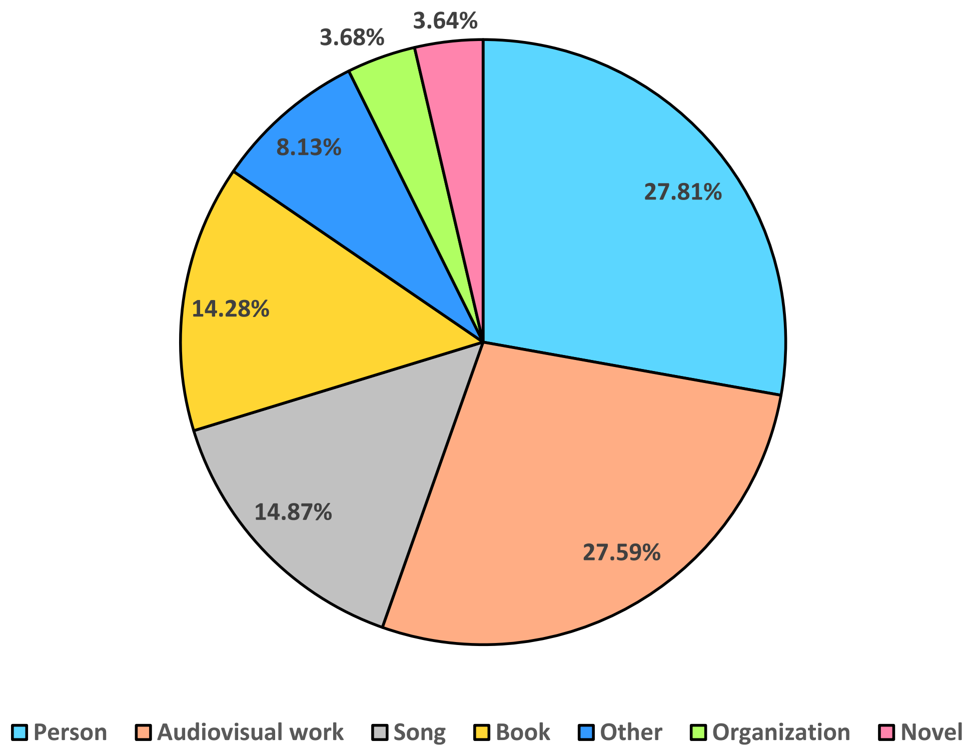

Figure 10. Proportions of different entity types in the high-quality Chinese dataset.

4.2. Baseline Comparison Experiment

To demonstrate the performance of REACT, we compared it with RSAN proposed by Yuan et al. [27], CasRel proposed by Wei et al. [32], and TPLinker proposed by Wang et al. [40] on the high-quality Chinese dataset we constructed. For the parameter settings of *Roberta*, we referred to the work of Cui et al. [41]. For Roberta, Cui et al. [41] conducted a large number of experiments in the field of Chinese natural language processing, achieving excellent results. Therefore, the parameter settings are relatively reliable, and the performance of the model can be guaranteed. As for *BiLSTM*, we conducted experiments for fine-tuning and selected the optimal parameters. The parameter settings for REACT are shown in Table 2.

Table 2. Settings for various parameters in REACT.

Module	Parameter	Value
Roberta	hidden_size	768
	max_position_embedding	512
	num_attention_heads	12
	num_hidden_layers	12
	pooler_fc_size	768
	pooler_num_attention_heads	12
	pooler_num_fc_layers	3
	pooler_size_per_head	128
	vocab_size	21,128
	input_size	768
BiLSTM	hidden_size	64
LSTM	hidden_size	64
	dropout	0.4
CNN	in_channels	768
	out_channels	128
	kernel_size	3

RSAN: Yuan et al. [27] hypothesized that words in a text have different degrees of importance under different relations. For this reason, they proposed a relation-based attention model, RSAN, which assigns different weights to contextual words under each relation. The reason for choosing this baseline for comparison is that the Relation Attention Mechanism in RSAN has some similarities with our proposed Entity Attention Mechanism.

CasRel: Wei et al. [32] adopted a new perspective instead of the previous classification mechanism, modeling the relation as a mapping from the head entity to the tail entity, and proposed a new cascading binary labeling framework (CaeRel). Excellent results were achieved due to the introduction of header entity information when identifying tail entities and relations. Since our model is an improvement on CasRel, a comparison with CasRel is necessary.

TPLinker: In order to alleviate the exposure bias and error propagation issues in CasRel, Wang et al. [40] proposed TPLinker. Joint extraction was described as a token-pair linking problem, and a new handshake labeling scheme was proposed to align entity pairs under each type of relation with boundary marking. This model ultimately achieved SOTA performance. Because our model is also an improvement on CasRel, it is necessary to compare REACT with TPLinker.

The experimental results for this section are shown in Table 3 and Figure 11. It can be seen that CasRel, TPLinker, and REACT outperformed RSAN in the relation extraction task, with an improvement of about 10% in the F1-score. REACT outperformed CasRel and TPLinker. Therefore, we can conclude that by utilizing head entity information to reduce word noise instead of treating head entity information as a parameter for identifying tail entities and relations, we can achieve better performance.

Table 3. The experimental data for REACT and the baselines on the high-quality Chinese dataset.

Model	Precision	Recall	F1-Score
RSAN	59.7	57.6	58.6
CasRel	65.7	64.2	64.9
TPLinker	65.3	66.4	65.8
REACT	68.5	66.0	67.2

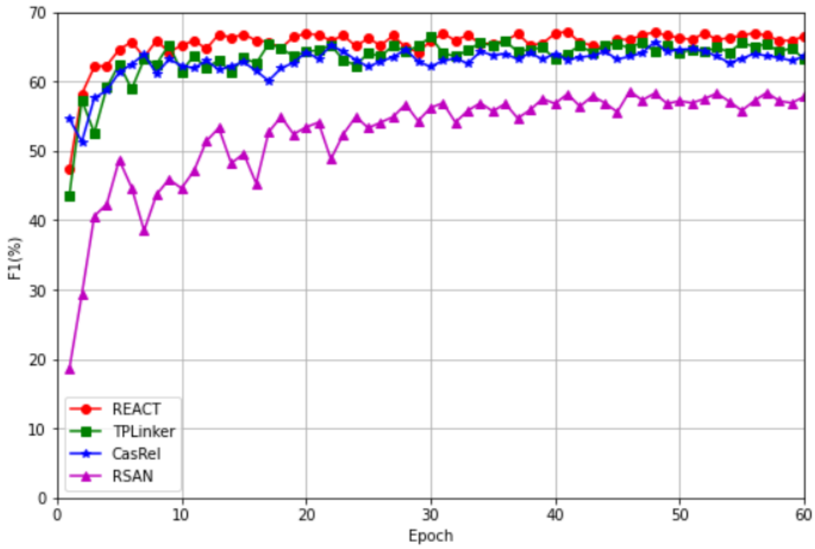

Figure 11. Comparison of convergence rates between REACT and the baselines.

4.3. Model Variants and Ablation Experiments

In order to comprehensively evaluate the performance of REACT and the effects of the proposed Entity Attention Mechanism (EAM) and Entity Gated Mechanism (EGM), we compared it with 14 variants, as shown in Table 4. We employed *Dictionary Encoding (DE)*, *BERT*, and *Roberta* as feature extraction methods, and replaced the *BiLSTM* in the Encoding layer with a *CNN* and *LSTM*. The parameter settings for the Dictionary Encoding in the experiments were based on the work of Liu et al. [42]. The parameter settings for *BERT* were based on the work of Wei et al. [32]. Liu et al. [42] conducted a large number of experiments in the field of Chinese relation extraction, and their proposed model achieved good results. The CasRel model proposed by Wei et al. [32] achieved state-of-the-art performance in 2020. Therefore, by referring to the above research to set model parameters, we can ensure the performance of the model. For *LSTM* and *CNN*, we conducted experiments and fine-tuned them to select the optimal parameters, as shown in Table 2.

Table 4. The fourteen variant models used as baselines. DE stands for Dictionary Encoding, EAM stands for Entity Attention Mechanism, and EGM stands for Entity Gated Mechanism.

No.	Encoding Layer	Additional Modules
1	DE+CNN	EAM+EGM
2	DE+LSTM	EAM+EGM
3	DE+BiLSTM	EAM+EGM
4	BERT+CNN	EAM+EGM
5	BERT+LSTM	EAM+EGM
6	BERT+BiLSTM	NULL
7	BERT+BiLSTM	EAM+EGM
8	Roberta+CNN	EAM+EGM
9	Roberta+LSTM	NULL
10	Roberta+LSTM	EAM
11	Roberta+LSTM	EAM+EGM
12	Roberta+BiLSTM	NULL
13	Roberta+BiLSTM	EAM
14	Roberta+BiLSTM	EGM
15 (REACT)	Roberta+BiLSTM	EAM+EGM

As shown in Table 5, the best performance was achieved using the *BiLSTM*, and the worst performance was achieved using the *CNN*, regardless of which encoding method was used. The relation extraction task usually requires the analysis of the whole sentence and needs to be inferred in context. The *BiLSTM* was able to perform forward and backward processing along the sentence sequence, thus capturing the long-term dependencies and context in the sentence, helping to better understand the semantics of the sentence and the relations between sentence components. *CNNs* are mainly used to process fixed-size local features and have a relatively low ability to model long-distance dependencies in sentences. Furthermore, *CNNs* are usually locally aware and do not have an explicit memory mechanism, thus may not be able to handle long-term dependencies and complex semantics in sentences.

Based on the overall results, compared to using *Dictionary Encoding* as the feature extraction method, using *BERT* and *Roberta* as the encoder was more effective, with their F1-scores both above 63%. Both *BERT* and *Roberta* are based on pre-trained language models that learn rich linguistic representations from large amounts of textual data through large-scale unsupervised learning. *BERT* and *Roberta* utilize the mechanism of parameter sharing to encode the individual words of an entire sentence at the same time. In contrast, *Dictionary Encoding* usually only extracts features based on independent words and lacks modeling of the sentence as a whole. In addition, *BERT* and *Roberta* employ the Transformer, which is capable of bidirectional context modeling to better capture the semantics and context of words in a sentence or text. This synthesis of contextual information is essential for understanding and expressing the meaning of a text. *Dictionary Encoding* usually performs feature extraction based on independent words or phrases and cannot capture the contextual information between words.

Table 5. Comparison of experimental results between REACT and the 14 model variants.

No.	Models	Precision	Recall	F1-Score
1	DE+CNN+EAM+EGM	70.7	40.5	51.5
2	DE+LSTM+EAM+EGM	74.3	44.7	55.8
3	DE+BiLSTM+EAM+EGM	74.6	46.7	57.4
4	BERT+CNN+EAM+EGM	67.5	62.0	64.6
5	BERT+LSTM+EAM+EGM	68.3	63.7	65.9
6	BERT+BiLSTM	64.8	63.8	64.1
7	BERT+BiLSTM+EAM+EGM	69.0	64.3	66.5
8	Roberta+CNN+EAM+EGM	69.2	60.1	64.3
9	Roberta+LSTM	65.7	62.2	63.9
10	Roberta+LSTM+EAM	66.4	65.1	65.7
11	Roberta+LSTM+EAM+EGM	68.1	64.8	66.4
12	Roberta+BiLSTM	66.2	63.9	65.0
13	Roberta+BiLSTM+EAM	67.6	65.3	66.4
14	Roberta+BiLSTM+EGM	67.8	64.7	66.2
15	(REACT) Roberta+BiLSTM+EAM+EGM	68.5	66.0	67.2

As can be seen from the 9th, 10th, 12th, and 13th data in Table 5, when the Entity Attention Mechanism was used, there was an increase of more than 1.4% in the F1-score, demonstrating that our designed Entity Attention Mechanism indeed enhanced model performance. The main purpose of an attention mechanism [36] is to enable the model to assign different attention weights to different words when processing input data in order to better understand and process the data. Therefore, the Entity Attention Mechanism can assist the model in focusing on words strongly associated with the head entity, which are typically crucial components in forming a correct entity-relation triad. This enables the model to encounter fewer disturbances from irrelevant words during prediction, thereby reducing the probability of misidentifying the tail entity or relation. Precision is the ratio of correctly identified entity-relation triads to the total number of identified entity-relation triads. Through the Entity Attention Mechanism, the model reduced instances of incorrectly pairing unrelated or erroneous tail entities with the head entity, thereby reducing false positives and directly enhancing precision. Consequently, compared to recall, the enhancement in precision was more pronounced.

From the 12th and 14th data in Table 5, we can observe that the F1-score improved by 1.2% when the Entity Gated Mechanism was used. A gated mechanism [43] can learn which information is more important for the current task and selectively retain or discard input information. The improvements in the F1-score demonstrate that our designed Entity Gated Mechanism indeed enhanced model performance.

As can be seen from the 6th, 7th, 9th, 11th, 12th, and 15th data in Table 5, when the Entity Attention network was used, the F1-scores improved by an average of about 2.5%. Therefore, the Entity Attention Mechanism and the Entity Gated Mechanism can be used together to improve the performance of the model.

4.4. Detailed Results on Different Types of Sentences

To further validate the capability of REACT in extracting overlapping relational triples, we conducted two extended experiments on different types of sentences and compared the performance of REACT with that of the baselines.

The detailed experimental results on three different datasets are shown in Figure 12. It can be seen that REACT achieved the best results in all three cases, proving that it has some advantages in addressing the relation overlapping problem.

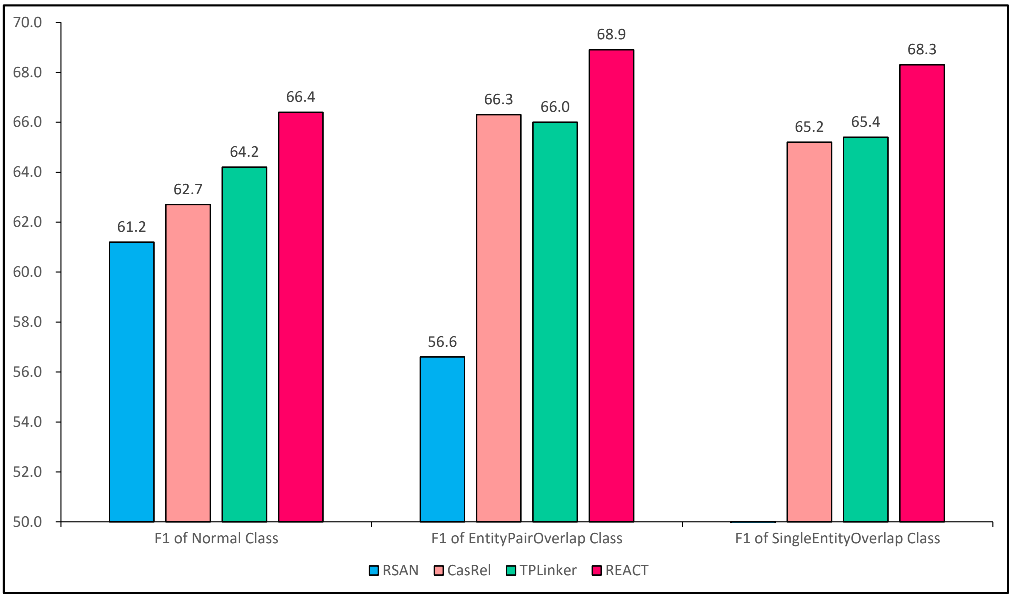

Figure 12. F1-score for extracting relational triples from sentences with different overlapping patterns.

We validated REACT's ability to extract relational triples from data containing different numbers of relational triples. Sentences were categorized into four classes, and the results are shown in Table 6. Again, REACT achieved excellent performance across all four classes. Although it is not surprising that the performance of most baselines decreased as the number of relational triples in a sentence increased, there are still some details that can be observed in the variation of the models' performance. Compared to these baselines, which are dedicated to addressing the relation overlapping problem, REACT exhibited the least degradation and achieved the best performance when confronted with complex situations.

Table 6. F1-score for extracting relational triples from sentences with different numbers (denoted as N) of triples.

Model	N = 1	N = 2	N = 3	N $\geq$ 4
RSAN	64.5	61.3	58.7	51.2
CasRel	62.9	65.0	68.6	63.4
TPLinker	65.7	67.6	68.4	61.5
RANGE	66.0	68.3	71.4	65.8

5. Conclusions

Regarding the problem of relation overlapping, we analyzed the current mainstream methods and proposed the Relation extraction method based on the Entity Attention network and Cascade binary Tagging framework (REACT). To demonstrate the effectiveness of REACT, we constructed a high-quality Chinese dataset. Experiments on this dataset showed that compared to the baselines, REACT achieved higher F1-scores, even when faced with complex situations, maintaining a certain advantage. In future work, we plan to integrate REACT with recently popular large language models. After identifying the head entity, we will use the large model to provide additional prior knowledge about the head entity. By introducing prior knowledge, the model can better understand the specific meaning and characteristics of the head entity, allocate attention weights more accurately, and improve the performance of identifying tail entities and relations.

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

Funding: This research was funded by the Major Science and Technology Projects in Xinjiang Uygur Autonomous Region (No. 2022A02012-1); National Natural Science Foundation of China (61966034).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are available on request from the first author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Golshan, P.N.; Dashti, H.R.; Azizi, S.; Safari, L. A study of recent contributions on information extraction. *arXiv* **2018**, arXiv:1803.05667.
2. Freitag, D. Machine learning for information extraction in informal domains. *Mach. Learn.* **2000**, *39*, 169–202. [CrossRef]
3. Hahn, U.; Oleynik, M. Medical information extraction in the age of deep learning. *Yearb. Med. Inform.* **2020**, *29*, 208–220. [CrossRef] [PubMed]
4. Zheng, S.; Hao, Y.; Lu, D.; Bao, H.; Xu, J.; Hao, H.; Xu, B. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing* **2017**, *257*, 59–66. [CrossRef]
5. Li, C.; Tian, Y. Downstream model design of pre-trained language model for relation extraction task. *arXiv* **2020**, arXiv:2004.03786.
6. Ratinov, L.; Roth, D. Design challenges and misconceptions in named entity recognition. In Proceedings of the CoNLL '09: Proceedings of the Thirteenth Conference on Computational Natural Language Learning, Boulder, CO, USA, 4–5 June 2009.
7. Sang, E.F.; De Meulder, F. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *arXiv* **2003**, arXiv:cs/0306050.
8. Zelenko, D.; Aone, C.; Richardella, A. Kernel methods for relation extraction. *J. Mach. Learn. Res.* **2003**, *3*, 1083–1106.
9. Bunescu, R.; Mooney, R. A shortest path dependency kernel for relation extraction. In Proceedings of the Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Vancouver, BC, Canada, 6–8 October 2005; pp. 724–731.
10. Zhou, Y.; Huang, L.; Guo, T.; Hu, S.; Han, J. An attention-based model for joint extraction of entities and relations with implicit entity features. In Proceedings of the Companion Proceedings of the 2019 World Wide Web Conference, Montreal, QC, Canada, 25–31 May 2019; pp. 729–737.
11. Miwa, M.; Bansal, M. End-to-end relation extraction using lstms on sequences and tree structures. *arXiv* **2016**, arXiv:1601.00770.
12. 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.
13. 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.
14. 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, Suntec, Singapore, 2–7 August 2009; pp. 1003–1011.
15. Gormley, M.R.; Yu, M.; Dredze, M. Improved relation extraction with feature-rich compositional embedding models. *arXiv* **2015**, arXiv:1505.02419.
16. Yu, X.; Lam, W. Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In Proceedings of the Coling 2010: Posters, Beijing, China, 23–27 August 2010; pp. 1399–1407.
17. Li, Q.; Ji, H. Incremental joint extraction of entity mentions and relations. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Baltimore, MD, USA, 23–25 June 2014; pp. 402–412.
18. Ren, X.; Wu, Z.; He, W.; Qu, M.; Voss, C.R.; Ji, H.; Abdelzaher, T.F.; Han, J. Cotype: Joint extraction of typed entities and relations with knowledge bases. In Proceedings of the 26th International Conference on World Wide Web, Perth, Australia, 3–7 April 2017; pp. 1015–1024.
19. 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, 11–17 December 2016; pp. 2537–2547.
20. 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 (Volume 1: Long Papers), Vancouver, BC, Canada, 30 July–4 August 2017; pp. 917–928.

21. Zhou, P.; Zheng, S.; Xu, J.; Qi, Z.; Bao, H.; Xu, B. Joint extraction of multiple relations and entities by using a hybrid neural network. In Proceedings of the Chinese Computational Linguistics and Natural Language Processing Based on Naturally Annotated Big Data: 16th China National Conference, CCL 2017, and 5th International Symposium, NLP-NABD 2017, Nanjing, China, 13–15 October 2017; Proceedings 16; Springer: Berlin/Heidelberg, Germany, 2017; pp. 135–146.
22. 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.
23. Bai, F.; Ritter, A. Structured minimally supervised learning for neural relation extraction. *arXiv* **2019**, arXiv:1904.00118.
24. 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.
25. 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.
26. Yuan, Y.; Liu, L.; Tang, S.; Zhang, Z.; Zhuang, Y.; Pu, S.; Wu, F.; Ren, X. Cross-relation cross-bag attention for distantly-supervised relation extraction. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 Januray–1 February 2019; Volume 33, pp. 419–426.
27. Yuan, Y.; Zhou, X.; Pan, S.; Zhu, Q.; Song, Z.; Guo, L. A Relation-Specific Attention Network for Joint Entity and Relation Extraction. In Proceedings of the IJCAI, Yokohama, Janpan, 11–17 July 2020; Volume 2020, pp. 4054–4060.
28. 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.
29. 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.
30. Li, X.; Feng, J.; Meng, Y.; Han, Q.; Wu, F.; Li, J. A unified MRC framework for named entity recognition. *arXiv* **2019**, arXiv:1910.11476.
31. Yao, L.; Mao, C.; Luo, Y. Graph convolutional networks for text classification. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; Volume 33, pp. 7370–7377.
32. Wei, Z.; Su, J.; Wang, Y.; Tian, Y.; Chang, Y. A novel cascade binary tagging framework for relational triple extraction. *arXiv* **2019**, arXiv:1909.03227.
33. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:1810.04805.
34. Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; Stoyanov, V. Roberta: A robustly optimized bert pretraining approach. *arXiv* **2019**, arXiv:1907.11692.
35. Zhong, P.; Wang, D.; Miao, C. Knowledge-enriched transformer for emotion detection in textual conversations. *arXiv* **2019**, arXiv:1909.10681.
36. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **2017**, *30*.
37. Chen, N.; Liu, F.; You, C.; Zhou, P.; Zou, Y. Adaptive bi-directional attention: Exploring multi-granularity representations for machine reading comprehension. In Proceedings of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 7833–7837.
38. Li, S.; He, W.; Shi, Y.; Jiang, W.; Liang, H.; Jiang, Y.; Zhang, Y.; Lyu, Y.; Zhu, Y. Duie: A large-scale chinese dataset for information extraction. In Proceedings of the Natural Language Processing and Chinese Computing: 8th CCF International Conference, NLPCC 2019, Dunhuang, China, 9–14 October 2019; Proceedings, Part II 8; Springer: Berlin/Heidelberg, Germany, 2019; pp. 791–800.
39. Feng, J.; Huang, M.; Zhao, L.; Yang, Y.; Zhu, X. Reinforcement learning for relation classification from noisy data. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
40. Wang, Y.; Yu, B.; Zhang, Y.; Liu, T.; Zhu, H.; Sun, L. TPLinker: Single-stage joint extraction of entities and relations through token pair linking. *arXiv* **2020**, arXiv:2010.13415.
41. Cui, Y.; Che, W.; Liu, T.; Qin, B.; Wang, S.; Hu, G. Revisiting pre-trained models for Chinese natural language processing. *arXiv* **2020**, arXiv:2004.13922.
42. Liu, Y.; Wen, F.; Zong, T.; Li, T. Research on joint extraction method of entity and relation triples based on hierarchical cascade labeling. *IEEE Access* **2022**, *11*, 9789–9798. [CrossRef]
43. Dey, R.; Salem, F.M. Gate-variants of gated recurrent unit (GRU) neural networks. In Proceedings of the 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, 6–9 August 2017; pp. 1597–1600.