Applied Soft Computing 119 (2022) 108604

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Joint entity and relation extraction with position-aware attention and relation embedding

Tiantian Chen, Lianke Zhou *, Nianbin Wang, Xirui Chen

College of Computer Science and Technology, Harbin Engineering University, Harbin, 150001, China

ARTICLE INFO

Article history:
Received 22 September 2021
Received in revised form 8 January 2022
Accepted 4 February 2022
Available online 16 February 2022

Keywords:
Entity recognition
Relation extraction
Relation embedding
Attention mechanism
Gate mechanism

ABSTRACT

The joint extraction of entities and relations is an important task in natural language processing, which aims to obtain all relational triples in plain text. However, few existing methods excel in solving the overlapping triple problem. Moreover, most methods ignore the position and order of the words in the entity in the entity extraction process, which affects the performance of triples extraction. To solve these problems, a joint extraction model with position-aware attention and relation embedding is proposed, named PARE-Joint. The proposed model first recognizes the subjects, and then uses the subject and relation guided attention network to learn the enhanced sentence representation and determine the corresponding objects. In this way, the interaction between entities and relations is captured, and the overlapping triple problem can be better resolved. In addition, taking into account the important role of word order in the entity for triple extraction, the position-aware attention mechanism is used to extract the subjects and the objects in the sentences, respectively. The experimental results demonstrate that our model can solve the overlapping triple problem more effectively and outperform other baselines on four public datasets.

1. Introduction

Extracting entities and relations from plain texts is the basic task of information extraction in natural language processing (NLP) [1]. These entity pairs connected by semantic relations form relational triples, generally in the form of (subject, relation, object). Extracting these triples can promote a series of downstream tasks, such as knowledge graph construction [2,3], machine translation [4], and question answering system [5].

Traditional pipelined methods [6–8] regard named entity recognition [9] and relation extraction [10] as two separate tasks. Although separate tasks can be easier and more flexible to train each model, the correlation between both tasks is ignored. In addition, because the relation extraction predicts relations after extracting entities, this may lead to error propagation or accumulation [11]. Therefore, more and more joint learning models about triples have been presented. Early joint learning models [12–14] need artificially well-designed features or rely heavily on NLP tools, which may lead to error propagation and accumulation [15]. Later studies mainly focused on learning the deep learning-based models [15–33], and have obtained satisfactory results in joint extraction tasks.

However, most existing models cannot effectively handle the scenarios where the sentence includes multiple entity pairs that overlap each other. These scenarios are first proposed by Zeng et al. [25]. As shown in Table 1, they divide the sentences into three classes according to the overlapping degree of entity pair, including Normal, EntityPairOverlap (EPO) and SingleEntityOverlap (SEO). The sentence is classified into the Normal pattern if none of its relational triplets have the same entity. The sentence is classified into SEO if some of its triplets share one entity. For example, the triplet (*Jackie R. Brown*, *Birth_place*, *Washington*) shares the entity "*Washington*" with (*Washington*, *Capital_of*, *United States of America*). The sentence is classified into EPO if some of its triplets share the same entity pairs as subjects and objects. For instance, the triplet (*Quentin Tarantino*, *Act_in*, *Django Unchained*) and (*Quentin Tarantino*, *Direct_movie*, *Django Unchained)* have the same subject "*Quentin Tarantino*" and object "*Django Unchained*". To be able to identify different overlapping patterns in sentences, Zeng et al. [25] present an end-to-end neural network model based on sequence-to-sequence (Seq2Seq) learning with a copy mechanism to extract triples. Considering the influence of the recognition order for the detecting triples, Zeng et al. [26] apply reinforcement learning to the Seq2Seq model. Later, Takanobu et al. [28] first predicts the relations and then extracts entities via reinforcement learning. Fu et al. [29] use graph convolutional networks (GCN) to identify triples. Nayak and Ng [32] propose two methods utilizing encoder–decoder architecture, which introduce a representation scheme and a

* Corresponding author.
E-mail address: zhoulianke@hrbeu.edu.cn (L. Zhou).

https://doi.org/10.1016/j.asoc.2022.108604
1568-4946/

Table 1
Examples of different overlapping patterns.

	Sentences	Triplets
Normal	The [United States] President [Trump] has a meet with [Tim Cook], the CEO of [Apple Inc].	(United States, Country_president, Trump) (Tim Cook, Company_CEO, Apple Inc)
EPO	[Quentin Tarantino] played a nobody in his directed film [Django Unchained].	(Quentin Tarantino, Act_in, Django Unchained) (Quentin Tarantino, Direct_movie, Django Unchained)
SPO	[Jackie R. Brown] was born in [Washington], the capital city of [United States of America].	(Jackie R. Brown, Birth_place, Washington) (Jackie R. Brown, Birth_place, United States of America) (Washington, Capital_of, United States of America)

pointer-network-based decoding approach to improve the performance of triple extraction. Most of the above triple extraction methods first identify entities and then learn a relation classifier $f(h, t) \rightarrow r$ for relation prediction. Apart from those approaches, Wei et al. [33] introduce the cascade binary tagging framework (CasRel). CasRel learns a relation-specific tagger $f_r(h) \rightarrow t$, which can model relations as functions that map subjects to objects. Under this framework, relational triple extraction is decomposed into the following two processes: First, all subjects in the sentence are determined; then all relations and object entities related to the subject are identified. Moreover, they further decompose the subject/object extraction process into start position recognition and end position recognition of the subject/object. Start position recognition and end position recognition use binary classifiers to detect the start and end position of subjects/objects, respectively. Although CasRel has achieved satisfactory results on the relational triple extraction task, there are still some limitations in the triple extraction process: (1) CasRel regards the start position recognition and the end position recognition as two independent subtasks. However, the position and order of words in the sentence are very important features in the NLP task. Therefore, the start position recognition and the end position recognition are related, and the result of the start position recognition affects the subsequent end position recognition. For example, in the entity "*Tim Cook*" in Table 1, we assume that the beginning position of the entity has been identified as the index of "*Tim*", then the end position of this entity must be after the word "*Tim*" and not too far away from it. If the boundary of the entity is not restricted, it is easy to affect the performance of the entity and triple extraction. (2) In the object extraction process, CasRel uses the subject as input to learning an object extractor, and then outputs the objects and relations corresponding to the subject at the same time. In other words, CasRel learns the sentence representation related to the subject, and then determines the corresponding objects and relations based on this single sentence representation. Even if the sentence contains multiple objects and relations about this subject, CasRel still only uses this single sentence representation. Take the second sentence in Table 1 as an example. Assuming that "*Quentin Tarantino*" has been identified as the subject, CasRel needs to identify both object "*Django Unchained*" and relation "*Act_in*" or object "*Django Unchained*" and relation "*Direct_movie*". It is difficult for CasRel to use only one sentence representation related to the subject "*Quentin Tarantino*" to determine two overlapping triples, which is particularly easy to affect the performance of extracting overlapping triples. In addition, CasRel ignores the interaction and correlation between relations and entities. For example, we assume that "*Quentin Tarantino*" is the recognized subject, and the relation labels "*Act_in*" and "*Direct_movie*" are considered to be known information. Subjects and relations interact to learn sentence representations related to different subjects and relations. There are as many sentence representations as there are combinations of subjects and relations. Thus, in the above example, according to the subject "*Quentin Tarantino*" and relation "*Act_in*", as well as the subject "*Quentin Tarantino*" and relation "*Direct_movie*", we can obtain two different sentence representations in the same sentence. According to these sentence representations, two overlapping triples can be easily extracted. The extraction scheme of obtaining the object through the subject and the relation seems to be easier to solve the overlapping triple problem than CasRel, and can obtain the interaction between entities and relations.

To solve the above problems, we present a joint extraction model based on position-aware attention and relation embedding, named PARE-Joint. PARE-Joint includes two modules: a subject extractor and an object extractor. The former extractor identifies all possible subjects in the sentence. The latter extractor extracts objects corresponding to the identified subjects and the known relations. Inspired by CasRel learning relation-specific object taggers $f_r(h) \rightarrow t$, we further study an object extractor $f(h, r) \rightarrow t$. Unlike previous methods treating relations as discrete labels, our object extractor uses relations as known information and input. According to the given relations and the identified subjects, the object extractor determines the possible objects in the sentence corresponding to each relation and subject; or return no object, demonstrating that there is no triple. In this way, the proposed model naturally solves the overlapping triple problem. Like CasRel, in the process of recognizing the subject and the object, we apply the pointer labeling scheme to extract all entities, that is, to identify the start and end positions of the entities. The difference with CasRel, which treats the start position recognition and the end position recognition as two independent subtasks, is that our method uses a position-aware attention mechanism to enhance the association of the two subtasks. In addition, to make full use of the interaction and correlation between relations and entities, the object extractor uses a specific subject-relation attention mechanism to obtain sentence representations under different subjects and relations, and applies a specific subject-relation gate mechanism to reduce useless noisy features.

The major contributions of this work can be summarized as follows:

- We propose a joint extraction model with position-aware attention and relation embedding (PARE-Joint), which is decomposed into two interrelated sub-modules, namely the subject extraction module and the object extraction module. The former module extracts all possible subjects, the latter module identifies the objects based on known relations and the extracted subjects. Our model can extract overlapping triples more efficiently it extracts objects under different relations and subjects.
- We present a position-aware attention mechanism that enhances the influence of the position and order of words in entity recognition.

• Considering the interaction and correlation between relations and entities, a subject and relation guided attention network is proposed. The attention network uses a specific subject-relation attention mechanism to construct sentence representations under different subjects and relations and a specific subject-relation gate mechanism to remove useless noise features.

• Experiments on four public datasets indicate that our approaches outperform existing methods.

2. Related work

Early methods [6–8] extract relational triples in a pipelined manner. These methods identify entities and relations in two steps: first, it uses named entity recognition to detect all possible entities; then, it extracts their relations by relation extraction. However, this pipelined manner has two disadvantages: (1) Wrong results from entity recognition will affect the subsequent relation extraction performance; (2) The pipelined methods divide the relational triplet extraction task into two separate subtasks, ignoring interactions between entity recognition and relation extraction.

To solve these shortcomings, many methods for jointly extracting entities and relations have been proposed. Early joint extraction methods [12–14] rely on manually engineered features or additional NLP tools, which may lead to the propagation and accumulation of errors. Due to the successful application of deep neural networks in other fields, e.g. speech recognition [34] and image processing [35], some methods with neural networks have been introduced in joint extraction tasks [15–33]. Zheng et al. [18] use the underlying coding representation of a shared neural network for joint learning. Similar to the above approach, Miwa et al. [19] also apply parameter sharing for joint learning and use different neural network models to extract entities and relations, respectively. Geng et al. [20] propose an end-to-end joint extraction model based on rich semantics to concatenate the semantic representation information from different modules, so that the word vectors can represent more comprehensive and rich contextual information of both entities and relations. Zhao et al. [21] transform entity-relation extraction into multi-turn question answering tasks, and improve the existing joint extraction model through a diversity question answering mechanism. Although these joint extraction methods learn shared parameters, it regards entity recognition and relation prediction as two independent subtasks. Therefore, it is difficult to make full use of interactions between two subtasks. Different from those studies, Zheng et al. [22] introduce a unified tagging scheme, and transform the extracting triple task into the sequence tagging problem. Yu et al. [23] propose a sequence labeling framework to decompose the task of extracting relational triples into several sequence labeling problems. Ye et al. [24] present a novel model, contrastive triple extraction with a generative transformer. The model is the first to integrate sequence generation with contrastive learning for triple extraction. Although these models implement the joint extraction of triples, they still fail to extract overlapping relations.

For the overlapping triple problem, Zeng et al. [25] present an end-to-end joint extraction method with a copy mechanism to identify entities and extract relations. As an improvement, Zeng et al. [26] apply reinforcement learning to the joint extraction model to study the influence of the recognition order. In addition, Zeng et al. [27] find that the model proposed by Zeng et al. [25] has problems with incorrect entity copying and incomplete entity extracting, then they present a multi-task learning framework to recognize complete entities. Fu et al. [29] present a joint extraction method using GCN to study the triple extraction. Zhao et al. [30] propose to construct heterogeneous graphs by treating relations as nodes on the graph and apply heterogeneous graph neural networks to obtain better representation for relation extraction tasks. Fei et al. [31] construct the entity graphs by enumerating possible candidate spans, then model the relational graphs between entities via a graph attention model. Nayak and Ng [32] propose a representation scheme and a pointer-network-based decoding approach to jointly extract. Nevertheless, these methods fail to extract the relations when the overlapping situation is relatively complex. Wei et al. [33] present a novel joint extraction framework named CasRel, which first detects subjects in the text, and then finds corresponding relations and objects for the predicted subject. Although CasRel has achieved reasonable performance, they ignore important information in the process of extracting relational triples, such as the position and order information between words in the entity, the interaction between entities and relations, and words under different relations and subjects should have different contributions to the semantic embedding of sentences.

In this paper, we propose a new method named PARE-Joint to obtain relational triples, which helps to deal with the overlapping problem. The model first extracts subjects and then extracts objects via the obtained subjects and known relations. Compared with the previous methods, which regard the relation labels as an independent and meaningless one-hot, we use the relation labels as the known information. We use the extracted subjects and relations to predict objects, which naturally solves the overlapping triple problem and fully learns interaction between the relations and entities. Our PARE-Joint treats the entity recognition task as a pointer labeling problem, and decomposes the entity recognition task into two subtasks, namely, the task of identifying the start and end positions of the entities. To further study the influence of start position recognition on end position recognition, we associate these two tasks by using position-aware attention.

Note that our work is inspired by CasRel, but the differences from CasRel are as follows: (1) CasRel regards the start position recognition and the end position recognition as two independent subtasks. We enhance the influence of the start position recognition on the end position recognition to limit the boundary of the entity. (2) CASREL learns relation-specific object taggers $f_r(h) \rightarrow t$, and regards the relations as general labels. We further study an object extractor $f(h, r) \rightarrow t$, that uses the relations as known information and considers the interaction between entities and relations.

3. Method

Fig. 1 shows the workflow of our model, including three steps:

Step 1: The shared sentence representation is obtained by BERT and passed to the corresponding downstream task layer.

Step 2: The subject extraction task converts each token vector in the shared sentence representation into the probability distribution of the corresponding start position tag and end position tag of the subject, respectively.

Step 3: The object extraction task converts the token vector into the probability distribution of the corresponding start position tag and end position tag of the object, respectively

We will describe the details of each part in the following subsections.

3.1. Task definition

3.1.1. Tagging scheme

For complicated information extraction tasks, decomposing the task into easier subtasks is very effective in solving the problem [23,36,37]. Thus, we adopt the decomposition strategy of the CasRel model. We decompose our model into two sub-modules: the subject extraction module and the object extraction module.

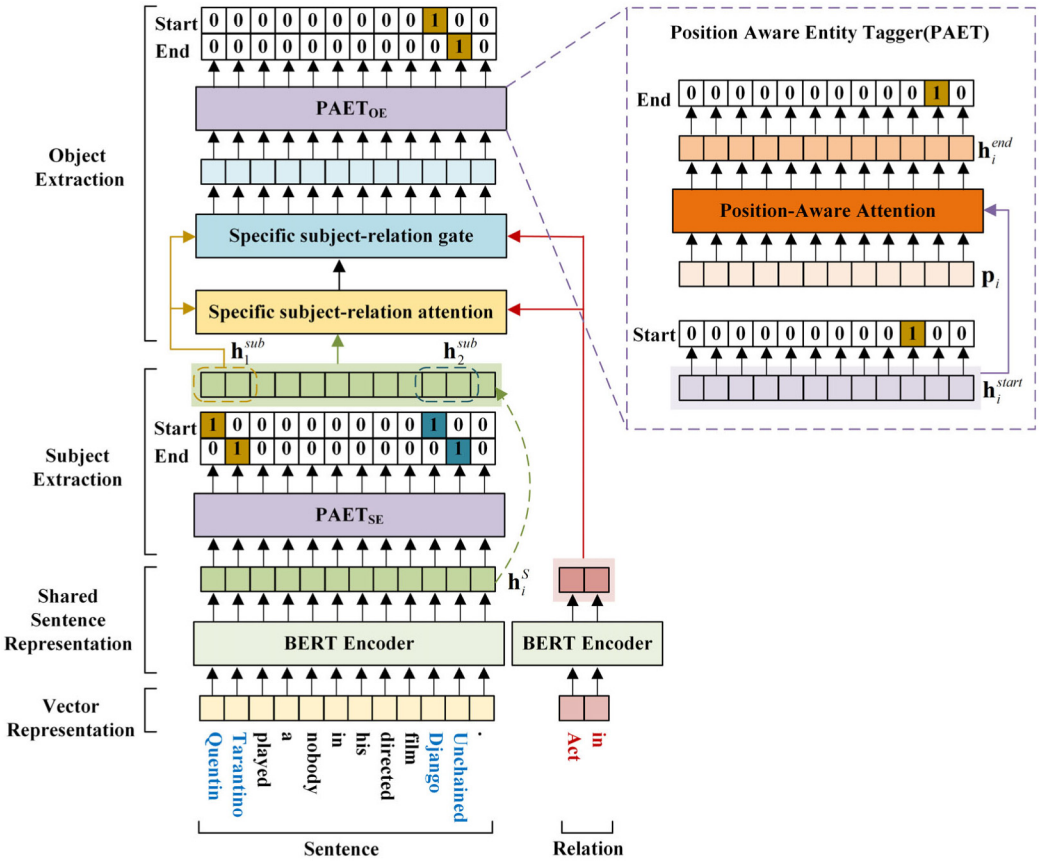

Fig. 1. Structures of the proposed PARE-Joint.

Two modules respectively identify the subjects and objects in the sentence. Both of these extractions can be regarded as entity extraction. In addition, to further decompose the task, the entity extraction task is converted into a pointer labeling problem. The extraction task is decomposed into two interrelated subtasks, namely, start position recognition and end position recognition. Specifically, start position recognition is designed to determine the start positions of entities. If a token is the start word of an entity, it will be assigned with a tag 1, otherwise, it will be tagged as 0. Compared with the start position recognition, end position recognition has a similar marking process and identifies the end position. Fig. 1 illustrates our entity tagging scheme. For example, when the entity "*Quentin Tarantino*" is labeled, the position corresponding to the word "*Quentin*" in its start tag sequence is tagged as 1, and the position corresponding to the word "*Tarantino*" in the end tag sequence is assigned with a tag 1. Therefore, the process of the subject/object extraction is to determine the boundary of the subject/object, that is, to determine the start position and end position of the subject/object. The whole process of triples extraction is to first confirm the boundary of the subject through the subject extractor, and then the object extractor takes the known relations label and the recognized subject as input and confirms the boundary of the object. For example, in Fig. 1, "*Quentin Tarantino*" is recognized as the subject, and its start position and end position are tagged as 1, respectively. The relation label "*Act_in*" is considered to be known information. Then, "*Quentin Tarantino*" and "*Act_in*" are used as the input of the object extractor. Finally, the corresponding tail entity "*Django Unchained*" is extracted, and its two boundary positions are tagged as 1

3.1.2. Task modeling

Joint Extraction of Entities and Relations aims to identify all possible triples (subject, relation, object) in a sentence, some of which may have the overlapping triple problem. To better solve this problem, we directly model the triples and use the quintuple prediction by considering the start and end positions of the two entities in the triple. Given a sentence set $\mathbb{S}$ ={s_1, s_2, …, s_n and a relation set $\mathbb{R}$ ={r_1, r_2, …, r_m the model aims to output a set of quintuples

$$Y = \{(p^{s\_start}, p^{s\_end}, r, p^{o\_start}, p^{o\_end}) | w_{[p^{s\_start}:e^{s\_end}]} \in E, w_{[p^{o\_start}:p^{o\_end}]} \in E, r \in \mathbb{R}\} \quad (1)$$

where $E = \{(w_i, \ldots, w_j) | 1 \leq i \leq j \leq n\}$ represents a set of candidate entity spans. p^{s_start} and p^{s_end} denote the start position and end position of the subject, respectively. p^{o_start} and p^{o_end} denote the start position and end position of the object, respectively. Note that $(p^{s\_start}, p^{s\_end}, r, p^{o\_start}, p^{o\_end}) \neq (p^{o\_start}, p^{o\_end}, r, p^{s\_start}, p^{s\_end})$ in terms of the relation between subject and object.

3.2. Position-aware entity extractor

We present a position-aware entity extractor (PAET) as a universal module to identify subjects and objects, respectively. As shown in Fig. 1, we apply the PAET module in the subject and object extraction tasks. For the sake of generality, subjects and objects are collectively referred to as entities in this section. Formally, the probability of identifying an entity e from a sentence S is generally modeled as:

$$p(e|S) = p(e^{start}|S)p(e^{end}|e^{start}, S) \quad (2)$$

where e^{start} is the start position tag of the entity, and e^{end} is the end position tag. The start position tag and end position tag of the subject are e^{s_start} and e^{s_end}, respectively. Similarly, the start position tag and end position tag of the object are e^{o_start} and e^{o_end}, respectively. The position index of e^{start} and e^{end} in the entity are expressed as p^{start} and p^{end}, respectively. Similarly, p^{s_start} and p^{s_end} denote start position and end position of the subject, respectively. p^{o_start} and p^{o_end} denote start position and end position of the object, respectively.

We use the decomposition strategy of Section 3.1.1 to decompose PAET into two parts: the start position tagger and the end position tagger. These both taggers respectively detect the start and end positions of entities. This decomposition reveals that there is a natural order between the start and end position of an entity, and determines the start position may affect the subsequent end position recognition. This prompted us to propose a position-aware attention mechanism that correlates the two tasks. Specifically, the start position tagger is utilized to extract the start positions of entities. Then, the relative distance information about the start position is obtained according to the tagging result of the entity start position task and is used as the input of the position-aware attention network to learn the encoding representation of each token. Finally, we use the end position tagger to identify the end position. The specific process is as follows:

Start position tagger. A binary classifier is used to assign a binary tag (0/1) to each token in a sentence to detect the start position of an entity. 1 means that the position of the current token corresponds to the start position of a certain entity. 0 means that it is not the start position. The specific operations of the start position tagger for each word are as follows:

$$p(\mathrm{y}_i^{start}) = \sigma(\mathbf{W}^{start}\mathbf{h}_i^{start} + \mathbf{b}^{start}) \tag{3}$$

where y_i^{start} is the binary label of start position for the ith word in the sentence. $p(\mathrm{y}_i^{start})$ expresses the probability that the ith word belongs to the start position. If $p(\mathrm{y}_i^{start})$ exceeds a certain threshold, the ith token will be given a tag 1, otherwise, it will be assigned a tag 0. $\mathbf{h}_i^{start} \in \mathbb{R}^{d_{start}}$ expresses the vector representation of the ith word in the sentence. σ represents the sigmoid activation function. $\mathbf{W}^{start} \in \mathbb{R}^{d_s \times d_{start}}$ expresses the weight and $\mathbf{b}^{start} \in \mathbb{R}^{d_s}$ represents the bias.

End position tagger. Similar to the start position tagger, it also uses a binary classifier to extract the end position. The calculation is as follows:

$$p(\mathrm{y}_i^{end}) = \sigma(\mathbf{W}^{end}\mathbf{h}_i^{end} + \mathbf{b}^{end}) \tag{4}$$

where $\mathbf{W}^{end} \in \mathbb{R}^{d_e \times d_{end}}$ and $\mathbf{b}^{end} \in \mathbb{R}^{d_e}$ are trainable parameters. y_i^{end} represents the tag of end position for the ith word. $p(\mathrm{y}_i^{end})$ expresses the probability that the ith word in an instance belongs to the end position of an entity. $\mathbf{h}_i^{end} \in \mathbb{R}^{d_{end}}$ expresses the encoding representation of the ith token learned by the position-aware attention mechanism.

Position-aware attention mechanism. In the process of identifying entities, we start from the entity start position to find the entity whose end position is the nearest to the start position as a predicted entity. For instance, from Fig. 1, we can find that the end word nearest to the first start word "*Quentin*" is "*Tarantino*", so the first entity recognized will be "*Quentin Tarantino*". It can be observed that the start position of an entity cannot appear behind the end position. Moreover, the distance between the start position and the end position is very close. Considering that the prediction result of the start position may affect the end position recognition and the attention network can more flexibly characterize the text features, we introduce a position-aware attention mechanism. We apply this attention mechanism to learn the word features about the start position, and try to give more weight to words closer to the start position. First, the relative distance between each word and the nearest starting position is calculated when we get the start position of all entities in the sentence. Second, we look up the vector of relative distance in the position embedding matrix. Third, attention weight is calculated through relative distance encoding representation and word vector representation. Fourth, the sentence representation is obtained by multiplying the attention weight and the word vector representation. The detailed operations of the position-aware attention mechanism are as follows

The relative distance between the start position of entities and the current word is as follows:

$$\hat{p}_i = \begin{cases} i - \hat{s}_i, & \text{if } \hat{s}_i \text{ exists} \\ C, & \text{otherwise} \end{cases} \tag{5}$$

where $\hat{s}_i$ is the nearest start position before current index i. $\hat{p}_i$ is the relative distance between the start position $\hat{s}_i$ and the current word x_i. If there is no start position before the current index i, $\hat{s}_i$ will not exist, then $\hat{p}_i$ is designated as a constant C. Taking the sentence in Fig. 1 as an example, we assume that "*nobody*" is the current word. So, "*Quentin*" is the word nearest to the start position before "*nobody*". The relative distance between them is 4. For the current word "*Unchained*", "*Django*" is the word nearest to the start position. The relative distance between them is 1. In this way, we explicitly limit the length of the identified entity and guide the model that the end position is impossible to be in front of the start position.

The initial position embedding matrix is randomly generated. Then, we look up the vector of relative distance $\hat{p}_i$ in the embedding matrix, which is represented as $\mathbf{p}_i \in \mathbb{R}^{d_p}$. When we have obtained the start position and try to identify the ending position, the words in the sentence play different roles under different start positions. If the word at the end position is assigned more weight, then the word may be more easily recognized as the end position. The attention score is obtained as follows:

$$e_i = \mathbf{v}^T \tanh(\mathbf{W}^e\mathbf{h}_i^{start} + \mathbf{W}^p\mathbf{p}_i) \tag{6}$$

$$\alpha_i = \frac{\exp(e_i)}{\sum_{k=1}^{n}\exp(e_k)} \tag{7}$$

where $\mathbf{W}^e, \mathbf{W}^p \in \mathbb{R}^{d_{patt} \times d_{start}}$, $\mathbf{v} \in \mathbb{R}^{d_{patt}}$, $\mathbf{W}^{patt} \in \mathbb{R}^{d_{end} \times d_{start}}$ are trainable parameters. Note that d_{start} is the input dimension of the position-aware attention network, which changes with the change of the dimension of the input. In this way, the attention score can measure the importance of each word to the nearest start position. The sentence representation is generated by multiplying the attention score and the word vector representation.

$$\mathbf{h}_i^{end} = \alpha_i(\mathbf{W}^{patt}\mathbf{h}_i^{start}) \tag{8}$$

where $\mathbf{W}^{patt} \in \mathbb{R}^{d_{end} \times d_{start}}$ are trainable parameters. $\mathbf{h}_i^{end} \in \mathbb{R}^{d_{end}}$ is the encoding representation corresponding to the index i obtained by using the position-aware attention.

We use binary cross-entropy to define the objective function for entity recognition:

$$\begin{aligned}\mathcal{L}_{PAET} = & -\frac{1}{n}\sum_{i=1}^{n} \mathrm{y}_i^{start}\log(p(\mathrm{y}_i^{start})) + (1-\mathrm{y}_i^{start})\log(p(1-\mathrm{y}_i^{start})) \\ & + \mathrm{y}_i^{end}\log(p(\mathrm{y}_i^{end})) + (1-\mathrm{y}_i^{end})\log(p(1-\mathrm{y}_i^{end}))\end{aligned} \tag{9}$$

3.3. Joint extraction system

We present a joint extraction model with position-aware attention and relation embedding, which is shown in Fig. 1. The model is mainly composed of three parts: BERT encoder, subject extractor, and object extractor. The BERT encoder provides the underlying shared encoding representation. The subject extractor mainly focuses on determining all possible subjects in the sentence. The object extractor identifies objects related to the relations and subjects. Specifically, we first use the BERT encoder to encode sentences or relational labels. Secondly, the subject extractor uses the PAET proposed in Section 3.1 to detect the start and end positions of subjects to obtain candidate subjects in the sentence. Thirdly, the object extractor obtains the sentence representations under different subjects and relations by the specific subject-relation attention mechanism, and reduces the useless noise features via the specific subject-relation gate mechanism, and uses the PAET module to extract objects.

3.3.1. Shared sentence representation with BERT encoder

BERT (Bidirectional Encoder Representation from Transformers) [38] is a large-scale pre-trained language model that focuses on learning general language representations. BERT is used to obtain vector representations and consists of a multilayer bidirectional transformer encoder [39]. BERT has made remarkable improvements in various NLP tasks, e.g. sentiment classification [40], question answering [41], and reading comprehension [42].

In PARE-Joint, we use BERT to encode relational labels or sentences containing entity pairs. The parameters of BERT are initialized by the pre-trained BERT weights. Formally, given a text $x = \{x_1^a, x_2^a, \ldots, x_k^a\}$, its encoding representation is $\mathbf{X}^a = \{\mathbf{x}_1^a, \mathbf{x}_2^a, \ldots, \mathbf{x}_{d_a}^a\}$, where $a = \{r, S\}$ represents the word type, r is the relational label class, S is the sentence class, and d_a is the length of the text. The encoding representation $\mathbf{X}^a$ is taken as the input of the BERT encoder to obtain contextual information. The calculation process is as follows:

$$\mathbf{H}^a = BERT(\mathbf{X}^a) \tag{10}$$

where $\mathbf{H}^a = \{\mathbf{h}_1^a, \mathbf{h}_2^a, \ldots, \mathbf{h}_{d_a}^a\} \in \mathbb{R}^{d_a \times d_w}$ represents the word vector representation generated by the BERT encoder, d_a expresses the length of the text, and d_w represents the dimension of the word encoding representation. $\mathbf{h}_i^a \in \mathbb{R}^{d_w}$ expresses the encoding representation of the ith word. Note that the input in the original BERT paper is a sentence pair, while in our model it is a single text. Thus, the segmented embedding of the original paper is not used in Eq. (10). For a more detailed description of BERT, we recommend that readers refer to Ref. [38,39].

3.3.2. Subject extraction

Subject extraction aims to identify all possible subjects in the sentence. We use the shared sentence representation $\mathbf{H}^S$ as input and send it to PAET to extract the subjects:

$$Q_{SE} = PAET(\mathbf{H}^S) \tag{11}$$

where $Q_{SE} = \{s_1, s_2, \ldots, s_m\}$ is the set containing all identified subjects, and m is the number of subjects.

3.3.3. Object extraction

Label embedding has been successfully used for image classification [43]. Moreover, it has been proven to effectively enhance the classification performance of images, text, and sound [44]. Recently, label embedding has been introduced into NLP, and some research progress has been made in the task of text classification [45,46]. In terms of named entity recognition, Luo et al. [47] use label embedding attention mechanism to learn enhanced sentence representation; Bai et al. [48] proposed an adversarial entity recognition model by Part-Of-Speech (POS) label embedding, which effectively utilizes complementary part-of-speech information. In terms of relation extraction, a few researchers currently use label embedding to solve the long-tail relation problem [49,50]. Han et al. [49] represent relation labels as tree structures, and connect sentences and each layer of tree structures with the weighted summation by attention mechanism as the encoding representation of sentences. In this way, the model can fully learn high-level label embeddings and bring some additional information to the long-tail relation with fewer training examples. Zhang et al. [50] combine knowledge graph embedding and GCN to study relational knowledge, and use coarse-to-fine knowledge-aware attention network to integrate relational knowledge into relation extraction, which effectively improved the relation extraction performance. Inspired by these tasks, we also use label embeddings while the models are completely different. We use known relational embeddings and obtained encoding representation of subjects to effectively guide the object extractor to predict objects, thus solving the overlapping triple problem. Specifically, we introduce a subject and relation guided attention network, which uses the specific subject-relation attention mechanism and a specific subject-relation gate mechanism to obtain the enhanced sentence vector representation for object extraction. The specific subject-relation attention mechanism assigns different attention weights to each word and learns specific word feature representations under each subject and relation. The specific subject-relation gate mechanism aims to reduce the influence of noise caused by unrelated features in word feature representation. Then, the PAET module is used to extract all possible objects in the sentence. We describe its detail below.

Specific subject-relation attention mechanism. To obtain the interaction between the subjects and the relations and learn different sentence representations related to different subjects and relationships, we propose a specific subject-relation attention mechanism. The solution to the traditional relation classification task use dot product operations to match sentence-level embeddings with label embeddings to obtain relevant information. The probability that a sentence belongs to a certain relational label largely depends on the overall relevance score of the sentence, not on word-level relevance information. However, the word-level relevance score can provide clear information for object recognition. Thus, we use word-level relevance scores for the object recognition task. The key idea behind the specific subject-relation attention mechanism is to explicitly calculate the relevance score between words and specific subjects and relations, assigning different weights to words to obtain the word representations under each subject and relation. The sentence representation obtained in this way is easier to find the object corresponding to the subject and relations. The attention weights obtained are as follows:

$$\mathbf{h}^{rs} = \mathbf{W}^{sub}\mathbf{h}_j^{sub} + \mathbf{W}^r\tilde{\mathbf{H}}_k^r \tag{12}$$

$$\hat{e}_i = \frac{(\mathbf{W}^{rs}\mathbf{h}^{rs} + \mathbf{b}^{rs})(\mathbf{W}^S\mathbf{h}_i^S + \mathbf{b}^S)^T}{\sqrt{d_{rs}}} \tag{13}$$

$$\hat{\alpha}_i = \frac{\exp(\hat{e}_i)}{\sum_{k=1}^{n}\exp(\hat{e}_k)} \tag{14}$$

where $\mathbf{h}_j^{sub} \in \mathbb{R}^{d_w}$ expresses average encoding representation between the start and end words of the jth subject from the shared sentence representation, such as $\mathbf{h}_1^{Quentin\ Tarantino} = mean(\mathbf{h}_{Quentin}^S, \mathbf{h}_{Tarantino}^S)$. $\tilde{\mathbf{H}}_k^r = \frac{1}{d_r}\sum_{i=1}^{d_r}\mathbf{h}_{ik}^r$ represents average encoding representation between the start and end words of the kth relation. d_r expresses the number of the tokens in the label. $\mathbf{h}^{rs} \in \mathbb{R}^{d_{rs}}$ represents the vector representation of the fusion of $\tilde{\mathbf{H}}_k^r$ and $\mathbf{h}_j^{sub}$. $\mathbf{W}^r, \mathbf{W}^{sub}, \mathbf{W}^S \in \mathbb{R}^{d_w \times d_w}$, $\mathbf{W}^{rs} \in \mathbb{R}^{d_w \times d_w}$,

Table 2
Statistics of the datasets.

Dataset	Training dataset	Validation dataset	Testing dataset	Relation types
NYT	56195	5000	5000	24
WebNLG	5019	500	703	246
NYT10	70339	352	4006	29
NYT11	62648	313	369	12

$\mathbf{b}^{rs}, \mathbf{b}^{S} \in \mathbb{R}^{d_w}$ are parameters. $\hat{e}_i$ represents relevance score between ith token and jth subject and kth relation. $\hat{\alpha}_i$ represents the attention weight. The vector representation $\hat{\mathbf{h}}_i \in \mathbb{R}^{d_w}$ under the subject and relation is generated by multiplying the shared sentence representation with the corresponding weights:

$$\hat{\mathbf{h}}_i = \hat{\alpha}_i(\mathbf{W}^1\mathbf{h}_i^S + \mathbf{b}^1) \tag{15}$$

where $\mathbf{W}^1 \in \mathbb{R}^{d_w \times d_w}$, $\mathbf{b}^1 \in \mathbb{R}^{d_w}$ are parameters.

Specific subject-relation gate mechanism. So far, we have obtained word vector representations related to subject and relation. However, word vector representations are meaningful for object extraction only if the subject and relation are associated with words, irrelevant representations will only confuse the subsequent extraction process. To adaptively control the word vector representations provided by the previous attention layer, we introduce a specific subject-relation gate mechanism, which is defined as follows:

$$g = \sigma(\mathbf{W}^4(\mathbf{W}^2\mathbf{h}_j^{sub} + \mathbf{W}^3\tilde{\mathbf{H}}_k^r) + \mathbf{b}^2) \tag{16}$$

$$\mathbf{h}_i^{obj\_s} = g \odot (\mathbf{W}^5\hat{\mathbf{h}}_i + \mathbf{b}^3) \tag{17}$$

where $\mathbf{W}^2, \mathbf{W}^3, \mathbf{W}^4, \mathbf{W}^5 \in \mathbb{R}^{d_w \times d_w}$, $\mathbf{b}^2, \mathbf{b}^3 \in \mathbb{R}^{d_w}$ are parameters. $\odot$ represents dot product operation. The value of g ranges from 0 to 1, which is regarded as the percentage of information to be retained. Eq. (17) aims to preserve the word features that are useful for the subject and relation.

Similar to the subject extraction task, object extraction also uses PAET to extract the objects. Thus, we take $\mathbf{H}^{object} = \{\mathbf{h}_1^{obj\_s}, \mathbf{h}_2^{obj\_s}, \ldots \mathbf{h}_{d_S}^{obj\_s}\}$ as the input of PAET proposed in Section 3.2 to extract the objects related to the subject and relation:

$$Q_{OE}^{sub,rel} = PAET(\mathbf{H}^{object}) \tag{18}$$

where $Q_{OE}^{sub,rel} = \{o_1, o_2, \ldots, o_n\}$ is the set of all extracted objects related to the subject *sub* and relation *rel*, and n denotes the number of objects in the set.

3.3.4. Loss

The proposed joint extraction model is mainly divided into two interrelated sub-modules, namely the subject extraction module and the object extraction module. The former module identifies subjects contained in the sentence. The latter module extracts all objects under different subjects and relations. The subjects, objects, and known relations constitute relational triplets. Thus, the loss function contains two parts: subject extraction loss $\mathcal{L}_{SE}$ and object extraction loss $\mathcal{L}_{OE}$. In the paper, we need to train the two sub-modules jointly. The joint loss is the sum of subject extraction loss and object extraction loss:

$$\mathcal{L} = \mathcal{L}_{SE} + \mathcal{L}_{OE} \tag{19}$$

Following previous methods [33], we use stochastic gradient descent to minimize $\mathcal{L}$.

3.3.5. Training process

To explain our model more clearly, we describe our algorithm flow in the form of pseudo-code in Algorithm 1. Because PARE-Joint is trained in a mini-batch way, a mini-batch $\mathbb{S}_{\text{batch}}$ of size m is sampled and fed into the proposed model in line 3. Line 8 to Line 14 are the subject recognition process using the position-aware attention network, in which the subject set and the loss value of subject extraction are obtained. Line 15 to Line 24 are the object extraction process using the subject and relation guided attention network and position-aware attention network. The object extraction takes the identified subjects and the known relations as input to extract the corresponding objects and calculate the loss value. Finally, the loss value of the model is obtained and the parameters are updated from Line 25 to Line 26.

4. Experiments

4.1. Dataset and settings

4.1.1. Datasets

We evaluate our method on four benchmark datasets shown in Table 2:

(1) NYT[1] [51] is a large-scale dataset, and it is generated by using the distant supervision (DS) [52] to align entity pairs from Freebase with plain texts in the New York Times corpus. We employ NYT published by Zeng et al. [25], which includes 56195/5000/5000 instances for training, validation, and testing, respectively.

(2) WebNLG [53] was utilized for natural language generation tasks and later employed by Zeng et al. [25] to jointly extract entities and relations. It uses 5019 sentences as the training dataset, 500 sentences as the validation dataset, and 703 sentences as the testing dataset.

(3) NYT10 is a dataset published by Wei et al. [33]. The dataset has 29 relations, consisting of 70339 training sentences, 4006 testing sentences, and 352 validation sentences.

(4) NYT11 is a smaller version than NYT10 and NYT, and it has only 12 relation types. Like NYT and NYT10, its training dataset is generated by DS, while the testing dataset is manually labeled by Hoffmann et al. [54]. We employ NYT11 published by Wei et al. [33], where 62648 sentences are used as the training dataset, 313 sentences are used for verification, and 369 sentences are used for testing.

Note that the number of words contained in the sentences of NTY and WebNLG are both less than 100. In addition, most sentences in the testing dataset of NYT11 only contain a single triplet.

4.1.2. Parameter settings

We set the batch size to 6 and the dimension of the position embedding to 30. We use Adam to optimize the network weights [55]. Following previous methods [33], the encoder we used is [BERT-Base, Cased].[2] The max length of the input sentence is set to 100, and thresholds for both start and end position taggers are set to 0.5.

[1] http://iesl.cs.umass.edu/riedel/ecml/.

[2] Available at: https://storage.googleapis.com/bert models/2018 10 18/cased L-12 H-768 A-12.zip.

Algorithm 1: Training Procedure of PARE-Joint

Input: Training sentence set $\mathbb{S}=\{s_1, s_2, \ldots, s_n\}$, relation set $\mathbb{R}=\{r_1, r_2, \ldots, r_m\}$, the pre-trained BERT parameters Φ

1. Initialize position embeddings and learnable parameters;
2. Obtain relation embeddings $\mathbf{R}=[\mathbf{r}_1, \mathbf{r}_2, \ldots, \mathbf{r}_m]$ with Eq. (10);
3. **for** *epoch* t=1 to T **do**
4. sample a mini-batch $\mathbb{S}_{\text{batch}}$ of size l;
5. $\mathcal{L}_{SE}=0$;
6. $\mathcal{L}_{OE}=0$;
7. **foreach** $s_i \in \mathbb{S}_{\text{batch}}$ **do**
8. Obtain sentence embeddings $\mathbf{s}_i$ with Eq. (10);
9. Get start positions $\{p_1^{s\_start}, p_2^{s\_start}, ..., p_{l_s}^{s\_start}\}$ of all possible subjects in the sentence s_i with Eq. (3);
10. Obtain new sentence embeddings obtained by using the position-aware attention with Eq. (5-8);
11. Get end positions $\{p_1^{s\_end}, p_2^{s\_end}, ..., p_{l_s}^{s\_end}\}$ of all possible subjects in the sentence s_i with Eq. (4);
12. Get subject set $Q_{SE}=\{e_1^s, e_2^s, ..., e_{l_s}^s\}$ in the sentence s_i according to start positions and end positions;
13. Get subject extraction loss $\mathcal{L}_{SE}^i$ with Eq. (9);
14. $\mathcal{L}_{SE}=\mathcal{L}_{SE}+\mathcal{L}_{SE}^j$;
15. **foreach** $e_j^s \in Q_{SE}$ **do**
17. **foreach** $r_k \in \mathbb{R}$ **do**
18. Obtain new sentence representation $\mathbf{s}_i$ with Eq. (12-17) via the identified subject and the existing relation embeddings;
19. Get start positions $\{p_1^{o\_start}, p_2^{o\_start}, ..., p_{l_o}^{o\_start}\}$ and end positions $\{p_1^{o\_end}, p_2^{o\_end}, ..., p_{l_o}^{o\_end}\}$ of objects with Eq. (3-8);
20. Get object set $Q_{OE}^{e_j^s, r_k}=\{e_1^o, e_2^o, ..., e_k^o\}$ according to start positions and end positions;
21. Get object extraction loss $\mathcal{L}_{OE}^j$ with Eq. (9);
22. $\mathcal{L}_{OE}=\mathcal{L}_{OE}+\mathcal{L}_{OE}^j$;
23. **end**
24. **end**
25. Get extraction loss $\mathcal{L}$ with Eq. (19);
26. Update the parameters of the model using the stochastic gradient descent;
27. **end**
28. **end**

4.1.3. Baselines

To show the effectiveness of our model on the four public datasets, we compared PARE-Joint with strong baselines.

- NovelTagging [22] introduces a sequence tagging strategy for the joint extraction task, which fails to detect overlapping triples.
- CopyRE [25] adopts an end-to-end neural network model based on Seq2Seq learning with a copy mechanism for extracting entities and relations.
- MultiHead [15] performs an entity recognition task, then models the relation classification task as a multi-head selection problem.
- OrderRL [26] further studies the influence of extraction order based on CopyRE, and applies reinforcement learning to the Seq2Seq model.
- CopyMTL [27] uses a multi-task learning framework to improve the inaccurate entity copying and entity incomplete problem of CopyRE.
- GraphRel [29] utilizes GCN to predict relations between subject and object, which further learns the interaction between entities and relations.
- HRL [28] adopts a hierarchical extraction paradigm, which first detects the relations and then identifies entities via reinforcement learning.
- ETL-Span [22] uses a novel decomposition strategy and a span-based tagging scheme to jointly extract relations and entities.

Table 3
Performance of different methods on the NYT and WebNLG. Bold marks indicate the best result of all methods. # marks results quoted directly from the original papers. * marks results reported by Yu et al. [23] and Wei et al. [33]. + marks results produced with official implementation.

Model	NYT-Partial			WebNLG-Partial			NYT-Exact			WebNLG-Exact		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
NovelTagging* [22]	62.4	31.7	42.0	52.5	19.3	28.3	–	–	–	–	–	–
CopyRE$_{One}$* [25]	59.4	53.1	56.0	32.2	28.9	30.5	–	–	–	–	–	–
CopyRE$_{Mul}$* [25]	61.0	56.6	58.7	37.7	36.4	37.1	–	–	–	–	–	–
MultiHead* [15]	–	–	–	–	–	–	60.7	58.6	59.6	57.5	54.1	55.7
CopyMTL$_{One}$* [27]	72.7	69.2	70.9	57.8	60.1	58.9	–	–	–	–	–	–
CopyMTL$_{Mu}$* [27]	75.7	68.7	72.0	58.0	54.9	56.4	–	–	–	–	–	–
GraphRel$_{1p}$* [29]	62.9	57.3	60.0	42.3	39.2	40.7	–	–	–	–	–	–
GraphRel$_{2p}$* [29]	63.9	60.0	61.9	44.7	41.1	42.9	–	–	–	–	–	–
OrderRL# [26]	77.9	67.2	72.1	63.3	59.9	61.6	–	–	–	–	–	–
HRL# [28]	84.2	77.8	80.9	–	–	–	–	–	–	–	–	–
ETL-Span# [23]	–	–	–	–	–	–	85.5	71.7	78.0	84.3	82.0	83.1
Attention as Relation# [16]	88.1	78.5	83.0	89.5	86.0	87.7	–	–	–	–	–	–
WDec# [32]	**94.5**	76.2	84.4	–	–	–	–	–	–	–	–	–
PNDec# [32]	89.3	78.8	83.8	–	–	–	–	–	–	–	–	–
CasRel+ [33]	89.0	89.5	89.2	92.4	90.2	91.3	88.7	89.2	89.0	92.0	90.3	91.1
PARE-Joint	92.9	**91.4**	**92.1**	**93.8**	**91.0**	**92.4**	**92.9**	**91.4**	**92.1**	**93.4**	**90.8**	**92.1**

• Attention as Relation [16] first applies a conditional random field to recognize the entities, then introduces a supervised multi-head self-attention mechanism to extract overlapping relations.

• WDec [32] utilizes a representation scheme for the relational triple extraction, which can identify entities containing multiple words.

• PNDec [32] adopts a decoding framework with pointer networks, which can extract entire entities by using their start and end positions.

• SPTree [19] proposes an end-to-end model that combines sequence feature and structural feature of the dependency tree to extract entities and relations.

• CasRel [33] is one of the SOTA approaches on four datasets in the relational triple extraction task, which first detects all possible subjects in the input text, and then finds relations and objects for the predicted subject.

4.1.4. Evaluation

Notably, there are two different evaluation metrics that can compare the proposed model against various baselines:

(1) ***Partial Match***: a recognized relational triplet is considered correct when the relation and the start position of subject and object are all correct [25,28,33].

(2) ***Exact Match***: a predicted relational triplet is considered correct when the relation and the boundaries of both subject and object are correct [33].

Since some methods such as CopyRE [25] cannot extract entities with multiple words and they can only be evaluated under the Partial Match. In addition, some methods such as OrderRL [26] are not open-source, so it is difficult to adopt uniform metrics to compare our model with previous baselines. To properly compare the proposed method with various methods, we use the Partial Match and Exact Match metrics for four different datasets respectively, namely NYT-Partial, NYT-Exact, WebNLG-Partial, WebNLG-Exact, NYT10-Partial, and NYT11-Partial. Following previous works [29,33], we employ the standard Precision (Prec.), Recall (Rec.), and F1-score to evaluate the experimental results.

4.2. Experimental results and analysis

The following subsections introduce the main results, discuss the experimental results, provide the detailed analysis and advantages/disadvantage analysis, and describe ablation and case study, respectively. Our experiments achieve the following six research targets

(1) Show the overall performance of our model.

(2) Analyze the performance and mutual influence of model subtasks.

(3) Analyze the ability of the model to extract complex entities when the entities contain different numbers of words. In addition, demonstrate the ability of the model to handle different overlapping patterns and complex scenarios

(4) Discuss the influence of different modules on the performance of relational triples extraction.

(5) Discuss the advantages/disadvantages of our model, and determine the reasons for the advantages/disadvantages by analyzing the performance of different elements in the triplets under different datasets.

(6) Show the actual effect of the model by case study.

4.2.1. Main results

Table 3 shows the results of different methods for the joint entity and relation extraction on NYT and WebNLG datasets. We can find that CasRel, which first recognizes subjects and then extracts objects, performs better than other baselines, especially in NYT-Exact and WebNLG-Exact. This reflects the superiority of this triple extraction scheme. Most importantly, we can see that PARE-Joint outperforms all other baselines in F1-score on the NYT and WebNLG datasets. Specifically, compared with CasRel, PARE-Joint improves 2.9 points on NYT-Partial, 1.1 on WebNLG-Partial, 3.1 on NYT-Exact, and 1.0 on WebNLG-Exact, respectively. Although the SOTA methods, CasRel, have achieved surprising results, they still inevitably have some problems. CasRel ignores the position between the words in the entity, the interaction between the entities and the relations, and the words under different relations and subjects should have different semantic representations. PARE-Joint can not only handle all the above problems, but also can better improve the extraction performance of overlapping triples. Moreover, the Precision and Recall of PARE-Joint on NYT-Partial are higher than CasRel 3.2 and 1.9 percentages. The Precision and Recall on NYT-Exact are higher than CasRel 4.2 and 2.2 percentages.

In addition, to further demonstrate the performance of PARE-Joint in identifying triples, we conducted experiments on NYT10-Partial and NYT11-Partial. As shown in Table 4, we compare the performance of PARE-Joint and other methods on NYT10 and

Table 4
Performance of different methods on the NYT10 and NYT1.

Model	NYT10-Partial			NYT11-Partial			NYT10-Exact			NYT11-Exact		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
NovelTagging* [22]	59.3	38.1	46.4	46.9	48.9	47.9	–	–	–	–	–	–
CopyRE* [25]	56.9	45.2	50.4	34.7	53.4	42.1	–	–	–	–	–	–
HRL# [28]	71.4	58.6	64.4	53.8	53.8	53.8	–	–	–	–	–	–
SPTree* [19]	49.2	55.7	52.2	52.2	54.1	53.1	–	–	–	–	–	–
CasRel+ [33]	77.3	68.9	72.8.	50.0	57.0	53.3	76.6	68.2	72.2	49.6	56.7	53.0
PARE-Joint	**80.8**	**70.3**	**75.2**	**52.2**	**60.0**	**55.8**	**80.1**	**69.7**	**74.5**	**52.0**	**59.7**	**55.6**

Table 5
F1-score of the subject extraction module and the object extraction module.

Model	NYT-Exact		WebNLG-Exact		NYT10-Exact		NYT11-Exact	
	S	O	S	O	S	O	S	O
CasRel	93.2	93.3	95.6	95.0	85.3	84.2	70.0	65.7
PARE-Joint	94.8	94.9	96.4	95.6	86.2	85.3	70.3	67.9

Table 6
Chi-square test results to determine whether subject extraction affects object extraction on the NYT10-Exact.

	O-correct	O-error.	Total
S-correct	3859	173	4032
S-error.	128	936	1064
Total	3987	1109	5096

NYT11. For the NYT10 dataset, PARE-Joint achieves the best results in Recall, Precision and F1 score. In detail, for NYT10-Partial, PARE-Joint is 3.5%, 1.4%, 2.4% higher than CasRel on Precision, Recall, and F1-score, respectively. For NYT10-Exact, PARE-Joint is 3.5%, 1.5%, 1.5% higher than CasRel on Precision, Recall, and F1-score, respectively. These results demonstrate that the proposed model can effectively extract overlapping triples. For the NYT11 dataset, although there are almost no overlapping triples in this dataset, the proposed PARE-Joint outperforms other baselines in all performances. In F1-score, our model increases 2.5% and 2.6% over the SOTA method on NYT11-Partial and NYT11-Exact, respectively. This indicates that our model is also suitable for extracting single triples.

The above results show that our model performs better than the previous method on the four datasets, which proves the effectiveness of our model in extracting single triples and overlapping triples.

4.2.2. Discussion on the experimental results

In this section, we will discuss the following two issues related to our experiments

Why can the performance of subject extraction, object extraction and triple extraction be improved? The previous experimental results (Tables 3 and 4) prove that our model is superior to the previous method in the triple extraction performance. In Section 3, we have introduced that our model is mainly divided into two sub-modules, namely the subject extraction module and the object extraction module. Therefore, to understand why our model outperforms the previous methods, we analyze the extraction capabilities of the two sub-modules. Table 5 shows the extraction results of the subject and the object, where S represents the subject and O represents the object. For the subject extraction module, the F1-score of PARE-Joint improve by 1.6% on NYT-Exact, 1.6% on WebNLG-Exact, 0.9% on NYT10-Exact, and 0.3% on NYT11-Exact. This is mainly because our model applies a position-aware attention mechanism to enhance the influence of the start position on the end position in the entity thereby effectively limiting the end position of the subject and improving the subject extraction performance. For the object extraction module, the F1-score of our model improve by 1.6% on NYT-Exact, 0.8% on WebNLG-Exact, 1.1% on NYT10-Exact, and 2.2% on NYT11-Exact. This is mainly due to the following two factors: (1) Our model employs a subject and relation guided attention network to learn sentence representations under different subjects and relations, which helps the model effectively define the start position of the object. (2) Then, the position-aware attention mechanism is used to determine the end position of the object.

Does subject extraction affect object extraction? Chi-square test can be applied to analyze the relation between two variables and determine whether the value of one variable affects that of the other. We adopt this method to confirm whether the subject extraction affects the object extraction in the proposed model. We divide the results of subject extraction into two classes: S-correct and S-error. Similarly, the results of object extraction are divided into two classes: O-correct and O-error. The experimental results are shown in Table 6. We determine whether the subject extraction affects the object extraction under $\alpha = 0.05$.

We assume that subject extraction and object extraction are independent and unrelated. According to experimental results analysis, $\chi^2 > \chi^2_{0.05}(1)$. Thus, the original hypothesis is rejected and we can confirm that subject extraction and object extraction are related. Besides, it can be observed from Table 6 that when the result of extracting the subject is correct, the accuracy rate of the object extraction is 95.7%, and when the subject extraction is incorrect, the error rate of the object extraction is 88.0%. This demonstrates that accurate subject extraction is helpful for object extraction in the next step. Therefore, the proposed position-aware attention mechanism can not only provide the performance of entity extraction but also the accuracy of triple extraction.

4.2.3. Detailed analysis

Analysis on the different number of words in the entity. To prove the ability of PARE-Joint to extract entities containing multiple words, we divided the entities on NYT10 into 4 subclasses, each of which includes entities with 1, 2, 3, or ≥ 4 words. As shown in Table 7, it can be seen that all the performances of the two models gradually decrease as the number of words increases. Compared with the SOTA model CasRel, PARE-Joint

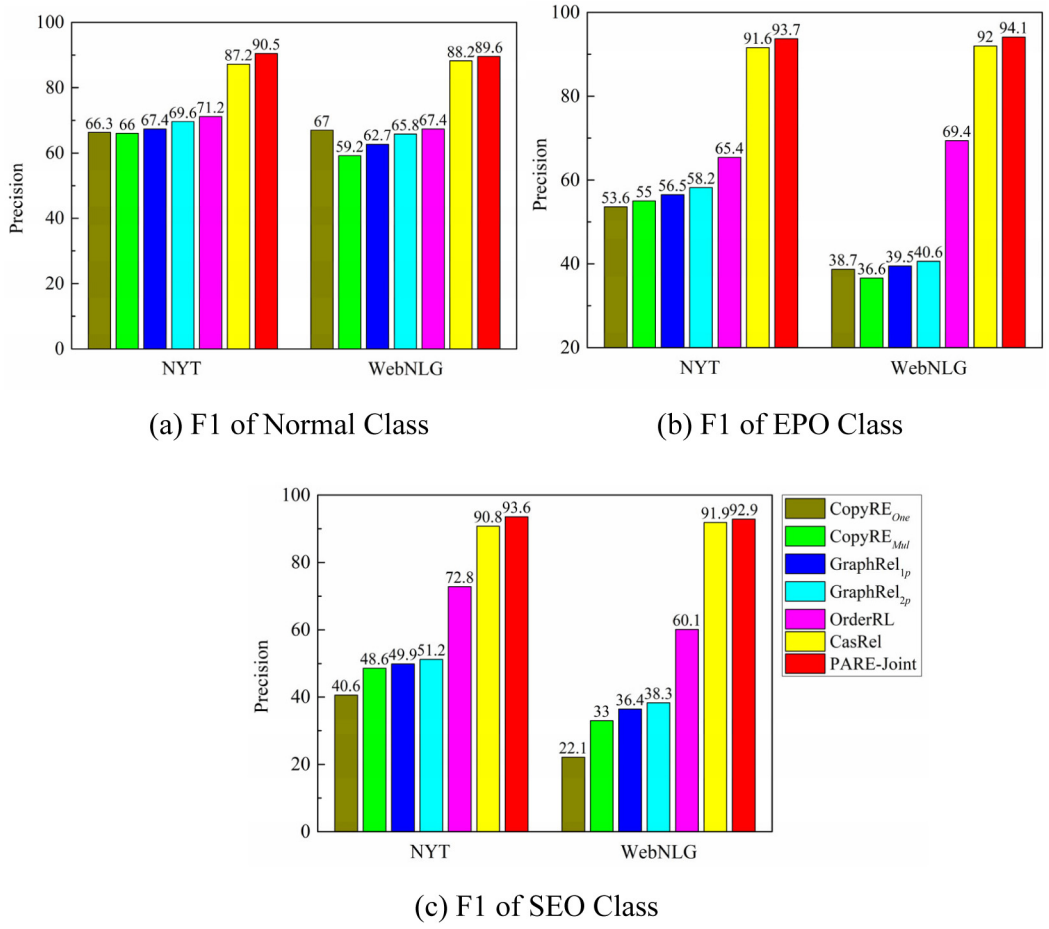

Fig. 2. F1-score of extracting triples from sentences with three overlapping patterns on NYT-Partial and WebNLG-Partial.

Table 7
Comparison of the SOTA model and PARE-Joint under the different number (denoted as N) of words in the entity on NTY10.

Model	N = 1			N = 2			N = 3			N ≥ 4		
	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
CasRel	81.9	64.3	72.0	75.0	62.7	68.2	73.1	62.0	67.1	72.7	65.3	68.9
Our	84.5	65.6	73.8	78.6	64.3	70.7	76.0	63.3	69.1	74.0	65.3	69.4

achieves better results for all the number of words in the entity. In detail, PARE-Joint increased by 1.8%, 2.5%, 2.0% and 0.5% in F1-score, respectively. The main reason for the improvement is that PARE-Joint can enhance the influence of the start position on the end position in an entity, and effectively determine the end position of the entity. These analyzes prove that our method can better recognize entities with more words, thereby improving the performance of extracting entities and triples.

Analysis on overlapping triples. To further research the capability of the proposed PARE-Joint in extracting overlapping triples, we follow previous works [23,33] to conduct experiments on NYT-Partial and WebNLG-Partial. Fig. 2 demonstrates the F1-scores of PARE-Joint and baselines when identifying triples from sentences with overlapping patterns of Normal, EPO, and SEO in the two datasets. We can observe that the F1-scores of most models on three overlapping patterns show a downward trend. In other words, identifying triples from sentences with overlapping patterns of Normal, EPO, and SEO is from easy to difficult, of which SEO pattern is the most challenging case. In addition, we find that most models have lower extraction performance on overlapping triples, which proves that these models do not excel in solving the overlapping triple problem. However, PARE-Joint achieves the best performance (especially SEO) and presents a steady trend on all three overlapping patterns. For the normal pattern, PARE-Joint is 2.3% and 1.4% higher than CasRel in F1-scores, respectively. For the EPO pattern, PARE-Joint is 2.1% higher than CasRel in F1-scores, respectively. For the SEO pattern, PARE-Joint is 2.8% and 1.0% higher than CasRel in F1-scores, respectively. The main reason is that PARE-Joint can capture the complex interactions between entities and relations, and can identify all potential objects between the subject and the relation. The above analysis demonstrates that PARE-Joint achieves a stable and competitive extraction performance on NYT-Partial and WebNLG-Partial, and can solve the overlapping triple problem more effectively.

Analysis on different number of triples. To certify the ability of PARE-Joint in handling complicated scenarios, we divide sentences on both NYT-Partial and WebNLG-Partial into 5 subclasses, each of which includes sentences with 1, 2, 3, 4, or ≥5 triples. As shown in Table 8, PARE-Joint achieves the best results under all numbers of relational triplets in the sentence. Furthermore, we observe that the F1-score of most models reduces as the number of triples contained in a sentence contains increases. In other words, identifying triples from sentences containing 1, 2, 3, 4, or ≥5 triples is easy to difficult, and triples with N≥5 are the most

Table 8
F1-score of extracting triples from sentences with different numbers (denoted as N) of triples.

Model	NYT-Partial					WebNLG-Partial				
	N = 1	N = 2	N = 3	N = 4	N ≥5	N = 1	N = 2	N = 3	N = 4	N ≥ 5
$CopyRE_{One}$	66.6	52.6	49.7	48.7	20.3	65.2	33.0	22.2	14.2	13.2
$CopyRE_{Mul}$	67.1	58.6	52.0	53.6	30.0	59.2	42.5	31.7	24.2	30.0
OrderRL	71.7	72.6	72.5	77.9	45.9	63.4	62.2	64.4	57.2	55.7
$GraphRel_{1p}$	69.1	59.5	54.4	53.9	37.5	63.8	46.3	34.7	30.8	29.4
$GraphRel_{2p}$	71.0	61.5	57.4	55.1	41.1	66.0	48.3	37.0	32.1	32.1
CasRel	88.0	90.0	91.1	93.6	82.6	88.2	90.6	94.1	91.2	91.3
PARE-Joint	**90.5**	**92.7**	**93.3**	**95.2**	**91.7**	**89.4**	**90.9**	**95.2**	**93.3**	**92.0**

challenging cases. However, our model PARE-Joint shows more stable performance with the increasing of triplets numbers in the sentence. Compared with the SOTA model CasRel, PARE-Joint has improved in all five subclasses on both datasets. In particular, the greatest improvement in the F1-score of our model on NYT-Partial comes from the most difficult subclass (N ≥ 5), and the F1-score has increased by 8.1%. The above result analysis shows that PARE-Joint can be applied to complicated scenarios.

Table 9
Ablation tests on the NYT testing dataset.

Model	NYT-Exact			NYT10-Exact		
	Prec.	Rec.	F1	Prec.	Rec.	F1
PARE-Joint	92.9	91.4	92.1	80.1	69.7	74.5
-position-aware attention	91.8	90.9	91.3	78.5	69.0	73.4
-specific subject-relation attention	91.0	90.9	90.9	78.0	68.8	73.2
-specific subject-relation gate	91.8	89.6	89.8	80.0	67.5	73.2

4.2.4. Ablation study

The ablation experiments are conducted on the NYT reported in Table 9 to demonstrate the effectiveness of position-aware attention mechanism, specific subject-relation attention mechanism, and specific subject-relation gate mechanism in our model. We remove one specific component at a time to observe its impact on the experimental results. It can be observed that these three parts can assist our model in jointly extracting triples. Among them, the specific subject-relation attention mechanism seems to play a more important role.

When the position-aware attention mechanism is removed, the F1-score drops by 1.2% on NYT-Extract and 0.9% on NYT10-Extract. These results show that the position and order of the words in the entity are very important. It is also proved that the position-aware attention mechanism can enhance the influence of the start position on the end position in the entity, thereby effectively improving the performance of the relational triple extraction task and entity recognition task.

Removing the specific subject-relation attention mechanism also degrades the performance of the triple extraction task (∼1.9% and ∼2.1% in terms of Precision, ∼0.5% and ∼0.9% in terms of Recall, ∼1.2% and ∼1.3% in terms of F1-score on NYT-Extract and NYT10-Extract, respectively). We believe that the reason for the decline in model performance is that the specific subject-relation attention mechanism can capture the complex interactions between entities and relations, and can obtain the enhanced word representations, thus affecting the results of triple extraction. It also indicates that our method benefits from relation embedding.

Finally, the specific subject-relation gate mechanism is removed. For NYT-Extract, PARE-Joint has dropped 1.1%, 1.8% and 1.3% in Precision, Recall and F1-score, respectively. For NYT-Extract, PARE-Joint has dropped 0.1%, 2.2% and 1.3% in Precision, Recall and F1-score, respectively. It can be found that the model has a faster decline in Recall. The main reason is that the sentence representation contains noise features, which can easily mislead the model when extracting objects. In addition, it also proves that the specific subject-relation gate can improve the performance of triple extraction

4.2.5. Advantages/disadvantages analysis

In the previous results, we can get the following conclusions: (1) Tables 3 and 4 prove that our model is superior to the previous baselines in the performance of triple extraction. (2) Table 5 proves that the position-aware attention mechanism is beneficial to entity extraction. (3) Table 6 also confirms that PARE-Joint can better extract entities containing multiple words. (4) Tables 7 and 8 demonstrate that our model can better deal with the problem of overlapping triple and can handle complex scenarios.

In this section, we continue to discuss the advantages/disadvantages of our model, and analyze the reasons for the advantages/disadvantages according to the performance of different elements in the triplet under different datasets.

Table 10 shows the extraction performance of the SOTA model CasRel and our model on different elements on the three datasets. (S, R, O) represents a relational triple, where S represents the subject, O represents the object, and R represents the relation between S and O. An element like (S, O) is considered correct only when the subject and the object in the predicted triples (S, R, O) are correct, regardless of the correctness of the predicted relation. Similarly, we consider an element R is correct as long as the relation in the recognized triple is correct, so is S and O.

For different elements, our model achieves the best performance. In detail, for element S, we found that the extraction performance of our model is better than CasRel on NYT, WebNLG and NYT10 datasets. Since the model only adds the position-aware attention mechanism during subject extraction, this result clearly confirms that the position-aware attention mechanism can effectively improve the performance of entity recognition. For element R and element O, compared with CasRel, our model has improved all extraction performance on the three datasets. It reveals that our model can identify more relations and more entities, and proves the effectiveness of our method in identifying relations and entities. For other combined elements like (S, O), the performance of our model also outperforms CasRel. It reflects the effectiveness of our model to deal with the problem of overlapping triples.

For different datasets, although our model has achieved better results than CasRel, the extraction performance and the degree of performance improvement are different on each dataset. In the three datasets, our model has the best extraction performance on WebNLG dataset, but the improvement degree is the lowest. Compared with the other two datasets, our model has the highest improvement degree on the NYT dataset. The performance of our model is greatly improved in the NYT10 dataset, but worse than the other two datasets. The main reasons are as follows

For the NYT dataset, we can find that the performance gap between the F1-scores of (S, R, O) and (S, O) is 0.3%, while the gap between (S, R, O) and (S, R)/(R, O) is 1.5%, respectively. Obviously, the performance gap between (S, R, O) and (S, O) is smaller than the gap between (S, R, O) and (S, R)/(R, O). It is indicated

Table 10
Result on relational triple elements.

Element	Model	NYT-Exact			WebNLG-Exact			NYT 10-Exact−10		
		Prec.	Rec.	F1	Prec.	Rec.	F1	Prec.	Rec.	F1
S	CasRel	94.0	92.5	93.2	97.9	93.4	95.6	85.4	85.2	85.3
	PARE-Joint	95.9	93.8	94.8	98.3	94.6	96.4	86.7	85.7	86.2
R	CasRel	95.8	94.0	94.9	96.1	91.6	93.8	88.9	76.6	82.2
	PARE-Joint	96.8	94.5	95.7	96.7	92.4	94.5	90.2	77.4	83.4
O	CasRel	93.2	93.4	93.3	97.0	93.0	95.0	86.5	81.9	84.2
	PARE-Joint	95.7	94.2	94.9	97.3	94.0	95.6	87.6	83.1	85.3
(S, R)	CasRel	92.9	90.8	91.9	93.9	90.5	92.2	81.7	73.2	77.2
	PARE-Joint	94.7	92.6	93.6	94.8	91.5	93.1	84.0	73.9	78.7
(R, O)	CasRel	92.2	91.2	91.7	94.6	90.9	92.7	82.4	70.5	76.0
	PARE-Joint	94.6	92.7	93.6	95.4	91.7	93.5	84.4	71.7	77.5
(S, O)	CasRel	88.5	90.2	89.3	94.2	91.7	93.0	78.2	77.6	77.9
	PARE-Joint	93.3	91.9	92.4	95.1	92.7	93.9	81.0	78.9	80.0
(S, R, O)	CasRel	88.7	89.2	89.0	92.2	90.0	91.1	76.6	68.2	72.2
	PARE-Joint	92.9	91.4	92.1	93.4	90.8	92.1	80.1	69.7	74.5

that when most entity pairs are correctly recognized, the relations will be correctly extracted. In other words, it implies that identifying relations is somehow easier than identifying entities for our model. Therefore, we believe that the main reason why the performance of our model has improved the most on the NYT dataset is that our model uses a position-aware attention mechanism to better recognize entities. In addition, we also find that the performance of S and O is consistent with that on (S, R) and (R, O), which proves the effectiveness of PARE-Joint in extracting subject and object.

For the WebNLG dataset, there is an obvious gap in performance between (S, R, O) and (S, O), but a small gap between (S, R, O) and (S, R)/(R, O). This is contrary to the results of the NYT dataset, demonstrating that incorrectly extracting relations can lead to model performance degradation more than incorrectly extracting entities. The main reason is that the two datasets contain different numbers of relations (i.e, 24 in NYT and 246 in WebNLG) and different training dataset sizes (i.e, the NYT training dataset contains 56195 sentences and the WebNLG training dataset contains 5019 sentences), which makes it more difficult for the model to extract the relations in WebNLG. Therefore, we have summarized the reasons why the model performance on the WebNLG dataset is lower than that on the NYT data. There are the following aspects: (1) One reason is the performances on WebNLG are already saturated. There are 246 relations and 5019 sentences in the training dataset of WebNLG. Thus, it is very difficult to extract triples containing a large number of relation types on such a small training dataset. These models achieving a 90+ Precision are likely to have exceeded human-level, that is to say, the room for improvement is very limited. (2) The other reason is that some relations in WebNLG have the same or similar meanings, such as *fullName* and *fullname*, *ethnicGroup* and *ethnicGroups*, *language* and *languages*. These relations will affect the extraction performance of the method. In many cases, the proposed method will obtain these two relations, but usually only one of them is labeled in the testing dataset. The lack of these correct labels will seriously affect the result of PARE-Joint obtaining triples.

For the NYT10 dataset, there is a significant gap in performance between (S, O) and (S, R, O), so is (S, R, O) and (S, R)/(R, O). These results show that misidentifying relations and entities will bring more performance degradation. Therefore, although our model has been greatly improved on the NYT10 dataset, its performance is worse than that of the other two datasets. We believe that the reason for this is that the sentences in NYT10 contain more words than those in the NYT and WebNLG datasets. The number of words in the sentences in NYT and WebNLG datasets are both less than 100, while 93.3% of the sentences have more than 100 words in the NYT10 training dataset. Longer sentences contain more complex semantic information, which may affects the performance of triple extraction.

4.2.6. Case study

Table 11 shows some extraction examples of our model and CasRel. Each example is divided into four lines, in which the first line is the sentence, the second line is the extraction result of CasRel, the third line is the result of our model, and the fourth line is the ground truth. We can observe that the first, the second and the third sentence are all Normal examples, and some entities that need to be extracted contain multiple words. In these three examples, our model and CasRel method can effectively extract relations and entities with only a single word. However, it can be clearly seen that CasRel is difficult to extract complex entities. For the first sentence, CasRel lost a word "*Pin*" when recognizing the subject "*Camille Pin*", while our model extracts the entire subject. This result proves that our proposed position-aware attention mechanism can enhance the effect of the start position on the end position of the subject, thereby effectively determining the boundary of the subject. For the second sentence, the object predicted by CasRel is too long, and our prediction is correct. The main reason is that our position-aware attention mechanism effectively limits the boundary of the object. For the third sentence, CasRel incorrectly predicts the start position of the subject, and our prediction is correct. We attribute this result to the fact that the subject and relation guided attention network can learn sentence features related to the subjects and relations, thereby effectively helping the model determine the start position of the object. The fourth sentence is an example of EPO. It can be observed that the three triples identified by CasRel are all wrong. The subjects and relations of the first and second triples are correct, and the objects are wrong. The subject of the third triple is wrong, and the relation and object are correct. The extraction result of CasRel reveals that the extraction scheme of determining the relations and the objects only by the subjects cannot effectively solve the overlapping triple problem. All the triples extracted by our model are correct, which proves that our method can extract overlapping triples more effectively. The last sentence is an instance containing SPO and EPO. We find that the relational triples (*Dreamworks*, */business/company/founders*, *Jeffrey Katzenberg*) and (*Jeffrey Katzenberg*, */business/person/company*, *Dreamworks*) should have been annotated in the sentence but were omitted. Compared with CasRel, our model extracts the entity "*Jeffrey Katzenberg*" that does not appear in the ground truth, and identifies these triples through

Table 11
Case study of our model.

Sentenc1	It has been an unusual tournament from the start for Sharapova , who was two points from defeat against the underpowered [Camille Pin] of [France] in the first round as temperatures soared well above 100 degrees on court and left Sharapova sluggish and disoriented
CasRel	(Camille, /people/person/nationality, France)
PARE-Joint	(Camille Pin, /people/person/nationality, France)
Ground	(Camille Pin, /people/person/nationality, France)
Sentenc2	And now that [South Forida] has filled up , metropolitan new yorkers are popping up in nearly every place where [Florida] sand meets the sea.
CasRel	(Florida, /location/location/contains, South Forida has filled up , metropolitan new yorkers are popping up in nearly every place where [Florida] sand)
PARE-Joint	(Florida, /location/location/contains, South Forida)
Ground	(Florida, /location/location/contains, South Forida)
Sentence3	And while [Port St. Lucie] , the largest city in [St. Lucie County], is n't on the ocean, it offers water of a different sort : the meandering st. lucie river , which was a setting in the james bond movie "moonraker" and which is like a piece of old Florida preserved.
CasRel	(St. Lucie County, /location/location/contains, St. Lucie)
PARE-Joint	(St. Lucie County, /location/location/contains, Port St. Lucie)
Ground	(St. Lucie County, /location/location/contains, Port St. Lucie)
Sentence4	[Zagreb] take Vienna's florid architecture , throw in Budapest's bubbling cafe culture , and you get [Zagreb], [Croatia]'s grand capital.
CasRel	(Croatia, /location/country/capital, Budapest) (Croatia, /location/country/administrative_divisions, Budapest) (Budapest, /location/administrative_division/country, Croatia)
my	(Croatia, /location/country/capital, Zagreb)
	(Croatia, /location/country/administrative_divisions, Zagreb) (Zagreb, /location/administrative_division/country, Croatia) (Croatia, /location/location/contains, Zagreb)
Ground	(Croatia, /location/country/capital, Zagreb) (Croatia, /location/country/administrative_divisions, Zagreb) (Zagreb, /location/administrative_division/country, Croatia) (Croatia, /location/location/contains, Zagreb)
Sentence5	It is hard to say whether the unusual heat behind the evening owes more to interest in Mr. Obama or to the three men who spearheaded the fund-raiser : the [Dreamworks] co-founders [David Geffen], [Jeffrey Katzenberg] and [Steven Spielberg].
CasRel	(Dreamworks, /business/company/founders, David Geffen) (David Geffen, /business/person/company, Dreamworks) (Dreamworks, /business/company/founders, Steven Spielberg) (Steven Spielberg, /business/person/company, Dreamworks)
PARE-Joint	(Dreamworks, /business/company/founders, David Geffen) (David Geffen, /business/person/company, Dreamworks) (Dreamworks, /business/company/founders, Steven Spielberg) (Steven Spielberg, /business/person/company, Dreamworks) (Dreamworks, /business/company/founders, Jeffrey Katzenberg) (Jeffrey Katzenberg, /business/person/company, Dreamworks)
Ground	(Dreamworks, /business/company/founders, David Geffen) (David Geffen, /business/person/company, Dreamworks) (Dreamworks, /business/company/founders, Steven Spielberg) (Steven Spielberg, /business/person/company, Dreamworks)

overlapping relation extraction. These results indicate that PARE-Joint can predict more triples. In addition, the annotated triples of the dataset are not always the ground truth, which may affect the evaluation of our model.

5. Conclusion

In this paper, we propose a joint extraction model with position-aware attention and relation embedding, named PARE-Joint. Different from previous studies, the model regards the relations as known information and determines the corresponding objects in the sentence through the existing relations and the identified subjects. In this way, the model not only captures the interaction between entities and relations but also better solves the overlapping triple problem. In addition, taking into account the effect of the word order in the entity on the performance of entity recognition, the model uses a position-aware entity extractor to extract subjects and objects in the sentence, respectively. By comparing with the experimental results of previous methods, the proposed model achieves state-of-the-art performance on four public datasets, which proves the effectiveness of our model. Furthermore, detailed experiments also show that our method can not only improve the performance of triple extraction but also better solve the overlapping triple problem.

In the future, the proposed model will be tried to be applied to other information extraction tasks, such as document-level tuple extraction and event extraction. Another direction is to extend our model to make it more suitable for the triple extraction of long sentences.

CRediT authorship contribution statement

Tiantian Chen: Conceptualization, Methodology, Software, Validation, Writing. **Lianke Zhou:** Supervision, Writing – review & editing, Funding acquisition. **Nianbin Wang:** Supervision, Writing – review & editing. **Xirui Chen:** Data curation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding statement

This work is supported by the Basic Research Project, China (JCKY2019604C004).

References

[1] L. Li, J. Wang, J. Li, Q. Ma, J. Wei, Relation classification via keyword-attentive sentence mechanism and synthetic stimulation loss, IEEE/ACM Trans. Audio Speech Lang. Process. 27 (9) (2019) 1392–1404.
[2] M. Nickel, K. Murphy, V. Tresp, E. Gabrilovich, A review of relational machine learning for knowledge graphs, Proc. IEEE 104 (1) (2015) 11–33, http://dx.doi.org/10.1109/JPROC.2015.2483592.
[3] Y.K. Lin, Z.Y. Liu, M.S. Sun, Y. Liu, X. Zhu, Learning entity and relation embeddings for knowledge graph completion, in: Proc. Twenty-Ninth AAAI Conf. Artif. Intel, 2015, pp. 2181–2187.
[4] C. Shi, S. Liu, S. Ren, S. Feng, M. Li, M. Zhou, H. Wang, Knowledge-based semantic embedding for machine translation, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2016, pp. 2245–2254.
[5] R. Das, M. Zaheer, S. Reddy, A. McCallum, Question answering on knowledge bases and text using universal schema and memory networks, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2017, pp. 358–365.
[6] B. Yang, C. Cardie, Joint inference for fine-grained opinion extraction, in: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2013, pp. 1640–1649.
[7] S. Singh, S. Riedel, B. Martin, J. Zheng, A. McCallum, Joint inference of entities, relations, and coreference, in: Proceedings of the 2013 Workshop on Automated Knowledge Base Construction, 2013, pp. 1–6.
[8] Y. Choi, E. Breck, C. Cardie, Joint extraction of entities and relations for opinion recognition, in: Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing, 2006, pp. 431–439.
[9] D. Nadeau, S. Sekine, A survey of named entity recognition and classification, Lingvisticae Investigationes 30 (1) (2007) 3–26, http://dx.doi.org/10.1075/li.30.1.03nad.
[10] D. Wang, P. Tiwari, S. Garg, H. Zhu, P. Bruza, Structural block driven enhanced convolutional neural representation for relation extraction, Appl. Soft Comput. 86 (2020) 105913.
[11] M.R. Gormley, M. Yu, M. Dredze, Improved relation extraction with feature-rich compositional embedding models, 2015, arXiv:1505.02419.
[12] X. Yu, W. Lam, Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach, in: Proceedings of the 23rd International Conference on Computational Linguistics: Posters, 2010, pp. 1399–1407.
[13] Q. Li, H. Ji, 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), 2014, pp. 402–412.
[14] X. Ren, Z. Wu, W. He, M. Qu, C.R. Voss, H. Ji, J. Han, Cotype: Joint extraction of typed entities and relations with knowledge bases, in: Proceedings of the 26th International Conference on World Wide Web, 2017, pp. 1015–1024.
[15] G. Bekoulis, J. Deleu, T. Demeester, C. Develder, Joint entity recognition and relation extraction as a multi-head selection problem, Expert Syst. Appl. 114 (2018) 34–45.
[16] J. Liu, S. Chen, B. Wang, J. Zhang, N. Li, T. Xu, Attention as relation: Learning supervised multihead self-attention for relation extraction, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 2020, pp. 3787–3793.
[17] S. Wang, Y. Zhang, W. Che, T. Liu, Joint extraction of entities and relations based on a novel graph scheme, in: IJCAI, 2018, pp. 4461–4467.
[18] S. Zheng, Y. Hao, D. Lu, H. Bao, J. Xu, H. Hao, B. Xu, Joint entity and relation extraction based on a hybrid neural network, Neurocomputing 257 (2017) 59–66.
[19] M. Miwa, M. Bansal, End-to-end relation extraction using LSTMs on sequences and tree structures, in: ACL, 2016, pp. 1105–1116.
[20] Z. Geng, Y. Zhang, Y. Han, Joint entity and relation extraction model based on rich semantics, Neurocomputing 429 (2021) 132–140.
[21] T. Zhao, Z. Yan, Y. Cao, Z. Li, Asking effective and diverse questions: a machine reading comprehension based framework for joint entity-relation extraction, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 3948–3954.
[22] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, B. Xu, Joint extraction of entities and relations based on a novel tagging scheme, in: ACL, 2017, pp. 1227–1236.
[23] B. Yu, Z. Zhang, X. Shu, T. Liu, Y. Wang, B. Wang, S. Li, Joint extraction of entities and relations based on a novel decomposition strategy, in: ECAI, 2020, pp. 2282–2289.
[24] H. Ye, N. Zhang, S. Deng, M. Chen, C. Tan, F. Huang, H. Chen, Contrastive triple extraction with generative transformer, Proc. AAAI Conf. Artif. Intell. 35 (16) (2021) 14257–14265.
[25] X. Zeng, D. Zeng, S. He, K. Liu, J. Zhao, 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), 2018, pp. 506–514.
[26] X. Zeng, S. He, D. Zeng, K. Liu, S. Liu, J. Zhao, 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, 2019, pp. 367–377.
[27] D. Zeng, H. Zhang, Q. Liu, Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning, Proc. AAAI Conf. Artif. Intell. 34 (05) (2020) 9507–9514.
[28] R. Takanobu, T. Zhang, J. Liu, M. Huang, A hierarchical framework for relation extraction with reinforcement learning, Proc. AAAI Conf. Artif. Intell. vol. 33 (01) (2019) 7072–7079.
[29] T.J. Fu, P.H. Li, W.Y. Ma, 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, 2019, pp. 1409–1418.

[30] K. Zhao, H. Xu, Y. Cheng, X. Li, K. Gao, Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction, Knowl.-Based Syst. 219 (2021) 106888.
[31] H. Fei, Y. Ren, D. Ji, Boundaries and edges rethinking: An end-to-end neural model for overlapping entity relation extraction, Inf. Process. Manage. 57 (6) (2020) 102311.
[32] T. Nayak, H.T. Ng, Effective modeling of encoder–decoder architecture for joint entity and relation extraction, Proc. AAAI Conf. Artif. Intell. 34 (05) (2020) 8528–8535.
[33] Z. Wei, J. Su, Y. Wang, Y. Tian, Y. Chang, A novel cascade binary tagging framework for relational triple extraction, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 1476–1488.
[34] G. Hinton, L. Deng, D. Yu, G.E. Dahl, A. Mohamed, N. Jaitly, V. Vanhoucke, P. Nguyen, T.N. Sainath, B. Kingsbury, Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal Process. Mag. 29 (6) (2012) 82–97.
[35] S. Wang, Y. Jiang, F.L. Chung, P. Qian, Feedforward kernel neural networks, generalized least learning machine, and its deep learning with application to image classification, Appl. Soft Comput. 37 (2015) 125–141.
[36] X. Zhang, D. Goldwasser, Sentiment tagging with partial labels using modular architectures, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 579–590.
[37] L. Liu, J. Shang, X. Ren, F. Xu, H. Gui, J. Peng, J. Han, Empower sequence labeling with task-aware neural language model, Proc. AAAI Conf. Artif. Intell. 32 (1) (2018).
[38] J. Devlin, M.W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2019, pp. 4171–4186.
[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proc. Adv. Neural Inf. Process. Syst, 2017, pp. 5998–6008.
[40] C. Du, H. Sun, J. Wang, Q. Qi, J. Liao, Adversarial and domain-aware bert for cross-domain sentiment analysis, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Online, 2020, pp. 4019–4028.
[41] I. Yamada, A. Asai, H. Shindo, H. Takeda, Y. Matsumoto, LUKe: Deep contextualized entity representations with entity-aware self-attention, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2020, pp. 6442–6454.
[42] P. He, X. Liu, J. Gao, W. Chen, DeBERTa: Decoding-enhanced BERT with disentangled attention, 2020, arXiv:2006.03654.
[43] Z. Akata, F. Perronnin, Z. Harchaoui, C. Schmid, Label-embedding for image classification, IEEE Trans. Pattern Anal. Mach. Intell. 38 (7) (2015) 1425–1438.
[44] Y. Wang, W. Zheng, Y. Cheng, D. Zhao, Two-level label recovery-based label embedding for multi-label classification with missing labels, Appl. Soft Comput. 99 (2021) 106868.
[45] G. Wang, C. Li, W. Wang, Y. Zhang, D. Shen, X. Zhang, L. Carin, Joint embedding of words and labels for text classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2018, pp. 2321–2331.
[46] H. Zhang, L. Xiao, W. Chen, Y. Wang, Y. Jin, Multi-task label embedding for text classification, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 4545–4553.
[47] Y. Luo, F. Xiao, H. Zhao, Hierarchical contextualized representation for named entity recognition, Proc. AAAI Conf. Artif. Intell. 34 (05) (2020) 8441–8448.
[48] Y. Bai, Y. Wang, B. Xia, Y. Li, Z. Zhu, Adversarial named entity recognition with POS label embedding, in: 2020 International Joint Conference on Neural Networks, IJCNN, 2020, pp. 1–8.
[49] X. Han, P. Yu, Z.Liu, M. Sun, P. Li, Hierarchical relation extraction with coarse-to-fine grained attention, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 2236–2245.
[50] N. Zhang, S. Deng, Z. Sun, G. Wang, X. Chen, W. Zhang, H. Chen, Long-tail relation extraction via knowledge graph embeddings and graph convolution networks, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 3016–3025.
[51] S. Riedel, L. Yao, A. McCallum, Modeling relations and their mentions without labeled text, in: Proc. Eur. Conf. Mach. Learn. Prins Pract. Knowl. Discovery Databases, 2010, pp. 148–163.
[52] M. Mintz, S. Bills, R. Snow, D. Jurafsky, Distant supervision for relation extraction without labeled data, in: Proc. 47th Annu. Meeting ACL 4th Int. Joint Conf. Natural Lang. Process. AFNLP, Aug. 2009, 2009, pp. 1003–1011.
[53] C. Gardent, A. Shimorina, S. Narayan, L. Perez-Beltrachini, Creating training corpora for nlg micro-planning, in: 5th Annual Meeting of the Association for Computational Linguistics, ACL, 2017, pp. 179–188.
[54] R. Hoffmann, C. Zhang, X. Ling, L. Zettlemoyer, D. Sweld, Knowledge-based weak supervision for information extraction of overlapping relation, in: Proc. Annu. Meeting Assoc. Comput. Linguistics, Human Lang. Technol., Jun. 2011, 2011, pp. 541–550.
[55] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014, arXiv preprint arXiv:1412.6980.