PasCore: A Chinese Overlapping Relation Extraction Model Based on Global Pointer Annotation Strategy

Peng Wang[1*], **Jiafeng Xie**[1], **Xiye Chen**[2], **Guozheng Li**[1] and **Wei Li**[3]
[1]School of Computer Science and Engineering, Southeast University
[2]Nanjing University of Finance & Economics
[3]Beijing Institute of Computer Technology and Application
{pwang, xjf, gzli}@seu.edu.cn, chemistrymaths2002@gmail.com, 418857603@qq.com

Abstract

Recent work for extracting relations from texts has achieved excellent performance. However, existing studies mainly focus on simple relation extraction, and these methods perform not well on overlapping triple problem because the tags of shared entities would conflict with each other. Especially, overlapping entities are common and indispensable in Chinese. To address this issue, this paper proposes PasCore, which utilizes a global pointer annotation strategy for Chinese overlapping relation extraction. PasCore first obtains the sentence vector via general pre-training model encoder, and uses classifier to predict relations. Subsequently, it uses global pointer annotation strategy for head entity annotation, which uses global tags to label the start and end positions of the entities. Finally, PasCore integrates the relation, head entity and its type to mark the tail entity. Furthermore, PasCore performs conditional layer normalization to fuse features, which connects all stages and greatly enriches the association between relations and entities. Experimental results on both Chinese and English real-world datasets demonstrate that PasCore outperforms strong baselines on relation extraction and, especially, shows superior performance on overlapping relation extraction.

1 Introduction

Relation extraction aims to obtain semantic relations between entities from text, which plays an important role in constructing knowledge graphs [Hogan *et al.*, 2020], question answering [Mohammed *et al.*, 2018] and recommender systems [Wang *et al.*, 2019]. Recent studies have made great progress in simple relation extraction [Miwa and Bansal, 2016; Lin *et al.*, 2016], but these methods perform poorly on overlapping relation extraction because BIO (short for Beginning, Inside, and Outside) tags cannot mark overlapping entities simultaneously. However, entities are often shared between triples in real world, which can be summarized into four patterns as shown in Figure 1. SEP (*SingleEntityPair*) represents only one entity pair exists in a sentence. When multiple triples exist, following patterns could happen: NEO (*NoEntityOverlap*), indicating no entity overlaps; SEO (*SingleEntityOverlap*), indicating only one entity overlaps; EPO (*EntityPairOverlap*), indicating both head entity and tail entity have overlapped. Intersection represents the triples hold multiple overlapping types in one sentence.

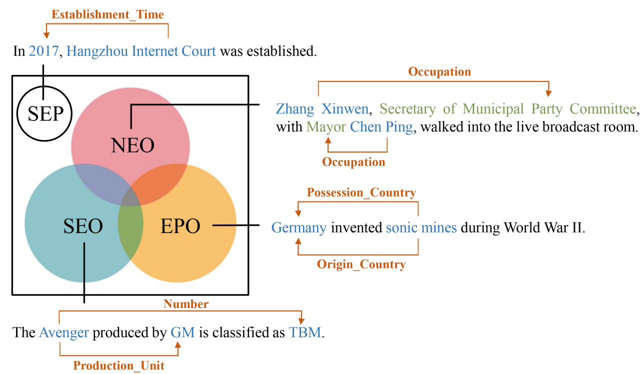

Figure 1: Four types of overlapping relations.

The frequency of overlapping entities varies for different languages, and shared entities are especially common and complicated in Chinese. To verify this phenomenon, this paper compares two Chinese real-world datasets, DuRED and ICRED, with English datasets NYT [Riedel *et al.*, 2010]. First, we find that Chinese datasets have a larger proportion of samples with overlapping entities, which conforms to a long-tailed distribution. Second, compared with English, the Chinese dataset is featured by (1) generally longer sentences, (2) more complex types of entities and relations, and (3) a larger number of entities and relations. For example, in a sentence: *Wang Qing, chief analyst of Oriental Jincheng*, it contains three overlapping entities, namely, *Wang Qing*, *Oriental Jincheng* and *chief analyst of Oriental Jincheng*. At the same time, it holds both SEO and EPO overlapping patterns, which demonstrates that Chinese is challenging for overlapping relation extraction.

Traditional relation extraction methods often use pipeline to first extract entities and then classify candidate entity pairs [Hoffmann *et al.*, 2011]. However, these methods ignore the association between them, and easily cause cumula-

*Corresponding author

tive errors [Li and Ji, 2014]. Moreover, joint relation model integrates the associations between entities and relations, which has achieved better performance in recent years [Tan *et al.*, 2019]. However, overlapping entities cannot be labeled well under these methods via the BIO tagging strategy.

To extract overlapping triples, CopyRE utilizes sequence-to-sequence model to avoid tagging conflict [Zeng *et al.*, 2018]. However, it only copies the last token of the entity and cannot resolve the situation where entity contains multiple tokens. GraphRel uses graph convolutional networks (GCNs) to model text as relational graphs [Fu *et al.*, 2019]. ETLSpan applies a span-based marking strategy to model the semantic relevance [Yu *et al.*, 2020]. Although these studies have made a good improvement, they all treat relations as discrete labels, making relation classification difficult: (1) The matching between entities is prone to a large number of negative cases; (2) When the same entity appears in multiple triples, the classifier is prone to confuse. We noticed that CasRel utilizes a two-stage cascading binary annotation framework to first label the head entity, and then label the relation and the tail entity simultaneously [Wei *et al.*, 2020]. It alleviates the problem of too many entity pair combinations, but ignores the interaction between features. In other words, the entity type features are not considered well in the subsequent element prediction. At the same time, CasRel combines multiple relations with the same head and tail entities into a string, and essentially performs single relation extraction. Therefore, in fact, CasRel does not perform well on Chinese overlapping relation extraction, and fails to achieve the claimed performance.

To address these issues, we propose PasCore, a simple and effective relation extraction model which uses global pointer annotation strategy [Su *et al.*, 2022] for Chinese overlapping relation extraction. It transforms the relation extraction into identifying relations, head entities and tail entities. First, the sentence vector obtained by pre-trained language model (PLM) encoders [Devlin *et al.*, 2018; Sun *et al.*, 2019] is used to predict the relation. Subsequently, PasCore utilizes the previous output as precondition to label the head entity and tail entity. The global pointer in entity annotation stages marks the start and end positions of the entities uniformly. The advantages of this strategy lie in: (1) It uses the idea of global normalization to label entities, and can effectively identify all kinds of entities with high accuracy; (2) Overlapping entities can be extracted via the pairs of different distances based on the global tags. However, BIO tagging strategy cannot deal with the shared entities because they would conflict with each other; (3) The order of relation prediction first and then entity annotation can make full use of features of relation and the entity type. Experiments show that PasCore outperforms state-of-the-art methods on Chinese real-world datasets and is competitive on English datasets, especially achieving excellent distinguished performance on Chinese overlapping relation extraction.

2 Related Work

Recent studies of relation extraction are mainly based on deep learning methods. Recurrent neural network (RNN) and convolutional neural network (CNN) were early utilized to solve the relation extraction problem [Zeng *et al.*, 2014]. Subsequently, to enlarge the distinction between relations, new loss function and attention mechanism were applied [Wang *et al.*, 2016]. Furthermore, Miwa and Bans adopted bidirectional LSTM and tree LSTM to model entities and sentences [Miwa and Bansal, 2016]. Lin et al. applied a sentence-level attention mechanism to assign weights to sentences [Lin *et al.*, 2016] . To reduce the noise in datasets, Su et al. tried embedding textual relations with global relation statistics [Su *et al.*, 2017]. Besides, reinforcement learning and token-level remote supervision were adopted on large-scale datasets [Liu *et al.*, 2018]. Afterwards, Wang et al. designed a graph model to transform joint extraction into a directed graph [Wang *et al.*, 2018]. Tan et al. proposed TME to adaptively discover triples via the ranking of the transfer mechanism [Tan *et al.*, 2019] .

Traditional relation extraction uses pipeline methods to first extract entities and then classify candidate entity pairs [Hoffmann *et al.*, 2011]. However, it completely separates the two subtasks, ignoring the interaction between them and causing errors to accumulate [Li and Ji, 2014]. On the other hand, joint extraction model has achieved good results. Bekoulis et al. designed a joint neural model to extract entities and relations at the same time [Bekoulis *et al.*, 2018]. However, these methods cannot extract overlapping entities well due to the BIO tags conflict.

To deal with the problem of entity overlapping, CopyRE utilizes copy mechanism to avoid overlapping BIO tags [Zeng *et al.*, 2018]. Besides, GraphRel uses a stacked BiLSTM sentence encoder and GCN dependency parsing encoder to extract hidden features of tokens, and decomposes entity pairs into token pairs for prediction [Zeng *et al.*, 2019]. In addition, such method further applied reinforcement learning to improve the effect of CopyRE. Yu et al. proposed an end-to-end sequence tagging framework ETLSpan, which applies a span-based tagging strategy to jointly extract entities and relations [Yu *et al.*, 2020]. However, these methods all treat relations as discrete labels, which causes a large number of negative entity pairs and increases the burden of the relation classifier.

To further effectively extract overlapping triples, Wei et al. proposed CasRel [Wei *et al.*, 2020], a two-stage cascading binary labeling framework, which greatly improves the performance of model on overlapping relation extraction. Besides, Wang et al. proposed a single-stage joint extraction model TPLinker and a novel handshake marking scheme [Wang *et al.*, 2020]. Recently, Ren et al. used a global feature-oriented relational triple extraction model while they cannot get strong performance on more complex extraction [Ren *et al.*, 2021] .

3 Methodology

3.1 Problem Statement

Overlapping relation extraction refers to extracting multiple triples from text where the same entities are shared in one sentence. Given a sentence S, an entity set $E = \{e_1, e_2, \ldots, e_M\}$ and a predefined relation set $R = \{r_1, r_2, \ldots, r_N\}$, all overlapping triples $T =$

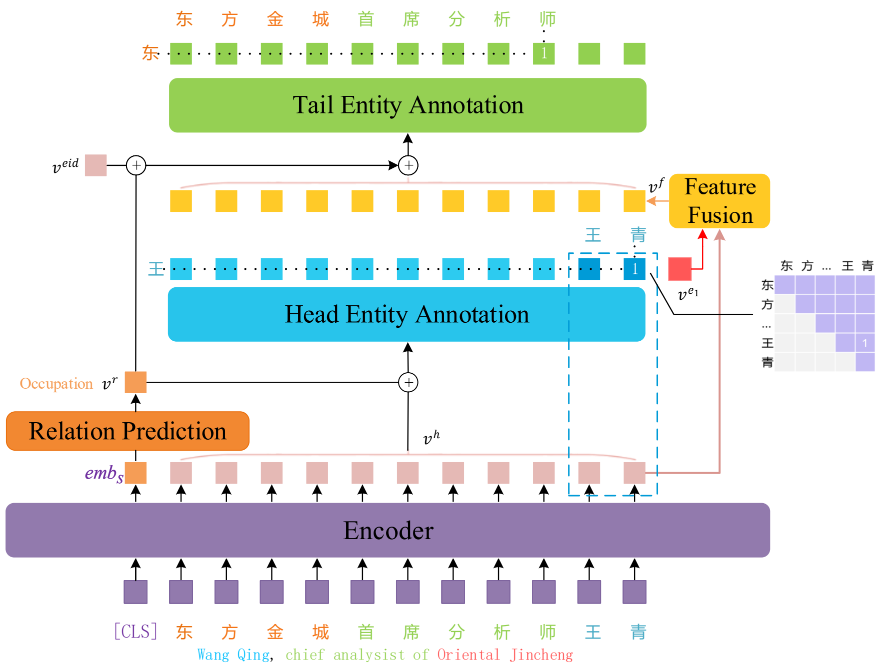

Figure 2: Framework of PasCore.

$\{(e_1, r_1, e_2), \ldots, (e_i, r_k, e_j)\}$ need extracting, where M and N represent the number of entities in sentence and all relations, respectively. $e_i, e_j \in E$, $r_k \in R$. This paper mainly focuses on Chinese overlapping relation extraction, namely, given a sentence S in Chinese, a model can extract all triples contained with overlapping entities simultaneously.

3.2 Framework

The framework of PasCore is depicted in Figure 2. PasCore mainly consists of three stages, namely, relation prediction stage, head entity annotation stage and tail entity annotation stage.

First, for the input sentence, PasCore uses PLMs to encode and obtains the sentence vector and the word vectors. Then, the relation classifier uses such encoders for relation prediction. After predicting the relations that exist in the sentence, PasCore uses the word vectors with relations to mark head entities via global pointer annotation strategy. Subsequently, PasCore fuses the head entity vector with word vectors via conditional layer normalization. Finally, the integrated feature will be cascaded with the entity type feature and the relation feature, to assist in tagging tail entities.

3.3 Encoder

In our work, two pre-training models BERT [Devlin *et al.*, 2018] and ERNIE [Sun *et al.*, 2019] are used to evaluate the effectiveness of the proposed framework.

BERT is a multi-layer bidirectional Transformer based language representation model. It is designed to learn deep representations by jointly conditioning on both left and right context of each word , and it has recently been proven surprisingly effective in many downstream tasks [Zhong *et al.*, 2019].

ERNIE is a pre-trained semantic representation model based on transformer. Compared with BERT, it can capture more semantic information on the Chinese corpus, because it introduces multi-source Chinese knowledge and models the combined semantics of words, improving the versatility and scalability. Let $X = [x_1, x_2, \ldots, x_n]$ represent the input, where x_i represents the vector representation of each token, composed of token embedding, sentence embedding, position embedding and task embedding. ERNIE contains F transformer encoders, denoted as $Trans(x)$ as follows.

$$h_i = Trans(h_{i-1}), i \in [1, F] \tag{1}$$

Suppose that the sentence contains n tokens, the vector representation of $n+1$ tokens including $[CLS]$ will eventually be obtained. $[CLS]$ represents the embedding of the entire sentence $emb_S = h_F[0]$, while the embedding of the i-th token is represented as $v_i^h = h_F[i]$.

3.4 Relation Prediction

The relation prediction layer extracts possible relations in the sentence according to the sentence vector obtained by the PLM encoder. For the sentence vector emb_S, the score of the relation o^r is calculated by Equation 2.

$$o^r = W_r \cdot emb_S + b_r \tag{2}$$

where $W_{(\cdot)}$ represents the trainable weight, and $b_{(\cdot)}$ represents the bias. Let o_i^r represent the score of the i-th relation. In order to obtain the probability of each relation, PasCore performs sigmod function on all relations.

$$p(r_i|S) = \frac{1}{1 + e^{-o_i^r}} \tag{3}$$

PasCore sets a threshold δ_r, and only keeps the relation whose probability exceeds the threshold.

3.5 Head Entity Annotation

PasCore concatenates v^h with the relation vector v^r to obtain the input for head entity annotation stage, which is denoted as $X^{e_1} = [[v_1^h, v^r], [v_2^h, v^r], \ldots, [v_n^h, v^r]]$. Then, it uses one classifier to identify the start and end positions of the head entity, which has a global view.

$$p_i^{e_1^s, e_1^e} = \sigma(W_{e_1^{s,e}} \cdot x_i^{e_1} + b_{e_1^{s,e}}) \tag{4}$$

where $p_i^{e_1^s, e_1^e}$ represents the probability that the i-th token is the spatial location of a head entity. Besides, PasCore sets a threshold δ_{e_1}. When the probability exceeds it, we set the tag to 1. σ is the sigmoid function. During training, the likelihood function is as follows.

$$p(e_1|r, S) = \prod_{t \in \{e_1^{s,e}\}} \prod_{i=1}^{n} (p_i^t)^{I(y_i^t=1)} (1 - p_i^t)^{I(y_i^t=0)} \tag{5}$$

where n represents the length of sentence. If z is true, $I(z) = 1$, otherwise, $I(z) = 0$. When the i-th token is the global position of the head entity, $y_i^{e_1^{s,e}} = 1$.

3.6 Feature Fusion

For the head entity, we obtain its corresponding embedding via the encoder. Specifically, let $v^h_{e_1^{s,e}}$ represent the embeddings at the spatial position of the head entity, respectively. We use reduced dimension $v^h_{e_1^{s,e}}$ to represent the head entity vector v^{e_1}.

Subsequently, PasCore uses conditional layer normalization to fuse v^{e_1} with v^h to obtain v^f. Let the input of the l-th layer be z^l, then the mean μ_l and variance σ_l^2 are calculated as follows.

$$\mu_l = \frac{1}{n_l} \sum_{i=1}^{n_l} z_i^l \tag{6}$$

$$\sigma_l^2 = \frac{1}{n_l} \sum_{i=1}^{n_l} (z_i^l - \mu_l)^2 \tag{7}$$

where n_l is the number of neurons. Then, output $\tilde{z}^l$ is obtained via conditional layer normalization.

$$\tilde{z}^l = \frac{z^l - \mu_l}{\sqrt{\sigma_l^2 + \epsilon}} f_\lambda(x) + f_\beta(x) \tag{8}$$

where f_λ and f_β are two conversion functions using full connection, representing zoom factors and translation factors, respectively. ϵ is a small constant used to smooth.

3.7 Tail Entity Annotation

Consequently, the head entity type vector v^{eid}, v^r and v^f are concatenated to form $X^{e_2} = [[v_1^f, v^r, v^{eid}], [v_2^f, v^r, v^{eid}], \ldots, [v_n^f, v^r, v^{eid}]]$ as the input. Similarly, PasCore identifies the start and end positions of the tail entity as follows.

$$p_i^{e_2^{s,e}} = \sigma(W_{e_2^{s,e}} \cdot x_i^{e_2} + b_{e_2^{s,e}}) \tag{9}$$

where $p_i^{e_2^{s,e}}$ represents the probability that the i-th token is the spatial location of a tail entity. At the same time, threshold δ_{e_2} is set, and only when the probability is greater than it, sets the binary tag to 1. During training, the likelihood function is as follows.

$$\begin{aligned} &p(e_2|e_1, r, S) \\ &= \prod_{t \in \{e_2^{s,e}\}} \prod_{i=1}^{n} (p_i^t)^{I(y_i^t=1)} (1 - p_i^t)^{I(y_i^t=0)} \end{aligned} \tag{10}$$

where n is the length of the sentence. When the i-th token is the global position of the tail entity, $y_i^{e_2^{s,e}} = 1$.

Dataset	DuRED	ICRED	NYT24
Samples	182,196	18,023	66,195
Triples	325,940	79,482	104,339
Entity Type	24	16	4
Relation Type	43	57	24
Average Text Length	68	117	37

Table 1: Statistics of DuRED, ICRED and NYT24 datasets.

3.8 Loss Function

PasCore processes the training data into a set of multiple sentences with their triples. Let D denote the number of sentences, and T_i denote the triples of the i-th sentence. The loss function L is expressed as Equation 11.

$$\begin{aligned} L = -\sum_{i=1}^{D} \Bigg[&\sum_{r \in T_i} \log p(r|S_i) + \sum_{e_1 \in T_i|r} \log p(e_1|r, S_i) \\ &+ \sum_{e_2 \in T_i|e_1, r} \log p(e_2|e_1, r, S_i) \Bigg] \end{aligned} \tag{11}$$

4 Experiments

To evaluate the proposed method, we conduct experiments on two Chinese datasets [1] and one general English dataset. Then we analyze the results comprehensively as well.

4.1 Datasets and Settings

Due to lack of Chinese overlapping relation extraction datasets, this paper constructs and releases two new datasets DuRED and ICRED for evaluation. DuRED and ICRED are based on the open datasets of relation extraction tasks in Baidu 2020 Language and Intelligent Technology Competition and Intelligent Computing Platform Competition[2]. Based on these raw real-world datasets, we first remove inapplicable multiple relations and wrong triples via manually fine-grained inspection. Secondly, we mine entity types through datasets and introduce external prior knowledge to complete the triples. Finally, we perform data augmentation and obtain the final datasets. In addition, there are many overlapping triples in other languages (such as English), and we choose the popular English dataset NYT24 for evaluating the generality of PasCore.

[1]https://github.com/seukgcode/pasCore

[2]http://lic2020.cipsc.org.cn

Type		DuRED			ICRED			NYT24		
		Train	Valid	Test	Train	Valid	Test	Train	Valid	Test
Overlapping Type	SEP	89,962	10,089	12,096	2,145	300	560	36,868	3,285	3,244
	NEO	56,269	6,244	7,536	10,563	1,440	3,015	21,504	2,053	2,258
	SEO	34,620	3,902	4,685	8,703	1,153	2,48	16,255	1,718	1,757
	EPO	4,511	498	593	2,110	258	583	16,255	1,718	1,757
Number of Triples	$N = 1$	89,962	10,089	12,096	2,145	300	560	36,868	3,285	3,244
	$N = 2$	31,735	3,500	4,237	1,349	185	389	12,058	1,032	1,045
	$N = 3$	10,713	1,178	1,384	2,241	335	672	3,663	323	312
	$N = 4$	6,464	745	861	1,853	257	542	2,618	243	291
	$N = 5$	2,699	344	385	1,163	153	305	517	55	51
	$N \in [6, 10]$	4,116	421	588	3,498	456	989	460	60	51
	$N \in [11, 20]$	527	54	79	434	52	115	11	2	6
	$N > 20$	15	2	2	25	2	3	0	0	0
Sum		146,231	16,333	19,632	12,708	1,740	3,575	56,195	5,000	5,000

Table 2: Statistics of dataset division, overlapping relation distribution and triple distribution.

Model	DuRED			ICRED			NYT24		
	P	R	F1	P	R	F1	P	R	F1
NovelTagging	61.6	56.3	58.8	49.5	34.1	40.4	61.5	41.1	49.5
CopyRE	62.4	62.6	62.5	48.5	41.9	45.0	61.0	56.6	58.7
GraphRel	66.5	63.4	64.9	52.2	42.7	46.9	63.9	60.0	61.9
ETL-Span	72.5	69.1	70.8	53.7	47.9	50.7	84.9	72.3	78.1
CasRel	74.6	75.0	74.8	55.0	55.0	55.0	89.7	89.5	89.6
GRTE	73.8	74.5	74.1	55.5	54.8	55.1	**92.9**	**93.1**	**93.0**
PasCore (BERT-base)	74.8	76.0	75.4	59.4	60.8	60.1	91.8	92.6	92.2
PasCore (ERNIE-base)	**75.1**	**76.2**	**75.7**	**60.3**	**61.1**	**60.7**	-	-	-
PasCore (ERNIE-en)	-	-	-	-	-	-	92.2	92.8	92.5

Table 3: Main experimental results on DuRED, ICRED and NYT24 datasets. P, R, F1 denote precision, recall, and F1 scores, respectively.

The details of datasets are shown in Table 1. The peak of text length distribution in DuRED lies between 50 and 100, and the maximum length does not exceed 300. Such peak in ICRED is around 100, and the maximum length is less than 500. The long-tail interval in ICRED is longer, which makes relation extraction more difficult. In both two datasets, the distribution of entity types and relations is extremely unbalanced. Furthermore, Table 2 reports the details of the division, overlap relation, and triple distribution of the two Chinese datasets and English dataset NYT24.

PasCore is implemented by PaddlePaddle[3], and uses AdamW [Loshchilov and Hutter, 2018] to optimize weights. In experiments, we use BERT-base for all datasets. Two versions of ERNIE are utilized, i.e., ERNIE-base which are experted in Chinese, another namely ERNIE-en in English.

4.2 Baselines

For comparison, we reproduce the following baselines: NovelTagging [Zheng *et al.*, 2017], which uses annotation strategy to combine entities and relations; CopyRE [Zeng *et al.*, 2018], an end-to-end model which uses copy mechanism to jointly extract relations; GraphRel [Fu *et al.*, 2019], a joint relation extraction model based on GCN network; ETLSpan [Yu *et al.*, 2020], which hierarchically decodes triples to model the semantic relevance; CasRel [Wei *et al.*, 2020], a two-stage cascading binary labeling framework which extracts head entities, tail entities and relations step by step; GRTE [Ren *et al.*, 2021], a global feature-oriented relational triple extraction model based on table filling.

[3]https://www.paddlepaddle.org.cn

4.3 Main Experimental Results

Table 3 reports the main experimental results. First, on DuRED dataset, PasCore outperforms all baselines on precision (P), recall (R) and F1 scores. When ERNIE-base is used as the encoder, PasCore achieves the best result, which is 0.9% higher than the F1 score of CasRel. Second, on ICRED dataset, the best performance is still obtained by PasCore. When using the BERT-base as encoder, PasCore also surpasses all baselines and is 5% higher than the F1 score of GRTE. When using the ERNIE-base as the encoder, the F1 score is 0.6% higher than that of BERT-base, reflecting the excellent and stable relation extraction performance of PasCore. Third, on English dataset NYT24, the performance of PasCore is still very competitive and is closed to GRTE. Note that, GRTE performs better on English dataset, probably because it employs a global feature-oriented table filling method, which is more suitable for English. However, for PasCore, its performance gap on different PLMs is very small, which reflects the fact that PasCore is less sensitive

Dataset	Model	SEP	NEO	SEO	EPO	$N=1$	$N=2$	$N=3$	$N=4$	$N=5$	$N \in [6,10]$	$N \in [11,20]$	$N > 20$
DuRED	NovelTagging	61.5	44.3	79.5	49.2	61.5	51.3	56.4	57.7	64.8	66.2	59.1	18.2
	CopyRE	65.2	49.3	82.0	56.4	65.2	56.6	60.3	61.0	67.3	67.5	62.3	34.8
	GraphRel	67.9	51.5	84.4	56.6	67.9	59.6	62.3	62.9	70.1	69.4	63.0	27.0
	ETL-Span	72.6	59.9	88.2	59.3	72.6	68.7	68.7	69.3	74.1	72.4	68.4	31.4
	CasRel	75.9	66.3	89.3	64.1	75.9	73.5	73.7	75.0	75.7	74.9	70.8	42.8
	GRTE	73.4	62.6	89.8	63.4	73.4	74.1	73.2	**76.3**	73.7	71.6	70.3	33.6
	PasCore (BERT-base)	76.4	65.5	90.8	66.7	76.4	73.4	73.3	73.6	76.8	75.9	70.7	46.3
	PasCore (ERNIE-base)	**76.9**	**67.0**	**91.0**	**67.2**	**76.9**	**74.4**	**73.9**	75.2	**78.4**	**76.2**	**73.0**	**47.5**
ICRED	NovelTagging	51.5	39.1	46.5	49.2	51.1	48.3	60.1	48.2	38.5	33.2	20.3	9.76
	CopyRE	51.6	46.9	46.2	32.4	51.6	47.3	62.9	51.8	46.7	38.8	28.6	11.0
	GraphRel	53.8	48.7	48.1	29.5	53.8	51.6	62.6	53.6	49.5	41.3	29.2	9.6
	ETL-Span	55.8	53.9	51.1	40.1	55.8	52.6	65.3	55.8	51.1	45.9	38.2	26.3
	CasRel	63.3	60.3	52.3	41.2	63.3	61.1	**70.6**	61.1	54.9	50.9	33.3	34.6
	GRTE	61.9	64.1	52.5	40.7	61.9	63.5	66.3	60.2	56.2	53.8	42.8	42.7
	PasCore (BERT-base)	63.8	61.5	58.1	42.9	63.8	61.7	70.2	61.4	55.2	53.9	43.1	34.8
	PasCore (ERNIE-base)	**64.1**	**67.7**	**56.8**	**46.4**	**64.1**	**64.9**	70.4	**62.9**	**58.2**	**59.8**	**46.8**	**45.8**
NYT24	NovelTagging	62.2	34.1	48.5	44.2	62.2	42.9	50.1	47.1	36.8	-	-	-
	CopyRE	67.1	63.4	52.6	21.7	67.1	58.6	52.0	53.6	30.0	-	-	-
	GraphRel	71.0	64.1	50.8	37.8	71.0	61.5	57.4	55.1	41.3	-	-	-
	ETL-Span	76.2	66.5	50.9	44.3	76.2	72.2	68.4	52.3	49.1	-	-	-
	CasRel	88.2	90.5	91.4	92.0	88.2	90.3	91.9	94.2	83.7	-	-	-
	GRTE	**90.8**	**93.6**	**94.5**	**95.0**	**90.8**	**93.7**	**94.4**	**96.2**	**93.4**	-	-	-
	PasCore (BERT-base)	88.9	90.6	94.0	93.4	88.9	91.6	94.1	95.7	92.2	-	-	-
	PasCore (ERNIE-en)	89.4	91.1	94.2	93.8	89.4	92.1	94.3	95.9	92.6	-	-	-

Table 4: Experimental results (F1 scores) of different numbers of triples on DuRED, ICRED and NYT datasets. Note that, on NYT24 dataset, $N = 5$ represents $N \geq 5$.

to PLMs and more versatile or generic. Moreover, CasRel performs better on DuRED and relatively poorly on ICRED. The reason is that CasRel combines multiple relations with the same head and tail entities into a string, and treats it as a single relation extraction, resulting in a poor performance on ICRED, which contains a lot of overlapping entities.

For different overlapping patterns, the left part of Table 4 reports F1 scores of all models on four overlapping types. It can be seen that, on DuRED and ICRED, PasCore achieves the best performances on all overlapping types. For ICRED, most models have the highest F1 scores on SEP, while for DuRED, the type is SEO. The reason is that a model should not only fit simple cases, but also needs to coordinate all overlapping types to maintain consistency. Besides, the distribution of each type could affect the performance. For English dataset NYT24, the F1 scores of SEO and EPO appear the highest alternately. Meanwhile, PasCore is very close to GRTE in English overlapping relation extraction, which demonstrates the effectiveness of PasCore in English.

In terms of different numbers of triples, the right part of Table 4 reports the F1 scores of PasCore and baselines according to N (the number of triples) in datasets. We can observe that PasCore (ERNIE-base) achieves the best performances in almost all cases, but is slightly less effective than GRTE (on DuRED and $N = 4$) and CasRel (on ICRED and $N = 3$). Especially, when $N \geq 5$, PasCore (ERNIE-base) has more significant performance advantages over other baselines. In addition, we can see that in DuRED, the performances of PasCore fluctuate less that of ICRED. This is mainly because the ICRED is only one-tenth of the DuRED, which makes the model cannot fully fit the distribution of triples, and fluctuates larger. On the English dataset NYT24, the F1 score of PasCore on each number of triples is only lower than GRTE, the gap between PasCore and GRTE is marginal.

Model	Test P	Test R	Test F1
PasCore (ERNIE-base)	**60.3**	**61.1**	**60.7**
-RelType	48.8	40.5	44.4
-HEntText	52.2	55.5	53.8
-HEntType	56.7	59.5	58.0
-All	39.1	31.4	34.8

Table 5: Ablation experimental results on ICRED dataset.

4.4 Ablation Experimental Results

We conduct ablation experiments with PasCore (ERNIE-base), and the results on ICRED dataset are shown in Table 5. -RelType represents removing v^r in feature vector. -HEntText represents removing v^{e_1} in feature vector. -HEntType denotes removing v^{eid} in feature vector.

It can be seen that after removing the relation vector, F1 scores drop dramatically. It means that the relation plays a great role in entity annotation. Another observation is that both the head entity and the head entity type can improve the performance of PasCore. After removing these two, F1 scores both drop. This is because PasCore uses the normaliza-

Model	1. relation			2. relation→entity			3. relation+head entity			4. relation+head entity→tail entity		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Gold	-	-	-	82.6	83.8	83.2	-	-	-	87.0	89.4	88.1
Predict	85.9	89.7	87.8	67.6	68.4	68.0	72.9	76.1	74.5	77.8	79.4	78.6

Table 6: Error Accumulation Analysis on ICRED dataset.

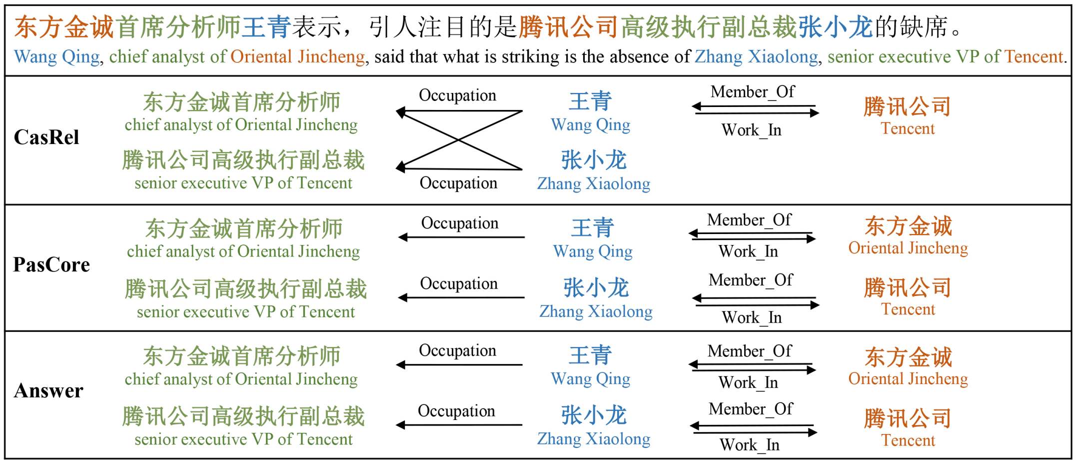

Figure 3: Experimental case analysis.

tion method to integrate them into the feature vector, which further promotes the performance. After removing all the features mentioned above, F1 scores drop significantly, fully verifying the importance of these features.

4.5 Error Accumulation Analysis

Considering that PasCore is a pipeline method, we carry out experiments on error accumulation between modules. In Table 6, experiments *1. relation* and *3. relation+head entity* represent gold and predicted experimental results. Experiments *2. relation→entity* and *4. relation+head entity→tail entity* use arrows to show that the input is gold or the results predicted in the previous module and the experimental results in the next module.

Compared with experiment 1 and experiment 2, it can be seen that when the relation predicted by the model is used as input, the F1 score of the predicted head and tail entities is not significantly reduced. Similarly, in experiments 3 and 4, the use of the gold and predicted relation with the prediction of the head entity to the tail entity does not cause significant performance drops. This indicates that the error of the upstream module of PasCore will not cause disastrous influence to the subsequent modules.

4.6 Case Study

To further analyze the advantages of PasCore, we also study the cases in Figure 3, which contains three overlapping patterns: NEO, SEO and EPO.

It can be seen that the results of PasCore are consistent with the answer, while for CasRel, the triples related to *Wang Qing* and *Zhang Xiaolong* are confused. This is mainly because CasRel first extracts the head entity, and then extracts the relations and tail entities at the same time, making the matching prone to errors. Specifically, after identifying *Wang Qing*, the relation *Work_in* and the entity *Tencent* are extracted simultaneously, resulting in a false match.

In contrast, PasCore firstly performs relation prediction. After *Work_in* is predicted in success, *Wang Qing* and *Zhang Xiaolong* are marked via global pointer annotation strategy. Take the former as an example, PasCore then fuses the features of *Work_in*, *Wang Qing* and its type, and finally marks the entity *Oriental Jincheng* successfully. This is because PasCore extracts triples in strict accordance with the order of relation prediction and entity annotation, and the features obtained from the previous stage are also retained to the latter stages, which enriches the association between relations and entities and ensures the correct matching of the entity pair.

5 Conclusion and Future Work

To extract overlapping relations especially in Chinese, this paper proposes PasCore, a relation extraction model based on global pointer annotation strategy. It transforms the relation extraction into relation predication, head entity annotation and tail entity annotation three stages in order. Besides, in order to enrich the feature representation, the output of each stage will be passed to the next stage as precondition. Experiments show that PasCore achieves excellent performances on both Chinese real-world datasets and English dataset.

In the future, we plan to explore a more effective method to integrate relations into entity annotation, and integrate the head entity into tail entity annotation. At the same time, we also plan to generalize PasCore to more languages.

References

[Bekoulis *et al.*, 2018] Giannis Bekoulis, Johannes Deleu, Thomas Demeester, and et al. Joint entity recognition and relation extraction as a multi-head selection problem. *Expert Systems with Applications*, 114(12):34–45, 2018.

[Devlin *et al.*, 2018] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[Fu *et al.*, 2019] Tsujui Fu, Penghsuan Li, and Weiyun 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*, pages 1409–1418, 2019.

[Hoffmann *et al.*, 2011] Raphael Hoffmann, Congle Zhang, Xiao Ling, and et al. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 541–550, 2011.

[Hogan *et al.*, 2020] Aidan Hogan, Eva Blomqvist, Michael Cochez, and et al. Knowledge graphs. *arxiv*, abs/2003.02320, 2020.

[Li and Ji, 2014] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics*, pages 402–412, 2014.

[Lin *et al.*, 2016] Yankai Lin, Shiqi Shen, Zhiyuan Liu, and et al. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 2124–2133, 2016.

[Liu *et al.*, 2018] Tianyi Liu, Xinsong Zhang, Wanhao Zhou, and et al. Neural relation extraction via inner-sentence noise reduction and transfer learning. In *Proceedings of the 56th Conference on Empirical Methods in Natural Language Processing*, pages 2195–2204, 2018.

[Loshchilov and Hutter, 2018] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arxiv*, abs/1711.05101, 2018.

[Miwa and Bansal, 2016] Makoto Miwa and Mohit Bansal. End-to-end relation extraction using lstms on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1105–1116, 2016.

[Mohammed *et al.*, 2018] Salman Mohammed, Peng Shi, and Jimmy Lin. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of the 16th North American Chapter of the Association for Computational Linguistics*, pages 291–296, 2018.

[Ren *et al.*, 2021] Feiliang Ren, Longhui Zhang, Shujuan Yin, Xiaofeng Zhao, Shilei Liu, Bochao Li, and Yaduo Liu. A novel global feature-oriented relational triple extraction model based on table filling. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 2646–2656, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.

[Riedel *et al.*, 2010] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. *Machine Learning and Knowledge Discovery in Databases*, 6323(3):148–163, 2010.

[Su *et al.*, 2017] Yu Su, Honglei Liu, Semih Yavuz, and et al. Global relation embedding for relation extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 820–830, 2017.

[Su *et al.*, 2022] Jianlin Su, Ahmed Murtadha, Shengfeng Pan, Jing Hou, Jun Sun, Wanwei Huang, Bo Wen, and Yunfeng Liu. Global pointer: Novel efficient span-based approach for named entity recognition. *arXiv preprint arXiv:2208.03054*, 2022.

[Sun *et al.*, 2019] Yu Sun, Shuohuan Wang, Yukun Li, and et al. Ernie: Enhanced representation through knowledge integration. *arxiv*, abs/1904.09223, 2019.

[Tan *et al.*, 2019] Zhen Tan, Xiang Zhao, Wei Wang, and et al. Jointly extracting multiple triplets with multilayer translation constraints. In *Proceedings of the 33rd Association for the Advance of Artificial Intelligence Conference on Artificial Intelligence*, pages 7080–7087, 2019.

[Wang *et al.*, 2016] Linlin Wang, Zhu Cao, Gerard De Melo, and et al. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, pages 1298–1307, 2016.

[Wang *et al.*, 2018] Shaolei Wang, Yue Zhang, Wanxiang Che, and et al. Joint extraction of entities and relations based on a novel graph scheme. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4461–4467, 2018.

[Wang *et al.*, 2019] Xiang Wang, Xiangnan He, Yixin Cao, and et al. Kgat: knowledge graph attention network for recommendation. In *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining International Conference*, pages 950–958, 2019.

[Wang *et al.*, 2020] Yucheng Wang, Bowen Yu, Yueyang Zhang, and et al. Tplinker: Single-stage joint extraction of entities and relations through token pair linking. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1572–1582, 2020.

[Wei *et al.*, 2020] Zhepei Wei, Jianlin Su, Yue Wang, and et al. A novel cascade binary tagging framework for relational triple extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1476–1488, 2020.

[Yu *et al.*, 2020] Bowen Yu, Zhenyu Zhang, Xiaobo Shu, and et al. Joint extraction of entities and relations based on a novel decomposition strategy. In *Proceedings of the 24th European Conference on Artificial Intelligence*, pages 2282–2289, 2020.

[Zeng *et al.*, 2014] Daojian Zeng, Kang Liu, Siwei Lai, and et al. Relation classification via convolutional deep neural network. In *Proceedings of the 25th International Conference on Computational Linguistics*, pages 2335–2344, 2014.

[Zeng *et al.*, 2018] Xiangrong Zeng, Daojian Zeng, Shizhu He, and et al. 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*, pages 506–514, 2018.

[Zeng *et al.*, 2019] Xiangrong Zeng, Shizhu He, Daojian Zeng, and et al. Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. In *Proceedings of the 57th Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 367–377, 2019.

[Zheng *et al.*, 2017] Suncong Zheng, Feng Wang, Hongyun Bao, and et al. Joint extraction of entities and relations based on a novel tagging scheme. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, pages 1227–1236, 2017.

[Zhong *et al.*, 2019] Peixiang Zhong, Di Wang, and Chunyan Miao. Knowledge-enriched transformer for emotion detection in textual conversations. *arXiv preprint arXiv:1909.10681*, 2019.