Knowledge-Based Systems 271 (2023) 110550

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

ERGM: A multi-stage joint entity and relation extraction with global entity match

Chen Gao [a], Xuan Zhang [b,c,d,*], LinYu Li [b], JinHong Li [a], Rui Zhu [b,c,d], KunPeng Du [b], QiuYing Ma [b]

[a] School of Information Science and Engineering, Yunnan University, Yunnan 650091, China
[b] School of Software, Yunnan University, Yunnan 650091, China
[c] Key Laboratory of Software Engineering of Yunnan Province, Yunnan 650091, China
[d] Engineering Research Center of Cyberspace, Yunnan 650091, China

ARTICLE INFO

Article history:
Received 14 October 2022
Received in revised form 8 January 2023
Accepted 4 April 2023
Available online 10 April 2023

Keywords:
Joint extraction
Global entity match
Model structure optimization
Triple overlap problem
Knowledge graph

ABSTRACT

Joint entity and relation extraction is a fundamental and important task in the process of building knowledge graphs. At present, many researchers have proposed their own methods to solve this task, but these studies often have some limitations, such as irrelevant relation prediction, and lack of information interaction between the relation and entity. The complex model structure leads to inefficiency and does not make good use of the associations between the various subtasks. We propose a novel lightweight joint extraction model based on a global entity matching strategy. Specifically, the proposed framework contains three components: Relation Extraction Module, Relation Attention Based Entity Recognition Module and Global Entity Pairing Module. The Relation Extraction Module extracts candidate relations in the sentence, and the Relation Attention Based Entity Recognition Module introduces a relation attention mechanism based on the obtained candidate relations to fuse the information of the relations so as to better identify entities in the sentence. Then use entity vector representations to construct an affine transformation-based global entity matching matrix under a specific relation for triple extraction. Our model decomposes entity and relation extraction into three sub-tasks, which greatly simplifies the model structure, and the tasks are interrelated, making full use of the relevant information. In addition, we introduce a negative sampling strategy to alleviate the exposure bias problem of the model. We validate Our model on public dataset, it not only can effectively solve the triple overlap problem but also achieved a significant time performance speedup and effectively reduce memory occupation.

1. Introduction

Joint entity and relation extraction is a subtask of information extraction and one of the necessary steps for building knowledge graphs [1], which has important applications in fields such as machine translation [2], question answering systems [3] and recommender systems [4]. Entity and relation extraction is aim to extract the set of triples (s, r, o) from a given sentence, where r is a predefined set of relations, s and o refer to the subject and object entities, respectively.

Early entity and relation extraction mainly adopts pipeline-based methods that comprise two subtasks, named entity recognition (NER) [5–7] and relation extraction (RE) [8–10]. The entities present in the sentence are first identified using the NER model, and then the RE model is used to classify the entity pairs extracted from the sentence and assign them to a predefined set of relations. This approach simplifies the task flow and makes the operation of each component more flexible, but there are still some problems. There is a lack of interaction between the two subtasks, and the model cannot use the information of the associated tasks. Secondly, there is a lot of redundancy in the entities generated by NER, and not all entity pairs will have relations. Finally, the extraction of the pipeline mode will incur error propagation, the accuracy of the NER model will directly affect the performance of subsequent relation extraction.

In order to alleviate the drawbacks of pipeline methods, researchers have proposed an entity relation extraction method based on joint learning. This method is mainly divided into two methods: joint decoding [11] and parameter sharing [12–14]. Joint decoding mainly combines entity and relation labels, and uses sequence labels to identify relations and entities at the same time. Parameter sharing generally adopts a multi-task learning

* Corresponding author at: School of Software, Yunnan University, Yunnan 650091, China.
E-mail address: zhxuan@ynu.edu.cn (X. Zhang).

https://doi.org/10.1016/j.knosys.2023.110550
0950-7051/

	Texts	Triplets
Normal	The United States president Trump will meet Putin, the president of Russia.	(The United States, President, Trump) (Putin, President, Russia)
SEO	Jack was born in Washington, the capital of the United States.	(Jack, Born in, Washington) (Washington, Capital, the United States)
EPO	Beijing is the capital of China.	(Beijing, Capital, China) (Beijing, Located in, China)

Fig. 1. Examples of Normal, SEO, EPO and SOO.

method, so that NER and RE models share a set of encoding layers that jointly optimize their loss functions. Traditional joint methods [15–18] require feature engineering and are less efficient. With the development of deep learning, many works [11,14,19–21] based on deep neural networks have achieved promising results.

However, the extraction of overlapping triples is still a challenging problem. The triples can be divided into the following three categories according to the type of overlap, namely Normal, Single Entity Overlap (SEO) and Entity Pair Overlap (EPO). Examples are illustrated in Fig. 1, where blue entities indicate ordinary triples, such as "Putin" and "Russia" in Normal. Red entities indicate overlap, such as "Beijing" and "China" in EPO. Recently, many researchers have proposed solutions, which can be roughly divided into two categories: multi-stage learning frameworks and single-stage learning frameworks. The multi-stage learning framework adopts the method of multi-task learning to model multiple subtasks. Similar to [14,22–25], they simplify the structure of the entity relation extraction model and improve the model extraction efficiency to a certain extent, but there is a problem of exposure bias. The exposure bias problem refers to the inconsistency of the data used by the model during the training phase and the prediction phase. During the training phase, each task can be trained using the golden labels. In the prediction stage, the pre-task needs to be performed first, and the label predicted by the pre-task is used as the input of the next task, the input data is not all correct. To alleviate the exposure bias brought by multi-stage learning, several novel joint extraction methods [20] have been proposed. All these methods [14,20,22–25] solve the triple overlap problem to a certain extent, but they face problems such as high complexity and lack of information interaction between relations and entities. Therefore, it is still challenging to identify overlap triples, and the performance of joint extraction models should be further improved.

Entity and relation extraction can be understood as the joint probability of relational triple extraction $\mathrm{p(t \mid X)}$, where t is (s, r, o), s represents the subject entity, o stands for the object entity and r denotes the relations, X is the input sentence. We decompose entity and relation joint extraction into three subtasks, thereby decomposing joint probabilities into conditional probabilities $\mathrm{p(t \mid X) = p(s, r, o \mid e, r, X)p(e \mid X)p(r \mid X)}$, e is the entity, which includes s and o.

The traditional entity and relation pipeline extraction method first identifies the entities in the sentence, and then pairs the entities. This process simplifies the steps of entity and relation extraction, but lacks the interaction between entities and relations. Inspired by pipeline model, we first identify all candidate relations in the sentence. Then, the entities in the sentence are identified. Different from previous models, we do not distinguish between subject and object entities here, nor do we implement entity identification under specific relations, but extract all entities in the sentence at once. This has two advantages, first, our method can effectively simplify the time complexity of the model and only need to perform sequence annotation once to take out all the entities. Second, it greatly reduces the task difficulty of entity recognition and improves the accuracy of entity recognition. Finally, based on the sentence vectors, and the results of entity recognition in the previous step, all entity vector representations are obtained, and entity matching matrices are constructed based on specific relations. Our entity matching matrix effectively incorporates the semantic information of the relations, because the entity-to-entity matching is directly related to the relations. In addition, the number of entities in a sentence is often much smaller than the number of tokens, so the dimensionality of the entity matching matrix is not too large, which avoids data sparsity and improves the effectiveness of model learning.

The main contributions of this paper are as follows:

1. We propose an affine transformation-based approach to construct a joint Entity and Relation extraction framework for Globally Matching entity pair matrices (ERGM) under specific relations. The method greatly simplify the model structure while reducing the irrelevant relation prediction and identifying overlapping triples.
2. We introduce a candidate relation attention mechanism to model the intrinsic connection between entities and relations, which can effectively improve entity recognition in sentences. In addition, we introduce a negative sampling strategy to alleviate the exposure bias problem of the model.
3. We evaluate ERGM on two public datasets. The experimental results show that ERGM is highly competitive with the previous baseline and improves more than 1.5 times in the time performance of the model. PRGC [21] as the fastest baseline model. On the NYT dataset, our training time is 1.51 times faster than PRGC. On the WebNLG dataset, our training time is 1.9 times faster than it.

2. Related work

In this section, we present related work in joint entity and relation extraction, which comprise two types of systems, (1) multi-stage pipeline solutions as well as (2) joint learning approaches, each which we discuss next.

Pipeline-based methods use two tasks for joint entity and relation extraction, named entity recognition and relation extraction. Zelenko et al. [26] proposed the use of kernel methods for relation extraction. Kernel methods use the object in algorithms only via computing a kernel function between a pair of objects. Chan et al. [27] used syntactic-semantic structure for relation extraction to reduce errors in pipeline propagation. In [28], CNN was used to classify relations. Then, Zhou et al. [29] introduced an attention layer based on the BiLSTM model to better encode sentences and improve the performance of relation classification.

However, pipeline-based methods have some limitations and cannot exploit the dependency information between two tasks, so researchers have proposed methods for extraction by joint

learning. Zheng et al. [11] first proposed the use of uniform tokenization methods to solve the multivariate triple extraction problem in sentences. They modeled joint entity and relation extraction as a sequence tagging task, so it is difficult to solve the triple overlap problem. To address the triple overlap problem, Zeng et al. [22] first explicitly defined three specific cases of overlapping triples and proposed a copy-based mechanism to deal with overlapping triples. However, each copy process can only be performed for one vocabulary, and the model needs to perform multiple rounds of copy for overlapping sentences and can only deal with a single token of the entity, so it has limitation. To simplify the model structure, Yu et al. [19] used hierarchical boundary labeling and multi-span decoding algorithms, which can simplify the structure of the task and achieve efficient extraction based on the realization of overlap triples extraction, but the model is based on hierarchical labeling of head entities and cannot solve the entity pair overlapping (EPO) problem. Wei et al. [14] proposed a tail entity recognition method based on relational mapping, they utilized a pre-trained model to model triples as head entities, which can effectively extract overlapping triples. However, this method needs to construct a sequence of relation markers for each head entity, which has high time complexity and space complexity. And because it is necessary to construct a label sequence for all relations, the relations matrix has a sparsity problem, and it is difficult for the model to learn through a small number of positive labels. Wang et al. [20] defined joint entity and relation extraction as a token pair joining problem and used an inverted triangular matrix to alleviate data sparsity. However, since the model is based on a matching matrix constructed from sentence sequence pair global relations and, therefore, there is still a data sparsity problem and the training speed is very slow. To avoid global relations prediction, Zheng et al. [21] proposed a global communication method to match entity pairs, but the method constructed a communication global matrix with high complexity, and the information about the relations was not well integrated into the sentence to guide entity recognition. Xu et al. [30] explicitly introduced relation representation, jointly represent it with entities, and novelty aligned them to identify valid triples. However, the label of the relation is introduced directly and the description of the relation will directly affect the model and will generate noisy information.

Based on the shortcomings of the above methods, we propose a global matching-based entity and relation extraction framework to address the triplet overlap problem. Our model is mainly composed of three parts, namely Relation Extraction Module, Relation Attention Based Entity Recognition Module and Global Entity Pairing Module. The model first performs a multi-label classification task based on the sentence feature vector generated by the encoder to extract possible relations in the sentence, and then uses the relation-based attention mechanism to construct the sentence feature representation under potential relations, and based on this, identify the entities. At the same time, the global entity pair matrix is constructed by affine transformation to determine the head-to-tail matching information of the subject and object entities, which takes into account the performance advantages of the model while improving its effectiveness. Finally, the output information of these three modules is fused to combine the triples existing in the sentence. In addition, the model introduces a negative sample strategy to mitigate the exposure bias problem. Experimental results on public datasets demonstrate the effectiveness of our proposed model.

3. Methodology

In this paper, we propose a three-stage model ERGM. In the first stage, ERGM employs a multi-label classification strategy to detect all potential relations. In the second stage, ERGM proposes a relation attention and adopts a binary labeling strategy to identify the subject and object entities. In the third stage, ERGM uses an affine matrix to model all entity pairs in a sentence under corresponding relations. Briefly, our model contains Encoder Module, Relation Extraction Module, Relation Attention Based Entity Recognition Module and Global Entity Pairing Module as shown in Fig. 2.

Table 1
Symbols and meanings.

Symbols	Meaning
$S = (w_1, w_2, \ldots, w_n)$	Sentence
$R = (r_1, r_2, \ldots, r_k)$	Relation Set
$H = (h_1, h_2, \ldots, h_n)$	Token embedding
$T = (s, r, o)$	Triple
$Avgpool$	Average pooling operation
σ	Sigmoid function
$\mathcal{L}$	Model loss
E	Entity Set
e_{max}	Max length of the entity set in sentence
r_{max}	Max num of the relation in sentence
θ	Threshold
P	Probability

3.1. Task definition

Give a sentence $S = (w_1, w_2, \ldots, w_n)$ and predefined relations $R = \{r_1, r_2, \ldots, r_k\}$, where n is the sentence length and k is the number of relations. The purpose of joint entity and relation extraction is to identify all possible triples $T = \{(s, r, o) \mid s, o \in E, r \in R\}$, where E is the set of entities, s and o are the subject and object entities respectively. All the symbols and their meanings are listed in Table 1.

3.2. Encoder module

The input of REGM is sentence $S = (w_1, w_2, \ldots, w_n)$. First, we pad the sentence to keep a uniform length T for all sentences. Then we employ a pre-trained BERT [31] as sentence encoder to capture the token embedding $H = (h_1, h_2, \ldots, h_n)$ for each token, as shown in formula (1). It is the summation over the corresponding token embedding and positional embedding.

$$H = BERT\,(w_1, w_2, \ldots, w_T) \tag{1}$$

where $h_i \in \mathbb{R}^d$, w_T is the input representation of each token, n is the sequence length after uses WordPiece embedding, d is the embedding dimension.

3.3. Relation extraction module

In generate, sentences contains multiple relations. In Fig. 2, we can see that there are two relations in the sentence, namely "Capital of" and "Located in". Therefore, we apply a multi-label classification strategy to identify all potential relations contained in sentences. For the BERT-based model, given the embedding $h \in \mathbb{R}^{n \times d}$ of a sentence with n tokens, project it into a relation-detection space for multi-label classification, as shown in formula (2) and (3):

$$h^{avg} = \text{Avgpool}(h) \tag{2}$$

$$P_r = \sigma\left(W_r h^{avg} + b_r\right) \tag{3}$$

where $Avgpool$ [32] is the average pooling operation, $W_r \in \mathbb{R}^{d_e \times 1}$, b_r are trainable weight and bias. d_e represents the dimension of the token output by BERT, $h^{avg} \in \mathbb{R}^{d_e}$ represents the sentence

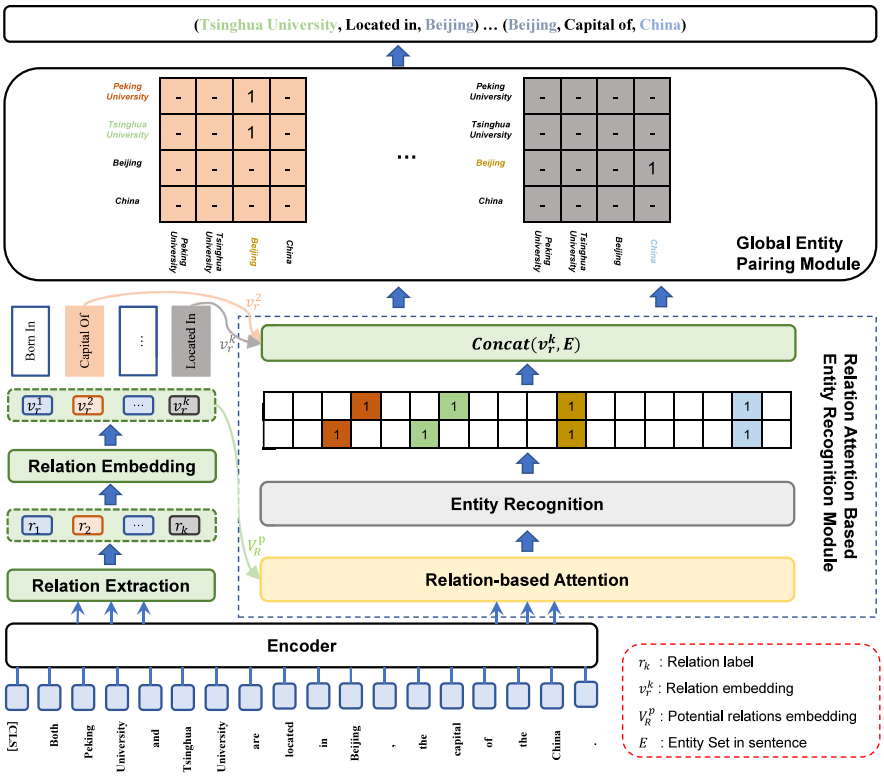


Fig. 2. The overall framework of ERGM.

vector. σ denotes sigmoid function. We denote the set of relations detected based on formula (3) as R_p.

The Relation Extraction Module minimizes the following binary cross-entropy loss function to detect relations, as shown in formula (4):

$$\mathcal{L}_r = -\frac{1}{n}\sum_{i=1}^{R}\left(y_i \log P_r + (1-y_i)\log\left(1-P_r\right)\right) \tag{4}$$

where $y_i \in \{0, 1\}$ indicates whether the current label is a relation existing in the sentence. R is the number of elements in relation set.

3.4. Relation attention based entity recognition module

As shown in Fig. 2, we can get the set of all potential relations in the sentence. Next, we need to extract all subject and object entities. To make better use of relation information in entity recognition, we propose an attention mechanism based on potential relations. Since sentences often contain only part of the relations in the relation set, using all relations with sentences for attention calculation would incorporate a large amount of irrelevant relational information, which instead has an interfering effect on the subsequent entity recognition. Therefore, we use the set of potential relations for relation attention calculation. Assuming that the maximum number of relations present in the sentence is r_{max}, and r_{max} is much smaller than R, then we use the relation encoder to encode it to get the vector V_p^r of potential relation set and use V_p^r with the sentence vector H to do the attention calculation, thus enriching the information of each token in the given sentence and improving the effect of entity recognition. The module calculates the relation-aware attention weights, which indicates the correlation between each token embedded in a given sentence and the relation embedding, and then obtains H^R. As shown in formula (5)–(8).

$$Q = W_{\text{query}}\, H + b_{\text{query}} \tag{5}$$

$$K = W_{key} V_p^r + b_{key} \tag{6}$$

$$R_H = \text{Softmax}\left(\frac{QK^\top}{\sqrt{d_k}}\right) V_p^r \tag{7}$$

$$H^R = W_{H\_R}\ \text{Concat}\ (H, R_H) + b_{H\_R} \tag{8}$$

where W_{query} , $W_{\text{key}} \in \mathbb{R}^{d_e \times d_k}$, $W_{H\_R} \in \mathbb{R}^{d_e \times 2d_k}$ are trainable parameters, $V_p^r \in \mathbb{R}^{r_{\max} \times d_e}$ is the trainable embedding of the relations in the sentence, $H^R \in \mathbb{R}^{n \times d_e}$ is the sentence representations that incorporates relevant relations information.

After obtaining the sentence representation that fuses relational information, we apply two binary classifiers to predict the start and end positions of entities respectively. Our model does not need to distinguish specific subject entities and object entities, which can improve the accuracy of entity recognition on the one hand, and simplify the extraction structure of the model on the other hand. We use a simple 1/0 tagging scheme, which means that if token is the start or end of an entity it will be tagged as 1, otherwise it will be tagged as 0. As shown in formula (9) and (10).

$$P_i^{\text{start}} = \sigma\left(W_{\text{start}}\, h_i^r + b_{\text{start}}\right) \tag{9}$$

$$P_i^{end} = \sigma\left(W_{end} h_i^r + b_{\text{end}}\right) \tag{10}$$

where W_{start} , $W_{\text{end}} \in \mathbb{R}^{d_e}$ are trainable weights, P_i^{start} and P_i^{end} represent the probability that the *i-th* token in the sentence is the start or end of the entity, respectively.

In this stage, ERGM minimizes the following loss function to discern the tag of entities. As shown in formula (11).

$$\mathcal{L}_e = -\frac{1}{n}\left(\log P_\theta^{start}\left(s \mid H^R\right) + \log P_\theta^{end}\left(s \mid H^R\right)\right) \tag{11}$$

where $P_\theta\left(s \mid H^R\right) = \prod_{i=1}^{n} p_i^{\{y_i=1\}} (1-p_i)^{\{y_i=0\}}$, i is represent the i-th token in the sentence, P_θ^{start} and P_θ^{end} are the likelihood for the start and end positions, respectively.

3.5. Global entity pairing module

Based on the work in Section 3.3 and Section 3.4, we can obtain the sets of potential relations and entities. The entities will form different entity pairs under different relations, thus forming a sequence of triples. We first obtain the vector representation sequence H_e of entities based on the sentence vector representation H and the entity set E. Under each potential relation, we construct an entity matching matrix based on the vector representation sequence of entities to determine whether the current entity pair can be matched as a set of triples under that relation. We propose a method of entity pair matching matrix based on affine transformation method, which is constructed based on entities. It is assumed that the maximum number of entities present in the sentence is e_{max}, and e_{max} is much smaller than the number of tokens in the sentence. Therefore, compared with the token-based matching method, not only the direct connection between entities and relations can be more fully utilized, but also the sparsity of the matrix is greatly reduced and the complexity of training space and time is reduced.

The specific matching process is as follows: first, all entities in set are identified according to the Entity Recognition Module, and then a sequence of entity set vectors is obtained from the sentence vector representation. The entity vector sequence is combined with a specific relation vector, and the entity matching matrix under a specific relation is constructed using the affine transformation method. We check the score corresponding to each entity pair in the entity matrix and keep it if the value exceeds a certain threshold θ_e, otherwise we filter it out. In the experimental section, we explore the effect of different values θ_e and θ_r on the experimental results. For the triples "Tsinghua University, Located in, Beijing" and "Beijing University, Located in, Beijing", we can first obtain the entity set "Tsinghua University, Beijing University, Beijing", and then based on the sentence vector H and entity tag to obtain the entity sets vector H_e, and finally combine the relation "Located in" and entity sets vector to build the corresponding entity matching matrix, thus forming two triples. Entity sets refers to all entity mentions contained in a sentence, which can be divided into subject entity and object entity. As shown in formula (12)–(14).

$$h^e = \frac{h^{\text{start}} + h^{end}}{2} \tag{12}$$

$$h_e^r = \text{concat}\left(h_e, v_k\right) \tag{13}$$

$$P^k = \sigma\left(W_{e\_r}\left[h_e^{r\ \mathrm{T}} W_e h_e^r\right] + b_{e\_r}\right) \tag{14}$$

Where $W_{e\_r}, W_e \in \mathbb{R}^{d_e \times d_e}$ are trainable weights, $v_k \in \mathbb{R}^{1 \times d_e}$ is the trainable embedding of the relations in the sentence, h^e is the entity vector, as an entity often contains multiple tokens, we average the token vectors of the start and end of the entity to obtain a vector representation of the entity. The entity matching matrix G under specific relation consists of matching probabilities P^k between different entities. We employ a biaffine model over the sentence to create a $l \times l$ scoring tensor. l is the max number of the entity sets in the sentence. We use the sigmoid function to determine whether the matching entity pair at each position meets the requirements.

As similar as Relation Extraction Module, Global Entity Pairing Module minimizes the following binary cross-entropy loss function. As shown in formula (15).

$$\mathcal{L}_g = -\frac{1}{n}\sum_{k=1}^{R_p}\sum_{i=1}^{e_{\max}}\sum_{j=1}^{e_{\max}} y_{i,j}^k \log\left(P_{i,j}^k\right) + \left(1-y_{i,j}^k\right)\log\left(1-P_{i,j}^k\right) \tag{15}$$

where e_{max} is stand for the max number of entity sets and R_p is the number of potential relation. We jointly train Relation Extraction Module, Relation Attention Based Entity Recognition Module and Global Entity Pairing Module. The total loss of our model is as shown in formula (16).

$$\mathcal{L} = \alpha\mathcal{L}_r + \beta\mathcal{L}_e + \gamma\mathcal{L}_g \tag{16}$$

where α, β and γ are custom constants. In our experiments, we set $\alpha = \beta = \gamma = 1$.

3.6. Model training process

To explain our model more clearly, we describe our algorithm flow in pseudocode in Algorithm 1. ERGM first identifies the relations in the sentences according to Eq. (3), and then constructs the set of potential relations R_p. Next, a relation-based attention mechanism is used to fuse the sentence and relations information to obtain a new sentence representation that identifies the entities in the sentence. In the training phase, we use a negative sample sampling strategy to introduce different proportions of false entities to alleviate the exposure bias of the model. Then the vector representation of the entity set is obtained based on the sentence vector representation and the entity matching matrix under each relation is constructed using Eq. (14). Finally combine entities and relations into triples.

Algorithm 1 The Training Process of ERGM

Input: Sentence Set $S = \{s_1, s_2, \ldots, s_n\}$, Relation Set $R = \{r_1, r_2, \ldots, r_k\}$
Output: A Set of Triples $T = \{(h, r, t)\}$ // h and t stands for subject and object entity

```
Initialize: RelationList, EntityList, Sentence Embedding H
for n = 1 → N do
    H = BERT(S)
    Obtain Potential RelationList R_P with Eq. (3)
    Compute Relation-based Attention with Eq. (5)-Eq. (8)
    Obtain EntitySet E with Eqs. (9) and (10) (negative sampling)
    Get EntitySet Embedding with Eq. (12)–Eq. (13)
    Build entity matching matrix G with Eq. (14)
    for i = 1 → k do
        if G_nm == 1 then // n, m is less than the length of the entity set
            T.append((E_n, R_i, E_m))
        end if
    end for
    Get the loss of the model with (16)
    Model Backpropagation Update Parameters
end for
```

4. Experiments

In the section, we present the details of the experiments and analyze the results of the experiments.

Table 2
The statistics of datasets. N is the number of triples in a sentence. Note that one sentence can have Normal, SEO and EPO overlapping patterns simultaneously.

Category	Dataset				Details of Test Set							
	Train	Valid	Test	Relations	Normal	SEO	EPO	$N = 1$	$N = 2$	$N = 3$	$N = 4$	$N >= 5$
NYT	56196	4999	5000	24	3266	1297	978	3244	1045	312	291	108
WebNLG	5019	500	703	171	245	457	26	266	171	131	90	45

Table 3
Hyper parameters on NYT.

Hyper-parameters	Values
Dimension of word embedding	768
Dimension of relation embedding	768
Dimension of relational attention	64
Dropout	0.3
Learning rate	0.001
Train/Test batch size	64
Valid batch size	24
Max sequence length	100
Max entity sets	5
Max relation sets	2
Epoch	100

4.1. Experiment settings

4.1.1. Datasets

We conduct experiments on two widely used public datasets. NYT [33] is a large-scale dataset constructed based on the "New York Times" news corpus using a remote supervision method. The dataset of WebNLG [34] is derived from articles in Wikipedia, it is constructed according to the manual annotation by the annotator. We follow the paper [14], divided data into train set, valid set and test set. Furthermore, to better evaluate the ability of the model to handle overlapping triples, we divide the sentences of NYT and WebNLG into three categories according to three overlapping patterns: Normal, EPO, and SEO. The statistics of datasets are shown in Table 2.

4.1.2. Evaluation metrics

For a fair comparison with previous work, we adopt the Precision (Prec.), Recall (Rec.) and F1 scores to evaluate our model. For NYT and WebNLG data, we use the Partial Match method. As long as the relation, head of the subject and object entities are matched, then we consider this triple is right.

4.1.3. Implementation details

In the experiments, we use a single RTX 3090 GPU to training our model in Ubuntu 20.04 OS. Our BERT uses the BERT-Base-Cased version, which contains 12 Transformer blocks and the hidden size d is 768, the number of self-attention heads is 12. We tune our model on the valid set to adjust important hyper parameters. For our own modules, the dimension of relation embedding and relation-attention is set to 768, the maximum length of sentence and epoch is set to 100, which is consistent with BERT. Considering the size of the datasets, we set batch sizes of 64 and 6 for NYT and WebNLG, respectively. We give the training parameters on the NYT dataset in Table 3.

4.1.4. Baseline models

Our model is compared with the following baseline models: (1) NovelTagging [11] uses a joint decoding scheme to extract triples, which unifies the entity and relation extraction as a sequence tagging task, so it fails to solve the overlapping problem. (2) CopyRE [22] first explores Seq2Seq model for the joint entity and relation extraction task, and generates the triplets in the sentence sequentially using copy mechanism. This model can only copy the last word of an entity. (3) ETL-Span [19] decomposes the joint extraction task into two interrelated subtasks, which first distinguish all head entities, then identify the corresponding tail entities and relations. (4) CasRel [14] proposes a novel tagging framework, which first extracts the subjects and then finds the corresponding objects according to each relation type. (5) PMEI [35] proposes a progressive multi-task learning model that exploits early predicted interactions to improve task-specific representations. (6) TPlinker [20] model transforms the joint entity and relation extraction task into a token pair linking problem and introduces a handshaking tagging scheme. (7) StereoRel [36] models triples utilizing three-dimensional space, which can reduce information loss. (8) RIFRE [37] proposes a representation iterative fusion based on heterogeneous graph neural networks for relation extraction. It models relations and words as nodes on the graph and fuses the two types of semantic nodes by the message passing mechanism iteratively to obtain nodes representation that is more suitable for relation extraction tasks. (9) PARE [24] proposes a joint extraction model with position-aware attention and relation embedding, which introduces an additional encoder to encode relational descriptions to incorporate relational features. (10) PRGC* [21] proposes a joint relational triple extraction framework based on Potential Relation and Global Correspondence; (11) EmRel [30] explicitly introduces relation representation for triple extraction.

4.2. Experimental results and analysis

4.2.1. Main result

Table 4 shows the comparison of experimental results between our model and the baseline models on datasets NYT and WebNLG. Bold marks represent the best results, while underlining represents the second best results. It can be seen from the table that ERGM achieved the best F1 value on NYT datasets, On the WebNLG dataset, the results are weaker than EmRel. For the New York Times dataset, our model outperforms the best method StereoRel by 0.21% in F1-score and continues to improve in accuracy. For the WebNLG dataset, our model achieves 0.21% less F1-score than the previous best model EmRel. For PRGC*, we use its public source code to reproduce the results obtained. Our model outperforms PRGC* by a large margin on the NYT dataset, while on the WebNLG dataset, the experimental results of both are comparable. We analyze the reasons may be as follows. First, ERGM uses a latent relational attention mechanism, which can model sentence and relation information in a better way. Second, the number of relations in WebNLG is much more than NYT, so it is more difficult to learn, and the effect and representation of relation recognition will directly affect the subsequent results of the model. This is the error propagation problem that our multi-stage model faces, and although we use negative samples to alleviate it, it still exists. In Section 4.2.3, we conduct related experiments to analyze different negative sampling strategies in detail. In the part of ablation experiments, we demonstrate that our negative sampling strategy is effective. We will further investigate the exposure bias issue in future work.

Table 4
Results of baseline models on NYT and WebNLG datasets. Bold marks the best result, underline marks the second best result.

Model	NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1
NovelTagging [11]	32.8	30.6	31.7	52.5	19.3	28.3
CopyRE [22]	61.0	56.6	58.7	37.7	36.4	37.1
ETL-Span [19]	84.9	72.3	78.1	85.5	71.7	78.0
CasRel [14]	89.7	89.5	89.6	93.4	90.1	91.8
PMEI [35]	90.5	89.8	90.1	91.0	92.9	92.0
TPlinker [20]	91.3	**92.5**	91.9	91.8	92.0	91.9
StereoRel [36]	92.0	92.3	92.2	91.6	92.6	92.1
RIFRE [23]	**93.6**	90.5	92.0	93.3	92.0	92.6
PARE [24]	92.9	91.4	92.1	93.8	91.0	92.4
PRGC*[21]	93.4	89.7	90.8	92.9	92.4	92.6
EmRel [30]	91.7	**92.5**	92.1	92.7	**93.0**	**92.9**
ERGM	93.3	91.5	**92.4**	**94.2**	91.2	92.7

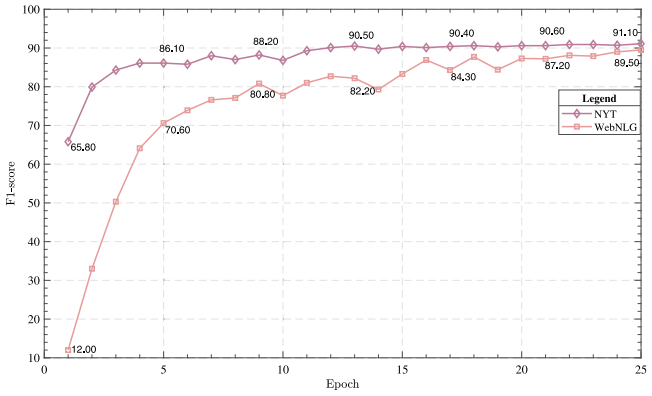


Fig. 3. F1-score in model training.

4.2.2. Efficiency of the model

Fig. 3 shows the variation of the F1 scores of our model over 25 epochs. In the first 5 epochs, the F1 score of the model increases rapidly. After 15 epochs, the model gradually converges. Until 25 epochs, the learning of the model has converged to fit.

To compare the performance of the models, we test the efficiency of the models on the NYT and WebNLG datasets using a unified configuration. (1) 3090 GPU (2) The training batch size is set to 6, and the test batch size is set to 1 (3) The max length of the sentence is set to 100. Statistical model training time(s) and inference time(ms). Among them, the training time(s) refers to the training time of one batch, the unit is seconds; and the inference time(ms) refers to the time of each instance, the unit is milliseconds. The memory occupation is the amount of GPU memory occupied by the model when it runs, and it is measured in G. We test the model on the NYT and WebNLG dataset with batch sizes of 6 and 64.

We select the following models as our baseline models for comparison, CasRel, TPLinker, RIFRE, PRGC and EmRel. The specific experimental results are shown in Table 5. Complexity is the theoretical decoding computational complexity with respect to sequence length n and relation set size k. CasRel jointly decodes relations and entities in sentences, and it has a computational complexity of $O(n + sro)$, where n is the length of the input sentences, $s/r/o$ represent the number of relations, subject and object entities identified in the sentence, respectively. TPLinker iterates over all token pairs and uses three matrices to tag token links to recognize relations, therefore it has a high computational complexity of $O(kn^2)$. RIFRE use a representation iterative fusion based on heterogeneous graph neural networks for relation extraction, and PRGC extracts the triples in sentences by constructing a token matching matrix. Their computational complexity is $O(n^2)$. EmRel uses TPLinker as the optimized baseline model with the addition of information embedding of relation labels and entity relation information fusion modules, so its time complexity is similar to TPLinker, while its model training time and memory usage far exceed TPLinker.

ERGM uses entity matching matrix to identify sentences and only needs one sequence annotation to get the entities in the sentence, so its computational complexity is $O(n + e^2)$, where n denotes the length of the input sentence and e denotes the number of entities in the sentence. e is much smaller than n, so our computational complexity has obvious advantages compared with other models. Furthermore, the training time of the model on WebNLG is much less than that on the NYT dataset, which is mainly due to the different sizes of the two datasets. The training time of our model on both datasets is significantly better than the other models for several main reasons. First, we simplified the model structure by using relation judgments to filter out irrelevant relations. This is similar to the method used by PRGC, so ERGM and PRGC are much more efficient in inference time on the WebNLG dataset. Because the number of relations in the WebNLG dataset is 171, which is much more than 24 in the NYT dataset, TPLinker and CasRel need to deal all relations, which greatly increases the training and inference time cost. In addition, compared with the PRGC model, we do not need to identify the specific types of entities and use one sequence token to obtain all entities. The length of the token matching matrix constructed by PRGC is much higher than our entity matching matrix, because the number of tokens in the sentence is more than the number of entities.

From Table 5, we can see that on the NYT dataset, our Training Time (TT) of each epoch is 1061 s, while the PRGC is 1607 s, and we are 1.51 times faster than it. About inference time for epoch size 1/24 on NYT, PRGC is 13.5/4.4, while ours is 9.4/1.4, we are 1.44/2.14 times faster than it. On the WebNLG dataset, our Training Time(TT) of each epoch is 112 s, while the PRGC is 213 s, and we are 1.9 times faster than it. About inference time for epoch size 1/24 on WebNLG, PRGC is 14.4/5.2, while ours is 8.5/1.4, we are 1.69/2.71 times faster than it. In terms of memory occupation, ERGM also achieves optimal results based on its model architecture. All these results fully illustrate that ERGM integrates the efficiency and performance of the model, which will be more competitive in practical application scenarios.

4.2.3. Negative sampling

Since our model is a multi-task structure, we need to extract entities first, obtain the vector representation sequence of entities, and then judge them based on the entity matching matrix. During the training phase, the entities we use are gold standard entities. However, in the testing and verification phase, the entities we use need to be extracted from the sentences, which will generate some wrong entities and lead to the exposure bias of the model. Therefore, we introduce a negative sample strategy to alleviate this problem. we determine the number of negative samples to be inserted according to the F1 value of the entity identified by the model. Inserting too many negative sample entities will have a negative impact on the model. We also select the best parameters through experiments. we train our model on the NYT dataset with different negative scales and verify the performance of the model. The specific experimental results are shown in Fig. 4.

We use four different negative sample proportions, for each sentence we randomly add an incorrect entity with probabilities of 25%, 50%, 75%, and 100%, respectively. In order to ensure the consistency of the entities in the training and testing phases, the negative sample proportions are therefore set to correlate with

Table 5
Comparison of the model efficiency. Bold marks the best result. MO: Memory Occupation (G), IT: Inference Time (ms), TT: Training Time (s). Complexity are the Computational complexity, we use Big O Notation. IT(1/24) denotes the inference time (ms) per instance with the batch size of 1 and 24. TT denotes the training time (s) with the batch size of 6. MO(6/64) (G) denotes the model memory occupation with the batch size of 6 and 64.

Model	Complexity	NYT			WebNLG		
		MO(6/64) (G)	TT (s)	IT(1/24) (ms)	MO(6/64) (G)	TT (s)	IT(1/24) (ms)
CasRel [14]	$O(n+sro)$	5.9/24.4	2481	24.2/-	6.0/24.8	298	30.5/-
TPLinker [20]	$O(kn^2)$	6.5/23.6	5228	38.8/7.7	5.0/21.4	599	41.7/13.2
RIFRE [23]	$O(n^2)$	–	1555	–	–	266	–
PRGC [21]	$O(n^2)$	5.1/22.2	1607	13.5/4.4	5.0/22.2	213	14.4/5.2
EmRel [30]	$O(kn^2)$	–	–	–	–	–	–
ERGM	$O(n+e^2)$	**4.0**/**8.6**	**1061**	**9.4**/**1.4**	**4.0**/**8.7**	**112**	**8.5**/**1.4**

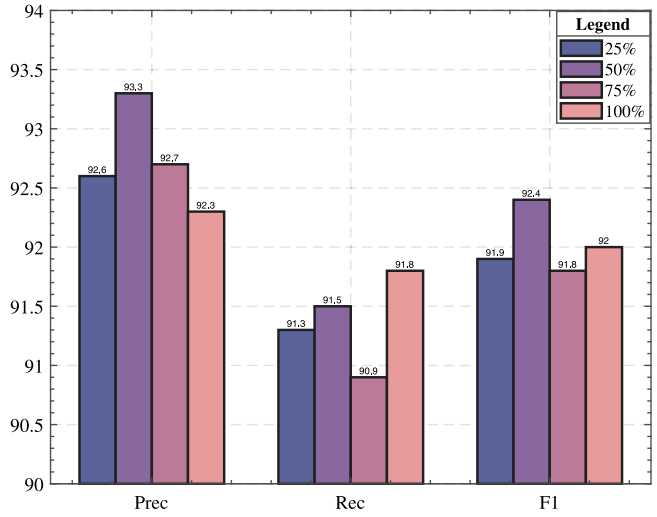


Fig. 4. Experimental results under different entity negative sample strategies. We select the negative samples of entities in the sentence according to different probabilities, which are 25%, 50%, 75%, and 100% respectively.

the error rate in our entity recognition task. At a rate of 50%, the model obtains the best Precision and F1 values, as well as the second best Recall. At ratios of 25% and 75%, the model performs poorly in general. And at the ratio of 100%, the model obtains the best Recall.

4.2.4. Detailed results on complex scenarios

To verify that our model handles the cases where sentences contain overlapping and multiple triples, following the work of CasRel, we conduct related extended experiments using the NYT and WebNLG datasets. We selected 5 previous models as baselines, and the specific experimental comparison results are shown in the following Tables 6 and 7. Bold text indicates best results and underlined text indicates second best results.

ERGM achieve the best F1 score on 8 of the 16 subsets and the second best F1 score on 6 of the 16 subsets. Our model has a clear advantage in handling simple overlapping cases, while the performance of the model slightly reduced in more complex cases. Overall, the performance of ERGM still holds a certain advantage among all baseline models. This is determined based on the structure of our framework. When the number of entities and relationships becomes larger, the error propagation of the model will be more serious. In the entity pairing link, it is more difficult for the model to match the correct triples.

4.2.5. Detailed results on different subtasks

We analyze the performance of the model under different subtasks, where e represents the subject and object entities, s represents the subject entities, o represents the object entities,

Table 6
Results of models in Normal, EPO and SEO patterns. Bold marks the best result, underline marks the second best result.

Model	NYT			WebNLG		
	Normal	EPO	SEO	Normal	EPO	SEO
CasRel	87.3	92.0	91.4	89.4	94.7	92.2
TPLinker	90.1	<u>94.0</u>	<u>93.4</u>	87.9	<u>95.3</u>	92.5
RIFRE	<u>90.5</u>	93.7	93.6	<u>90.1</u>	94.7	93.1
PRGC*	88.4	92.7	93.4	88.4	93.4	**95.4**
PARE	<u>90.5</u>	93.7	93.6	89.6	94.1	92.9
ERGM	**90.9**	**94.1**	**93.6**	**90.3**	**96.0**	<u>93.8</u>

and r represents the relations between them. According to Table 8, it can be seen that the model has a higher than 96.0% accuracy in the entity recognition and relation extraction tasks, but in combination, the performance drops significantly. The joint extraction is to alleviate the error between the two tasks. In addition, for the datasets NYT and WebNLG, there is a clear gap between the two subtasks i.e., (s, o) entity pairs recognition and r extraction, because the difficulty of entity recognition subtask is much higher than that of relation extraction, which is also one of the main challenges for triple extraction.

Compared with NYT, for WebNLG, the performance gap between (s, o) and r is much smaller. An important reason is that there are a large number of EPO triples in the NYT dataset, and the proportion is much higher than that of WebNLG (26.4% vs 14.2%), which increases the difficulty of this subtask. For the (s, o) entity pair recognition and (s, r, o) triple extraction tasks, NYT performs better because the number of relations in NYT is much less than WebNLG (24 in NYT and 171 in WebNLG). In addition, the best results for the combined task are achieved by (s, r) on the NYT and (s, o) on the WebNLG. This is mainly due to the different number of relations in the two datasets and thus affecting the performance of the combined task on different datasets.

4.2.6. Ablation study

To verify the effectiveness of our proposed module, we conduct ablation experiments on the NYT and WebNLG datasets. The results in Table 9 show that both relational attention and global entity pairing have an important impact on the model, as the attention mechanism can better integrate relational information into sentences and improve the accuracy of entity recognition. In our model structure, the final effect of the model will be affected by the sub-tasks of entity recognition and relation recognition. According to Table 8, it can be seen that the effectiveness of the model gradually decreases as the number of subtask combinations increases. On the NYT dataset, the (s, r, o) task decreases by 3.5% compared to the e task and by 4% compared to the r task. Therefore, the improvement of entity recognition effect will improve the final recognition effect of the model. To verify the impact of global entity pairing on the model, we change the entity recognition strategy to identify entities under a specific relation.

Table 7
F1-score of baseline models on NYT and WebNLG datasets with multiple patterns. Bold marks the best result, underline marks the second best result.

Model	NYT					WebNLG				
	N = 1	N = 2	N = 3	N = 4	N >= 5	N = 1	N = 2	N = 3	N = 4	N >= 5
CasRel	88.2	90.3	91.9	94.2	83.7	89.3	90.8	94.2	92.4	90.9
TPLinker	90.0	92.8	93.1	**96.1**	90.0	88.0	90.1	94.6	93.3	91.6
RIFRE	90.7	92.8	**93.4**	94.8	89.6	90.2	**92.0**	94.8	93.0	92.0
PRGC*	89.0	91.9	92.5	95.6	86.2	88.4	91.9	94.0	**94.8**	**92.9**
PARE	90.5	92.7	93.3	95.2	**91.7**	89.4	90.9	**95.2**	93.3	92.0
ERGM	**90.9**	**93.4**	93.1	95.7	90.1	**90.3**	91.5	95.1	94.0	92.5

Table 8
Results on relational triple elements. Bold marks the best result.

Model	NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1
e	96.6	94.9	95.7	**98.8**	**94.3**	**96.5**
s	96.2	94.4	95.2	98.6	93.7	96.1
o	96.0	94.6	95.3	98.4	92.0	95.1
r	**96.7**	**95.6**	**96.1**	95.5	91.2	93.3
(s, r)	95.2	92.8	94.0	95.6	90.3	92.9
(r, o)	94.9	92.7	93.8	96.1	90.5	93.2
(s, o)	93.5	92.2	92.9	96.8	90.6	93.6
(s, r, o)	93.3	91.5	92.4	94.2	91.2	92.7

Table 9
Ablation study of ERGM on the NYT and WebNLG dataset. Bold marks the best result.

Model		Prec.	Rec.	F1
NYT	w/o Relational Attention	93.0	90.9	91.9
	w/o Global Entity Pairing	91.5	90.0	90.7
	w/o Negative Sample	91.6	91.3	91.4
	Origin	**93.3**	**91.5**	**92.4**
WebNLG	w/o Relational Attention	93.9	90.8	92.3
	w/o Global Entity Pairing	91.4	**92.2**	91.8
	w/o Negative Sample	92.5	90.9	91.6
	Origin	**94.2**	91.2	**92.7**

"w/o Global entity pairing" is to first extract the relations in the sentence, and then use the method of sequence labeling to extract the head and tail entity under the specific relation. We use a nearest neighbor matching strategy to match entities, "Nearest neighbor matching strategy" refers to matching the two closest entities in a sentence into a triplet, and the same method is used in the work [11]. The ERGM sets a higher threshold and results in lower recall. We combined the final F1 values to determine our thresholds, and the specific experiments are shown in Fig. 6 of Section 4.4. In addition, increasing negative samples in a moderate proportion can effectively alleviate the exposure bias problem of the model, which leads to better generalization of the model.

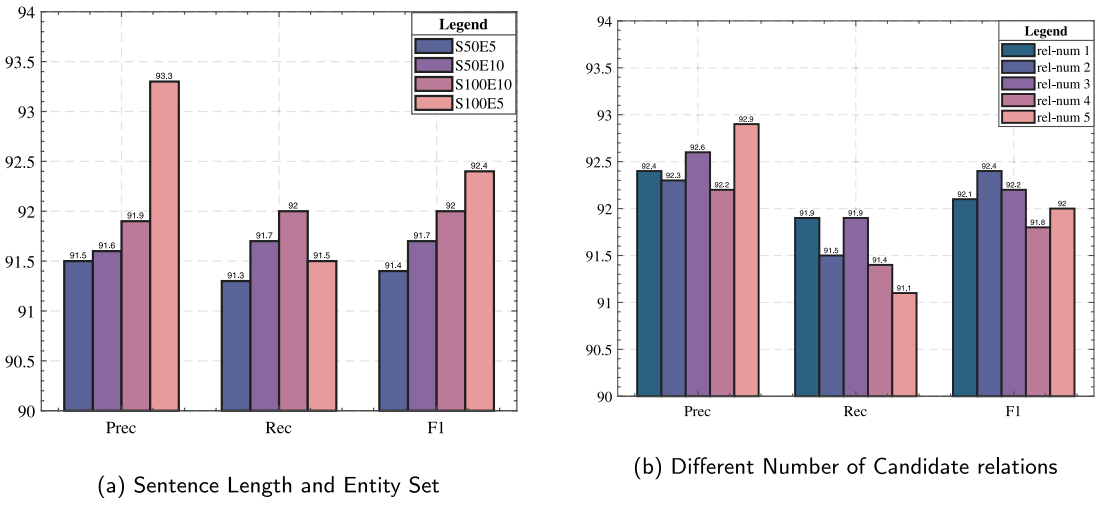
4.3. Parameter selection

We conduct experiments on several important hyperparameters of the model to select the most suitable parameters for the model. The combination of sentence length and entity set candidate number includes the following four: S100E10, S50E5, S100E5, S50E10, S represents the sentence length, and E represents the number of entity set candidates. In addition, we also perform experimental comparisons on different numbers of candidate relations. The specific experimental results are shown in Fig. 5. We need to identify the entities from the sentences to construct the entity matching matrix. The Max entity sets need to meet the number of entities in most of the sentences, but too large an entity set size will make the matrix too sparse, so we experimentally determine the optimal entity set size. The same idea of setting the Max relation set size as Max the entity set size, if the relation set size is too large, more irrelevant relations will be generated, so we also experimented to select the final relation set size.

It can be seen from Fig. 5(a) that our model achieves the best F1 value when the combination of sentence length and the number of entity sets is S100E5. This is because the number of entities presents in a sentence is limited, and if we set too many entities, it will generate too many negative samples for the subsequent entity matching task. Conversely, if the sentence length is too short, it is difficult to identify the entities in a complete long sentence. According to Fig. 5(b), the number of relations in our analyzed sentences is limited, and if we use too large a set of candidate relations, too many negative sample relations will be introduced in the process of relation attention calculation as well. Choosing the right size of the relations set is very important for the results of the model. The number of relations in most of the sentences is between 1 and 3, so the model achieves best results when we set the set of candidate relations to 2.

To validate the experimental results of each subtask of our model under different thresholds, we conduct experiments on the thresholds of relation, entity and entity pair on the NYT dataset. We fix the other two thresholds respectively, adjust the current threshold, and observe the changes in the experimental results. The default value of relations threshold is 0.9, the default value of entity threshold is 0.9, and the default value of entity pair threshold is 0.5. The specific experimental results are shown in Fig. 6.

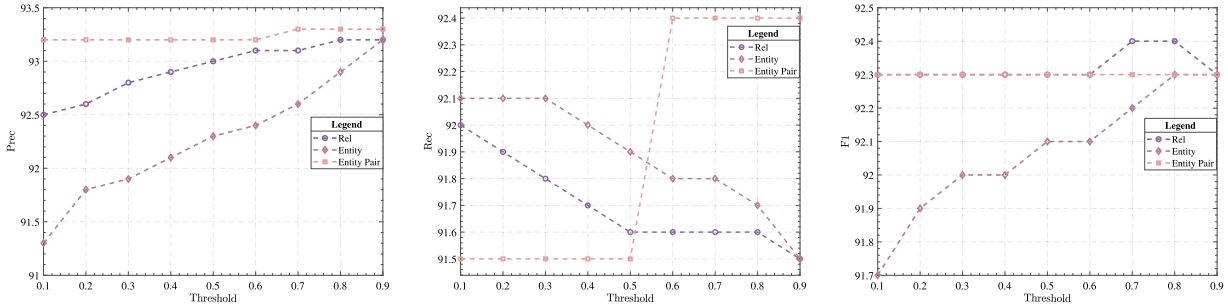
In Fig. 6(a), the precision of the model is proportional to the change in the size of the threshold value. The larger the threshold value, the greater the number of incorrect triples the model filters out and the higher the confidence level of the remaining triples, which will improve the accuracy of the model recognition. In Fig. 6(b), the size of the threshold for relations and entities is inversely proportional to the recall of the model. The smaller the threshold value, the lower the confidence level of the triples identified by our model, thus allowing more correct triples to be identified, but also increasing the proportion of incorrectly identified triples. However, the size of the entity pairs threshold is positively proportional to the recall, and we speculate that the reason may be that the entity pair matching task is more difficult compared to entity recognition and relation recognition, and many negative samples entity pairs also exist with higher confidence levels. According to Fig. 6(c), we can see that the F1 value of the model is proportional to the threshold size of the relations and the entities. When the threshold value of entities and relations is fixed to 0.9, the F1 value is not affected by the effect of entities on the threshold value. We speculate that the main reason is that most of the relations and entities identified in the predecessor task are correct, so the model will have a higher confidence level because most of the entity pair matches in the final entity pair are positive cases.



(a) Sentence Length and Entity Set

(b) Different Number of Candidate relations

Fig. 5. Experimental results under different parameter selection on the NYT dataset.



(a) Precision

(b) Recall

(c) F1 Score

Fig. 6. Experimental results under different thresholds on the NYT dataset.

4.4. Case study

We select two specific examples in this section to analyze the modules of our model. As shown in Fig. 7, the green marker indicates the triples present in the sentence, red indicates the incorrectly identified triples, and blue indicates the correctly identified triples. The 18 and 10 refer to the relations "location/contains" and "place lived", respectively. We compare the specific recognition results of the model under ERGM and three different condition settings. According to the first sentence, we can observe that our model identifies more incorrect entities if the relational attention mechanism and the negative sample module are missing. And if there is no global entity pairing, according to the nearest neighbor matching principle, the model generates more incorrect triples under the condition of incorrect entity recognition. Distinguishing specific head and tail entities is difficult for the model because there are cases of overlapping head and tail entities in the triples. For example, an entity is both a head and a tail entity. The relation "18 in the sentence does not exist, and ERGM filters the entities that are not related to the relation "18" in the matching process, so the number of incorrect triples is effectively reduced. The second sentence is a complex scenario of triple extraction, because the sentence to be recognized is long and includes several overlapping triples. It is very difficult for all four models to recognize all correct triples completely. However, compared with the models under other settings, ERGM still maintains certain advantages. Specifically, the introduction of relational attention can effectively enhance the effect of entity recognition, which directly affects the extraction of subsequent triples, because wrong entities will generate wrong entity pair matching matrices, which will have an impact on the subsequent results. In addition, the entity pair matching matrix can effectively filter irrelevant relations, thus reducing the number of incorrect triples extracted.

5. Conclusion

This paper proposes an entity and relation extraction method based on global entity pairing and relational attention mechanism, which can not only effectively solve the overlapping triples problem, but also improve the time performance of the model. The model uses a BERT pre-trained encoder and consists of three modules, which are Relation Extraction Module, Relation Attention Based Entity Recognition Module and Global Entity Pairing Module. The Relation Extraction Module identifies possible relations in sentences and reduces irrelevant relation recognition. The Relation Attention Based Entity Recognition Module enhances the information interaction between relations and entities. We extract all the entities at once, with only one tagging sequence, and the task is significantly easier, which improves the accuracy of entity recognition. The Global Entity Pairing Module can greatly save memory space, improve the modeling performance, and solve the entity nesting problem in triples. However, although we use the negative sampling strategy, our model still suffers from the exposure bias problem, which leads to error propagation. In future work, we hope to further explore the problem of joint entity and relation extraction in model exposure bias and few-shot scenarios.

Instance	
Sentence #1	Senator Ernie Chambers , Nebraska 's only black legislator , who argued that Omaha schools were already segregated.
Ground Truth	[(Nebraska, Omaha, 10)]
w/o Attention	[(Nebraska, Omaha, 10), (Chambers, Omaha, 18)]
w/o Negative Sample	[(Chambers, Omaha, 18), (Nebraska, Omaha, 10)]
w/o Global Entity Pairing	[(Nebraska, Omaha, 10), (Chambers, Omaha, 10)]
ERGM	[(Nebraska, Omaha, 10)]
Sentence #2	But except for a short stretch north of Luang Prabang , traveling the river itself , which runs from China through Myanmar -LRB- formerly Burma -RRB- Thailand , Laos , Cambodia and Vietnam , was almost unheard of , and much of the 2,610-mile-long Mekong , home to some 1,200 species of fish , remained unknown .
Ground Truth	[(Laos, Mekong, 10), (Laos, Prabang, 10), (Vietnam, Mekong, 10), (Cambodia, Mekong, 10), (Thailand, Mekong, 10)]
w/o Attention	[(Cambodia, Prabang, 10)]
w/o Negative Sample	[(Cambodia, Mekong, 10)]
w/o Global Entity Pairing	[(Prabang, Cambodia, 10), [(Cambodia, Vietnam, 10)]
ERGM	[(Laos, Prabang, 10), (Cambodia, Mekong, 10)]

Fig. 7. Instances in NYT dataset. The green marker indicates the triples present in the sentence, red indicates the incorrectly identified triples, and blue indicates the correctly identified triples.

CRediT authorship contribution statement

Chen Gao: Methodology, Software, Validation, Writing – original draft. **Xuan Zhang:** Conceptualization, Funding acquisition, Supervision, Resources, Writing – review & editing. **LinYu Li:** Visualization. **JinHong Li:** Formal analysis. **Rui Zhu:** Data Curation. **KunPeng Du:** Formal analysis. **QiuYing Ma:** Writing – review & editing.

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.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant No. 61862063, 62002310, 61502413,61262025; the Science Foundation of Young and Middle-aged Academic and Technical Leaders of Yunnan, China under Grant No. 202205AC160040; the Science Foundation of Yunnan Jinzhi Expert Workstation under Grant No. 202205AF150006; the Science and Technology Project of Yunnan Power Grid Co., Ltd. under Grant No. YNKJXM20222254; the Yunnan Provincial Natural Science Foundation Fundamental Research Project under Grant No. 202101AT070004; the Yunnan Provincial Department of Education Science Research Fund Project under Grant No. 2023Y0253; the Open Foundation of Yunnan Key Laboratory of Software Engineering under Grant No. 2020SE301; the Science Foundation of "Knowledge-driven intelligent software engineering innovation team".

References

[1] S. Ji, S. Pan, E. Cambria, P. Marttinen, S.Y. Philip, A survey on knowledge graphs: Representation, acquisition, and applications, IEEE Trans. Neural Netw. Learn. Syst. 33 (2) (2021) 494–514.

[2] Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al., Google's neural machine translation system: Bridging the gap between human and machine translation, 2016, arXiv preprint arXiv:1609.08144.

[3] V.S. S. K. Dwivedi, Research and reviews in question answering system, Proc. Technol. 10 (2013) 417–424.

[4] Q. Guo, F. Zhuang, C. Qin, H. Zhu, X. Xie, H. Xiong, Q. He, A survey on knowledge graph-based recommender systems, IEEE Trans. Knowl. Data Eng. (2020).

[5] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, in: Proceedings of NAACL-HLT, 2016, pp. 260–270.

[6] Z. Huang, W. Xu, K. Yu, Bidirectional LSTM-CRF models for sequence tagging, 2015, arXiv preprint arXiv:1508.01991.

[7] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, J. Li, A unified MRC framework for named entity recognition, 2019, arXiv preprint arXiv:1910.11476.

[8] S. Kumar, A survey of deep learning methods for relation extraction, 2017, arXiv preprint arXiv:1705.03645.

[9] M. Cui, L. Li, Z. Wang, M. You, A survey on relation extraction, in: China Conference on Knowledge Graph and Semantic Computing, Springer, 2017, pp. 50–58.

[10] Z. Geng, G. Chen, Y. Han, G. Lu, F. Li, Semantic relation extraction using sequential and tree-structured LSTM with attention, Inform. Sci. 509 (2020) 183–192.

[11] 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: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 1227–1236.

[12] A. Katiyar, C. Cardie, Going out on a limb: Joint extraction of entity mentions and relations without dependency trees, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 917–928.

[13] Y. Yuan, X. Zhou, S. Pan, Q. Zhu, Z. Song, L. Guo, A relation-specific attention network for joint entity and relation extraction, in: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence, 2021, pp. 4054–4060.

[14] 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.

[15] Q. Li, H. Ji, Incremental joint extraction of entity mentions and relations, in: 52nd Annual Meeting of the Association for Computational Linguistics, 2014, pp. 402–412.

[16] X. Yu, W. Lam, Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach, in: Coling 2010: Posters, 2010, pp. 1399–1407.

[17] M. Miwa, Y. Sasaki, Modeling joint entity and relation extraction with table representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP, 2014, pp. 1858–1869.

[18] X. Ren, Z. Wu, W. He, M. Qu, C.R. Voss, H. Ji, T.F. Abdelzaher, 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.

[19] 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.

[20] Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu, L. Sun, Tplinker: Single-stage joint extraction of entities and relations through token pair linking, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020, pp. 1572–1582.
[21] H. Zheng, R. Wen, X. Chen, Y. Yang, Y. Zhang, Z. Zhang, N. Zhang, B. Qin, X. Ming, Y. Zheng, PRGC: Potential relation and global correspondence based joint relational triple extraction, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 6225–6235.
[22] 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, 2018, pp. 506–514.
[23] R. Li, D. Li, J. Yang, F. Xiang, H. Ren, S. Jiang, L. Zhang, Joint extraction of entities and relations via an entity correlated attention neural model, Inform. Sci. 581 (2021) 179–193.
[24] T. Chen, L. Zhou, N. Wang, X. Chen, Joint entity and relation extraction with position-aware attention and relation embedding, Appl. Soft Comput. 119 (2022) 108604.
[25] X. Li, Y. Li, J. Yang, H. Liu, P. Hu, A relation aware embedding mechanism for relation extraction, Appl. Intell. (2022) 1–10.
[26] D. Zelenko, C. Aone, A. Richardella, Kernel methods for relation extraction, J. Mach. Learn. Res. 3 (Feb) (2003) 1083–1106.
[27] Y.S. Chan, D. Roth, Exploiting syntactico-semantic structures for relation extraction, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, 2011, pp. 551–560.
[28] K. Xu, Y. Feng, S. Huang, D. Zhao, Semantic relation classification via convolutional neural networks with simple negative sampling, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 536–540.
[29] P. Zhou, W. Shi, J. Tian, Z. Qi, B. Li, H. Hao, B. Xu, Attention-based bidirectional long short-term memory networks for relation classification, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016, pp. 207–212.
[30] B. Xu, Q. Wang, Y. Lyu, Y. Shi, Y. Zhu, J. Gao, Z. Mao, Emrel: Joint representation of entities and embedded relations for multi-triple extraction, in: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2022, pp. 659–665.
[31] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, 2018, arXiv preprint arXiv:1810.04805.
[32] M. Lin, Q. Chen, S. Yan, Network in network, 2013, arXiv preprint arXiv:1312.4400.
[33] S. Riedel, L. Yao, A. McCallum, Modeling relations and their mentions without labeled text, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2010, pp. 148–163.
[34] C. Gardent, A. Shimorina, S. Narayan, L. Perez-Beltrachini, Creating training corpora for nlg micro-planning, in: 55th Annual Meeting of the Association for Computational Linguistics, 2017, pp. 179–188.
[35] K. Sun, R. Zhang, S. Mensah, Y. Mao, X. Liu, Progressive multi-task learning with controlled information flow for joint entity and relation extraction, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2021, pp. 13851–13859.
[36] X. Tian, L. Jing, L. He, F. Liu, Stereorel: Relational triple extraction from a stereoscopic perspective, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021, pp. 4851–4861.
[37] 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.