International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-021-01491-6

ORIGINAL ARTICLE

A joint extraction model of entities and relations based on relation decomposition

Chen Gao[1] · Xuan Zhang[1,2,3] · Hui Liu[1] · Wei Yun[1] · Jiahao Jiang[1]

Received: 27 February 2021 / Accepted: 6 December 2021

Abstract
Extracting entities and relations from unstructured text is an essential task in the field of information extraction. Existing work mainly pipeline extraction and joint decoding methods. However, these methods are unable to extract overlapping entities and relations, and ignore the task correlation between entity and relation extraction. In this paper, we first introduce the BERT pre-training model to model the text more finely. Then, we decompose the extraction into relation extraction and entity recognition. Relation extraction is transformed into a relation classification task. Entity recognition is transformed into a sequence labeling task. The recognition entity includes a head entity and a tail entity. We evaluate the model on the New York Times (NYT) and WebNLG datasets. Compared with most existing models, excellent results have been obtained. Experimental results show that our model can fully capture the semantic interdependence between the two tasks of entity and relation extraction, reduce the interference of unrelated entity pairs, and effectively solve the problem of entity overlap.

Keywords Relation decomposition · BERT · Attention · Joint extraction · Entities and relations

1 Introduction

As an essential part of information extraction, triple extraction acquires structured knowledge in the form of (head entity, relation, tail entity) from a set of unstructured texts, which is also called joint entity and relation extraction. This is one of the critical tasks for building a knowledge graph, and an essential foundation for other related natural language processing(NLP) tasks, such as machine translation, text summarization, recommendation systems, etc.

Early extraction methods mostly used pipeline-based methods [1, 2] for entity and relation extraction. Such methods regard the extraction task as two independent subtasks called named entity recognition [3] and relation classification [4]. First, all the entities in the sentence are identified, and then paired and classified. These methods are flexible and simplify the processing flow, but have some disadvantages. The first is the accumulation of errors. The entity identified in the named entity recognition task cannot always be guaranteed to be the correct one. The relation extraction based on the wrong entity obtained in the previous task will lead to incorrect transmission and accumulation. Errors in entity recognition will affect the performance of the next step of relation extraction. The second is entity redundancy. Since the extracted entity pairs are recognized first, and then the relation is classified. The redundant information brought by candidate entities without relation will increase the error rate and computational complexity. Finally, there is a lack of interaction. The pipeline-based methods ignore the internal connection and dependency between entity recognition and relation extraction tasks.

To solve the shortcomings of the pipeline extraction method, joint extraction methods that simultaneously extract entities and relations are proposed. The initial joint extraction methods are mostly feature-based models [5–8]. These models require a complicated preprocessing process and rely on feature extraction tools, which is not only complicated, but also easy to introduce errors. In order to reduce the manual process of feature engineering, neural networks are used for end-to-end joint extraction of entities and relations.

✉ Xuan Zhang
zhxuan@ynu.edu.cn

[1] School of Software, Yunnan University, Kunming 650091, Yunnan, China

[2] Key Laboratory of Software Engineering of Yunnan Province, Kunming 650091, Yunnan, China

[3] Engineering Research Center of Cyberspace, Kunming 650091, Yunnan, China

Published online: 24 January 2022

According to different modeling objects in the end-to-end joint extraction method, the joint extraction method is divided into a joint decoding method and a parameter sharing method. The joint decoding method adopts a new labeling strategy to uniformly label entities and relations. It converts the original joint learning model including entity recognition and relation classification into a sequence labeling problem [9]. Zheng et al. [10] proposed a novel marking scheme, which transforms the joint extraction of entities and relations into a marking task, and introduces the principle of nearest matching to extract multiple triples contained in a sentence. The parameter sharing method performs joint learning by sharing the encoding layer parameters of the joint model so as to realize the mutual dependence between entity recognition and relation extraction tasks. In [11–13], Tree-LSTM for relation classification and Bi-LSTM for entity recognition were used. They used parameter sharing to jointly extract entities and relations. Katiyar et al. [14] proposed a pointer network that extracted relations while identifying entities, instead of using dependency trees. The end-to-end joint extraction method uses the interactive information between entities and relations to recognize entities and classify relations between entity pairs, which solves the problems caused by the pipeline method.

The traditional joint entity and relation extraction scheme only considers the case of extracting a triple in a sentence. But in fact, the sentences we extract often contain multiple triples, and these triples may also have overlap entities and relations. Zeng et al. [15] divided the triples in the text into the following three categories based on the overlapping content, namely Normal, SingleEntityOverlap (SEO) and EntityPairOverlap (EPO). As shown in Fig. 1, Normal means that all triples in a sentence have no overlapping entities. SEO means there are overlapping entities, but no overlapping entity pairs. EPO means there are overlapping entity pairs.

To solve all the above problems, a new end-to-end neural network model based on relation decomposition is designed in this paper. The model we proposed is a multi-task joint learning model based on parameter sharing. The entity and relation extraction loss can be optimized jointly by sharing the parameters of the encoding layer and using a cost function, which strengthens the information interaction between entities and relations. Compared with the traditional pipeline-based method, this can alleviate the problem of error accumulation to a certain extent. Our goal is to extract all triples, including those with overlapping entities. We integrate relation features into sentences to generate vector representations of each sentence in different relations, and then perform sequence labeling to extract its corresponding head and tail entities. Our model mainly consists of the following parts. First, the sentence features are coded through the encoding layer. Then according to the sentence vector representation obtained by the encoding layer, the relationship between the sentences is classified, and the relation contained in the sentence is obtained. At last, a relation is randomly selected and its features are merged into sentence features. The head and tail pointers are used to mark the sequence under the specified relation, and the corresponding head and tail entities are obtained. In summary, the main contributions of this paper are as follows:

1. We propose an end-to-end neural network model to accomplish the joint extraction of entities and relations by transforming the triple extraction problem into multiple sequence label problems. Multiple triples can therefore be extracted efficiently.
2. Our model is a joint extraction scheme of entities and relations based on relation decomposition. In our method, vector representations of sentences are constructed under different relations, and relational decomposition strategies are used to extract triples under different relations. The problem of triple overlap can be solved effectively.
3. The model uses a BERT encoder based on the transformer structure and adds an attention layer to better

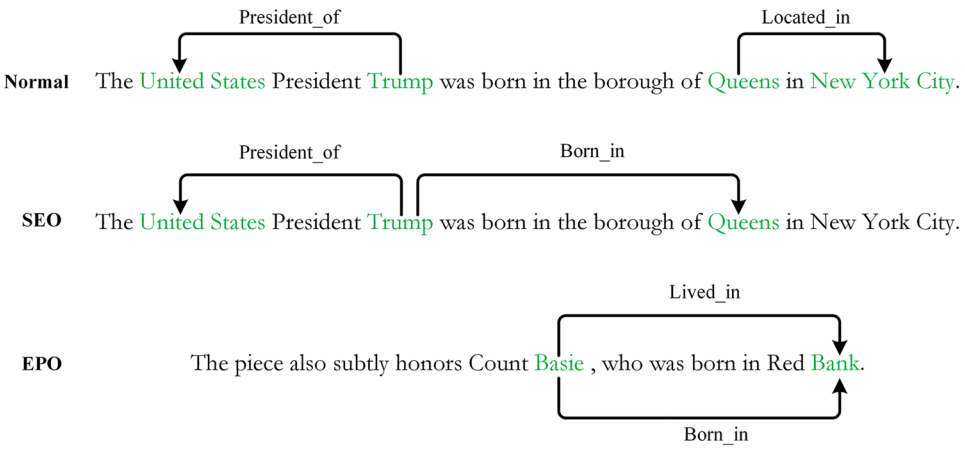

Fig. 1 Examples of Normal, EntityPairOverlap (EPO) and SingleEntityOverlap (SEO)

encode sentences to improve performance. We conduct experiments on two widely used public datasets. Experimental results show that our model has reached better performance and F1 score exceeds 90%.

2 Related work

With the rise of knowledge graphs in the field of NLP, entity-relation extraction, as the most important step in building a knowledge base, has become a research hotspot. At present, entity-relation extraction frameworks based on supervised learning are mainly divided into the following two types: pipeline extraction methods and joint extraction methods. The pipeline-based method divides entity-relation extraction into two steps: named entity recognition and relation extraction. The pipeline-based method uses two completely independent encoders to construct the input of the relational model according to the entity recognition model [16]. As a fundamental component of information extraction tasks, named entity recognition plays a very important role in other natural language tasks. Early named entity recognition tasks were mainly feature-based methods [17]. With the rise of deep learning, methods based on the BiLSTM-CRF [18] model became the mainstream model. The relation classification is to classify entity pairs into specific relation categories on the premise that entities in the sentence are known. Similar to named entity recognition, with the exception of feature-based methods [2], the mainstream is still neural network-based methods. In [19–21], CNN was used to classify relations. Then, Zhou et al. [22] introduced an attention layer based on the BiLSTM model to better encode sentences and improve the performance of relation classification. Zhang et al. [23] propose an attention-based model to extract the multi-aspect semantic information for the Chinese medical relation extraction by multi-hop attention mechanism. The pipeline method refines the entity relation extraction task, making the entire task process clearer, but it also causes some problems. It mainly includes: error accumulation, entity redundancy and missing interaction.

The joint extraction method [24] used a single model to simultaneously extract entities and relations. Based on the joint learning method, these two subtasks are projected into a structured prediction framework, or multi-task learning is performed through a shared encoding layer. It overcomes the shortcomings of the pipeline method, but also faces new challenges. The early joint extraction methods were mainly feature-based. However, due to the complexity of entity-relations extraction tasks, feature engineering is not an easy task, and the efficiency is low. Researchers turned their attention to the end-to-end joint extraction method based on neural networks, which can be divided into two categories: joint decoding and parameter sharing. Based on the joint decoding method, the entity and relation extraction are transformed into a sequence labeling task, and the interaction between the entity model and the relation model is enhanced through an overall label. Zheng et al. [10] proposed a novel labeling scheme, which converts the extraction of the relation between the sequence labeling task and the classification task into a labeling problem. They identify multiple triples contained in the sentence by decoding the labels. Based on the work of [10], Zhou et al. [24] introduced pre-trained entity features and attention mechanisms into the model to improve the performance of model recognition. Meng et al. [25] proposed to use a new method to obtain character features and integrate features into model training. To solve the problem of overlapping triples, Luo et al. [26] improved the marking scheme and defined rule constraints for decoding.

The parameter sharing method realizes joint extraction by sharing input features or internal hidden layer states. During training, the loss of named entity recognition and the loss of relation extraction are added together for optimization. Miwa et al. [11] first proposed the application of neural networks to the end-to-end entity-relations joint extraction model. The model is mainly composed of two parts: word sequence and tree structure, sharing the parameters of entity recognition and relation extraction. Subsequently, Katiyar et al. [14] did not use the dependency tree structure. They proposed a joint extraction method based on the pointer network structure, which extracts relations while identifying entities. In [27, 28], a multi-head selection mechanism was used to extract multiple triples in a sentence.

The above methods all encounter challenges in solving the problem of extracting overlapping triples. Zeng et al. [15] was the first to use a neural network model based on the copy mechanism of the sequence to sequence learning framework to extract overlapping triples. It extracts triples in three steps, sequentially extracting relations, head entities, and tail entities.Their method only considers the triples composed of a single token dimension entity, and it is unable to solve the problem of multi-dimensional token entity triples extraction. Subsequently, they applied reinforcement learning to the sequence-to-sequence model to improve the recognition effect of overlapping triples [29]. Dai et al. [30] proposed a position attention mechanism for the problems of overlapping triples and difficulty in modeling long-distance relations. They generate different position-aware sentence representations according to the query position, which can be used to decode different tag sequences and extract overlapping triples. The model needs to encode a sentence n times, n is the sentence length, so the time complexity of decoding is $O(n^2)$. For our model, the number of relations contained in a sentence is generally less than the number of entities, which can effectively reduce the computational time complexity. In addition, Zeng's method [15] has two main problems.

One is that the model is difficult to predict multi-token entities, and the other is that the model is very weak in distinguishing between head and tail entities, which lead to the problem of inaccurate entity extraction. To solve these two problems, Zeng et al. [31] proposed a multi-task learning framework equipped with copy mechanism. It better extracts the relational triples in the sentence. Yu et al. [32] decomposed triple extraction into two steps: head entity extraction and tail entity and relation extraction. The model first recognizes the head entity in the sentence. Then, the vector corresponding to the head entity and the sentence vector are spliced. At last, the tail entity under the corresponding relation of the head entity is recognized. Since the tail entity is labeled according to the type label of the relation, it is impossible to identify triples with overlapping relations. The other two related works are [33, 34]. Liu et al. [33] proposed an attention-based joint model and designed a supervised multi-head self-attention mechanism as a relation detection module to learn the token-level correlation of each relation type. Wei et al. [34] proposed a new framework to extract triples. It models the relation between the subject and the object in the sentence to solve the overlap problem. Ye et al. [35] proposed a novel model, contrastive triple extraction with a generative transformer. Their model introduced a single shared transformer module for an encoder-decoder-based generation. The model we proposed first extracts the relation in the sentence, and extracts the head entity and the tail entity according to different relations. The number of relations contained in a sentence is generally less than the number of entities, which can effectively reduce the computational time complexity. In addition, our model can effectively solve the three overlapping problems in entity relation extraction.

3 Proposed method

In this section, we first introduce the tagging scheme of the model. The tagging content is divided into relation tag and entity tag. Subsequently, we describe the joint extraction model based on relation decomposition in detail.

3.1 Tag scheme

We propose an entity tagging scheme based on relation decomposition, using BIO tag annotations, which can reduce the time complexity of decoding. As shown in Fig. 2, our tagging scheme is mainly composed of relation extraction and relation-based entity recognition. The relation extraction module mainly contains a classifier. The relation-based entity recognition module mainly contains two classifiers, which are used to mark the head and tail entities. For a sentence with multiple triplets, we generate separate tag sequences according to different relations. In the tag sequence of a certain relation, only its corresponding head and tail entities are annotated, while the rest of words are assigned with label O. However, the extracted entities are only the boundaries of the entities, and there is no information about the types of entities. Given a sentence, if the sentence contains k relation categories, $2k+1$ tagged sequences are generated.

3.2 Joint extraction model

The joint entity and relation extraction model consists of three modules: encoder, relation extraction and entity recognition, as shown in Fig. 3. Our model decomposes the extraction of triples into two subtasks: relation extraction

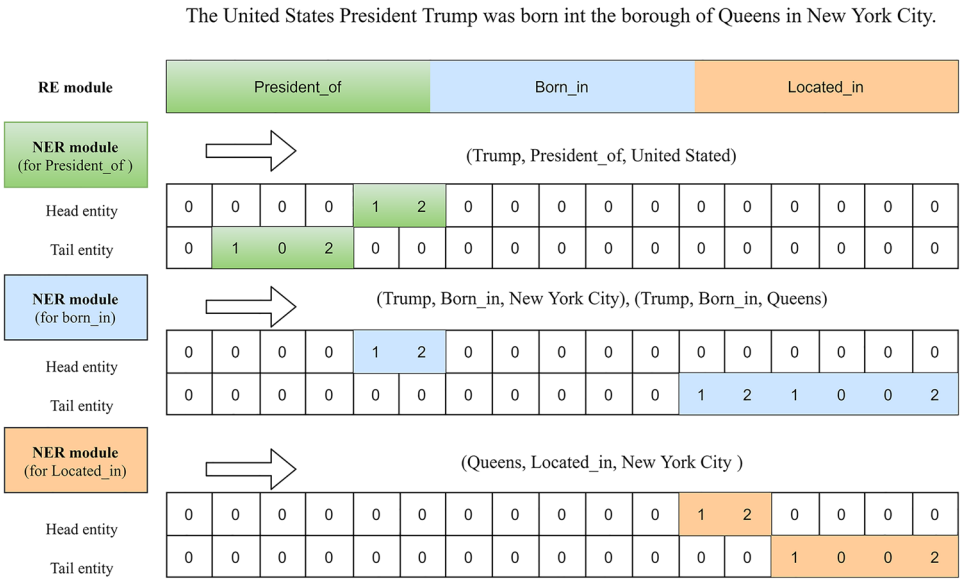

Fig. 2 Tag scheme based on relational decomposition

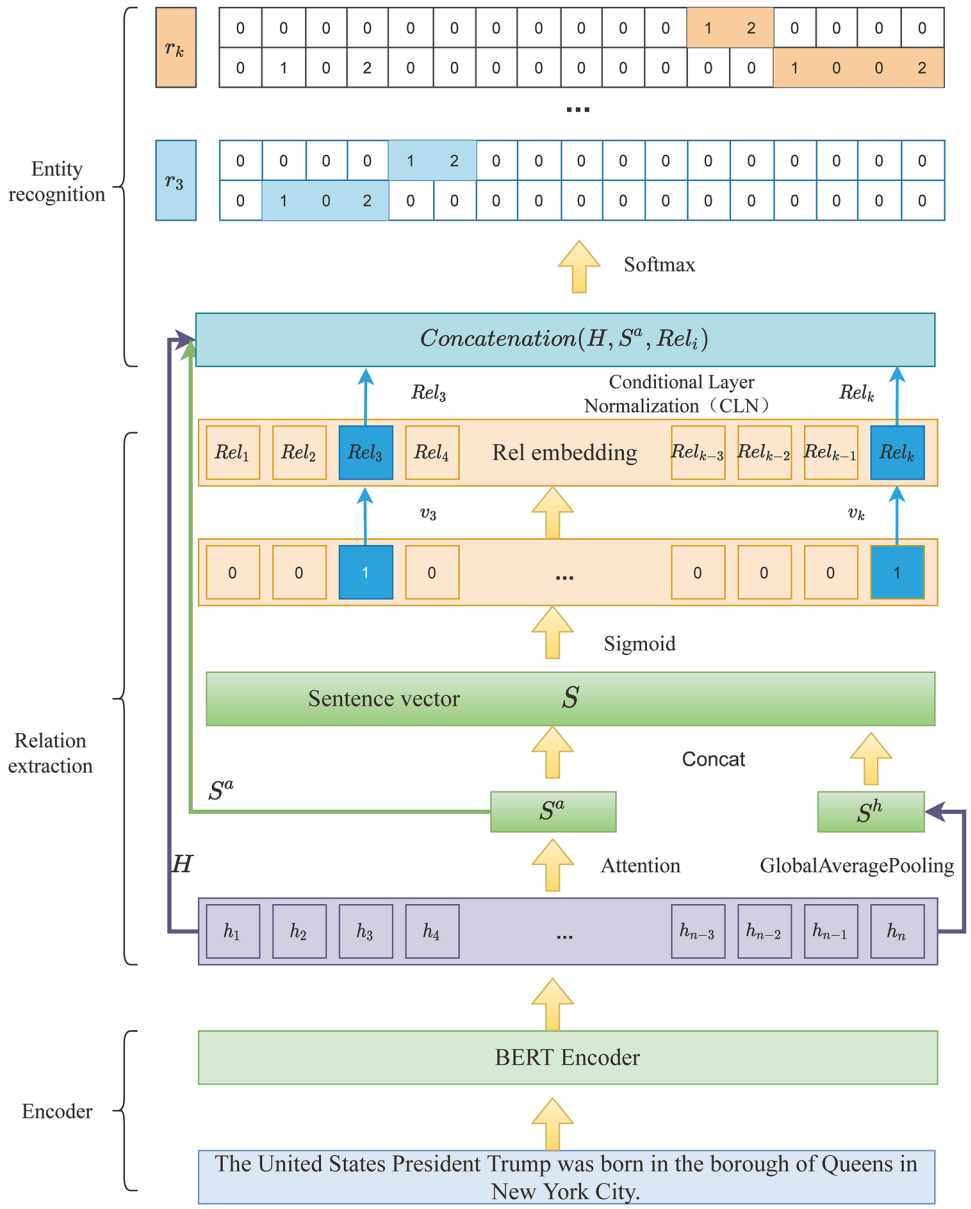

Fig. 3 The overall structure of the joint extraction model based on relation decomposition. The figure shows the steps of entity recognition based on different relations

and entity recognition. By sharing the encoding layer, during training, both subtasks will update the shared parameters of the encoding layer through the backward propagation algorithm to achieve two subtasks. The interdependence between tasks finally finds the best parameters of the global task. The input text is first generated by an encoder to generate a Word-based text vector representation. Then, the obtained Word-based text vector representation is pooled to reduce the dimensionality, thereby obtaining the Sentence-based text vector representation. The attention mechanism is introduced in the process of sentence vector generation to capture the importance of different words in sentence classification. We combine these two parts to get a new Sentence-based text vector representation, so as to perform multi-relation classification. Finally, the specific relation vector and the Word-based text vector representation are combined to identify the entities under the specific relation.

The specific method is based on the encoding vector of relation as the condition, and the Conditional Layer Normalization (CLN) [36] is applied to the encoding sequence. In Transformer models, such as BERT, the main Normalization method is LayerNormalization. Therefore, we can turn the corresponding β and γ into a function of input conditions to control the generation behavior of the Transformer model. This is the clue idea of Conditional Layer Normalization. For the pre-trained model, there are ready-made, unconditional β and γ, and they are all vectors of fixed length. We can transform the input conditions to the same dimensions as β and γ through two different transformation matrices, and then

add the two transformation results to β and γ respectively. In our model, the relational encoding vector and sentence attention encoding vector are respectively used as conditions, and a conditional LayerNormalization is performed on the encoding sequence. This scheme has a better expressive ability.

As shown in Fig. 3, the BERT pre-training model is first used to encode the text. BERT [37] is a language representation model based on the structure of transformer [38]. There are three inputs in BERT layer, namely: token embedding, segment embedding, and position embedding. In order to meet the input conditions of BERT, each input word is processed by the 'Wordpiece' operation before entering the BERT layer. The 'Wordpiece' operation inserts '[unused1]' as a separator between words, and embedding [CLS] and [SEP] at the beginning and end of the sentence. The purpose of using 'Wordpiece' is to divide words into smaller units, compress the size of the vocabulary, and better handle unknown words. Therefore, it reduces the size of the vocabulary list, thereby speeding up the training, and generating higher quality word embedding. Based on our marking method, the introduction of "[unused1]" to separate vocabulary is to avoid the overlap of entity head tags and tail tags composed of a single word. BERT's special vocabulary retains some special marks, such as "[unused1]" and so on. These tags are not trained and initialized randomly, and will not affect the result of sentence embedding. The preprocessed token is converted into a fixed-dimensional vector in the token embedding. It should be noted that entities and ordinary words together constitute a vector representation of a sentence, without special tags. The label of the entity is only used for the calculation of the loss function and is not contained in training.

Given the initial input sequence $W = [w_1, w_2, ..., w_m]$, the new sequence $X = [x_1, x_2, ..., x_n]$ is obtained by using 'Wordpiece' operation.

$$x_t = \text{Wordpiece}\left(\mathrm{w}_i\right) \quad t \in [1,n], i \in [1,m] \tag{1}$$

Then, the pre-trained BERT model is finetuned to encode context information to generate sentence sequence embeddings $H = [h_1, h_2, ..., h_n]$.

$$h_t = \mathrm{BERT}\left(x_t\right) \quad t \in [1,n] \tag{2}$$

where $h_t \in \mathbb{R}^{d_\omega}$ and d_ω represents the dimension of the hidden state of BERT. We use $H = [h_1, h_2, ..., h_n]$ to represent sentence features based on the context level of the word. It should be noted that the words after 'Wordpiece' operation are decomposed into multiple tokens, and the phrases are recombined into the original words in the final encoding.

3.2.1 Relation extraction

To extract relations from sentences, the overall feature representation of the sentence is obtained based on the word-based sentence features. Here we use the GlobalAveragePooling(GAP) [39] method to reduce the dimension of the word-based sentence feature $H = [h_1, h_2, ..., h_n]$ to obtain the overall feature representation of the sentence S^h, that is, the dimension changes from $[batchSize, seqLen, d_\omega]$ to $[batchSize, d_\omega]$. Attention neural networks have achieved success in many tasks such as machine translation, speech recognition to image recognition. The importance of each word in a sentence is different. By introducing an attention mechanism [40], relations can be classified better. The sentence vector representation S^a, which is incorporated into the attention mechanism, is formed by the weighted sum of these word-based sentence feature vectors $H = [h_1, h_2, ..., h_n]$. Finally, S^h and S^a are conditionally merged to obtain the final sentence feature vector representation S. The specific calculation method is shown in the following formulas:

$$S^h = \text{GAP (H)} \tag{3}$$

$$\mathrm{M} = \tanh(\mathrm{H}) \tag{4}$$

$$\alpha = \mathrm{softmax}\left(\omega^T M\right) \tag{5}$$

$$S^a = \mathrm{H}\alpha^T \tag{6}$$

$$\mathrm{S} = \mathrm{concat}\left[S^h, S^a\right] \tag{7}$$

where $H \in \mathbb{R}^{d_\omega \times n}$, S^h, S^a, $S \in \mathbb{R}^{d_S}$, d_ω represents the dimension of the BERT hidden state, d_S represents the dimension of the sentence vector feature, and ω is the training parameter vector, ω^T is transposed. The dimensions of ω, α are d_ω, d_α.

After obtaining the final vector feature representation of the sentence, we can classify its relation. The specific formula is as follows:

$$v_j = \sigma\left(W_1 \mathrm{S} + b_1\right) \tag{8}$$

where $W_1 \in \mathbb{R}^{d_k \times d_\omega}$, $b_1 \in \mathbb{R}^{d_k}$, k represents the total number of relation categories, and σ represents the sigmoid activation function. This function returns a value in the range of 0 to 1, which can be used as a threshold for us to judge whether the specified relation exists. According to the formula, we can get all the relation types contained in the sentence. Then, the vector representation of the specified relation is transformed based on the previously initialized relation embedding and the current relation category.

Our model first recognizes the relation in the sentence. If there is a relation in the sentence, it will perform entity recognition based on the specific relation. For sentences that have no relation, the sentence does not contain valid triples, so entity recognition will not be performed. In the training set, all sentences contain relations, but the number of relations they contain is not limited.

3.2.2 Entity recognition

Entity recognition is performed as a sequence labeling task. As a pre-trained network model, BERT has a very powerful fitting ability. In the process of decoding and operation, the marking scheme based on the head and tail pointer is more concise than the traditional CRF marking method, which can greatly reduce the time and space overhead. There have been many papers, such as [32, 34], using pointer network-based structures to annotate and decode entities. Considering a sentence often contains multiple relations and the triples are different in each relation, the entity recognition process is decomposed according to different relations. Since the number of triples under each relation is greatly reduced and the overlap is also reduced, our method can solve the problem of triple overlap. Through the previous relation classification module, we obtain the relation category contained in the sentence. Therefore, we first embed the relation to generate the vector representation of all the relation categories $Rel = [Rel_1, Rel_2, ..., Rel_k]$. A relation label is generated based on the relation contained in the sentence, that is, its vector representation (0, 1). The relation vector is input into the embedding layer to obtain the word embedding representation of the relation. Then, the corresponding relation vector is obtained based on the identified relation category. Next, the sentence and a specific relation vector representation are combined to generate a relation-based sentence vector representation. At last, performing entity recognition under a specific relation. It should be noted that during the training process, we randomly extract a relation in the sentence and combine it with the sentence each time. In the process of prediction, sentences are copied according to the number of relations in the sentence, and each sentence is combined with different relations. The purpose of randomly extracting relations is to better simplify the training process. Each sentence contains multiple relations, and there may be multiple triples under the same relation. For each sentence, one of the relations is randomly selected as the training set for each training. By increasing the number of training rounds, all the relations will be added to the training set. All relation embeddings are initialized parameters randomly, and then the relation extraction is performed according to the sentence vector we constructed. According to the extracted relation and the previous relation embedding, the vector representation of the specific relation is obtained. The specific formulas is described as follows:

$$h_i^j = \mathrm{CLN}\left(h_i, \mathrm{Rel}_j\right) \quad i \in [1, n], j \in [1, k] \tag{9}$$

$$o_i = \mathrm{CLN}\left(h_i^j, S^a\right) \tag{10}$$

$$P\left(y_i^{\mathrm{head}}\right) = \mathrm{Softmax}\left(W^{\mathrm{head}} \cdot o_i + b^{\mathrm{head}}\right) \tag{11}$$

$$P\left(y_i^{\mathrm{tail}}\right) = \mathrm{Softmax}\left(W^{\mathrm{tail}} \cdot o_i + b^{\mathrm{tail}}\right) \tag{12}$$

$$\mathrm{head}\left(x_i\right) = \arg\max_{\mathrm{tagNum}} P\left(y_i^{\mathrm{head}}\right) \quad \mathrm{tagNum} = 3 \tag{13}$$

$$\mathrm{tail}\left(x_i\right) = \arg\max_{\mathrm{tagNum}} P\left(y_i^{\mathrm{tail}}\right) \quad \mathrm{tagNum} = 3 \tag{14}$$

where W^{head}, $W^{head} \in \mathbb{R}^{d_\omega \times k}$, b^{head}, $b^{tail} \in \mathbb{R}^{d_k}$, k represents the number of categories of the relation, Rel_j represents the relation vector representation combined with the current sentence, and *tagNum* represents the number of tag categories, including three types of 0, 1, and 2. $P(y_i^{head})$ and $P(y_i^{tail})$ respectively represent the probability that the *i-th* character is predicted to be the head entity and tail entity label under the condition of the relation Rel_j.

Since our model contains two parts: relation extraction and entity recognition, the training loss of the model also consists of two parts, namely relation classification loss function and entity recognition loss function. The entity recognition loss part includes the head entity loss and the tail entity loss. The training loss of the model $\mathcal{L}_{\mathrm{model}}$ (to be minimized) is defined as the sum of the relation label of the predicted distribution and the negative log probability of the entity, as shown below:

$$\begin{aligned}\mathcal{L}_{\mathrm{model}} = &-\frac{1}{n}\sum_{i=1}^{n} \log P\left(v_j = \hat{v}_j\right) + \log P\left(y_i^{\mathrm{head}} = \hat{y}_i^{\mathrm{head}}\right) \\ &+ \log P\left(y_i^{\mathrm{tail}} = \hat{y}_i^{\mathrm{tail}}\right)\end{aligned} \tag{15}$$

3.3 Decoder

In reasoning, to adapt to the task of multi-object extraction, a multi-span decoding algorithm is proposed to combine relation decomposition and rules, as shown in *Algorithm 1*. Relation extraction and entity recognition are two stages in the algorithm. According to the specific analysis of the dataset, the following rules are defined: 1. Head and tail entities in the same triplet cannot contain each other. 2. The

length of the head and tail entities is limited, which cannot be empty or exceed 5.

In the relation classification stage, the sentence vector representation is obtained based on the input S, and the relations are identified therein (Line 2). When all the relations are obtained, a new sentence representation is generated based on each different relation and the entities are identified in the sentence (Line 11). At last, match according to the number of head and tail entities to obtain triples (Line 13, 17, 20).

In the decoding stage, for a particular relation, if there are different head entities and tail entities, below is our combination plan. First, there is only one head entity and one tail entity, the combination and relation between the two entities directly form a triple. Then, if there is one head entity and multiple tail entities or one tail entity and multiple head entities, the principle of one-to-many is combined. Finally, if there are multiple head entities and multiple tail entities, we will match them according to the closest matching principle, and each entity is matched only once. For the example sentence "Trump was born in New York and Obama in Hawaii". This sentence contains two triples, they are (Trump, born in, New York) and (Obama, born in, Hawaii). According to the model we proposed, based on the relation "born in", the head entity "Trump, Obama" and the tail entity "New York, Hawaii" can be identified. In the decoding process of our model, this is the third case, that is, multiple head entities and tail entities. They are combined in pairs according to the principle of closest distance matching. The distance from Trump to New York is 3, the distance from Trump to Hawaii is 8, the distance from Obama to New York is 2, and the distance from Obama to Hawaii is 1. Finally, two sets of triples can be extracted, namely (Trump, born in, New York) and (Obama, born in, Hawaii).

Algorithm 1 The Multi-span decoding

Input: Sentence: S, Relation types: k, Length of sentence after Wordpiece: m
Output: A list of Tuple=()

1: **initialize**: RelationList,SubjectList,ObjectList
2: Obtain idx = RelTag(S) by Eq.(8)
3: **for** $i = 0; i < k; i++$ **do**
4: **if** $idx[i] == 1$ **then**
5: RelationList.append(i)
6: **end if**
7: **end for**
8: **for** $idx_R = 1; idx_R < len(RelationList); idx_R++$ **do**
9: Obtain HeadTag(S) by Eq.(13) and TailTag(S) by Eq.(14)
10: **for** $idx_E = 1; idx_E < m; idx_E++$ **do**
11: Obtain SubjectList,ObjectList
12: **end for**
13: **if** $len(SubjectList) == 1$ **then**
14: $Head\ entities\ match\ all\ tail\ entities \rightarrow (s, r, o)$
15: $Nearest\ match\ principle \rightarrow (s, r, o)$
16: **end if**
17: **if** $len(ObjectList) == 1$ **then**
18: $Tail\ entities\ match\ all\ head\ entities \rightarrow (s, r, o)$
19: **end if**
20: **if** $len(SubjectList) > 1$ and $len(ObjectList) > 1$ **then**
21: $Nearest\ match\ principle \rightarrow (s, r, o)$
22: **end if**
23: Tuple.append((s,r,o))
24: **end for**
25: **return** Tuple

4 Experiments

In this section, we first describe the datasets used in the experiment, then explain the design process of the experiment in detail, and finally analyze the experimental results.

4.1 Datasets

Our model was evaluated on two widely used public datasets, namely: NYT[41] and WebNLG [42]. NYT is a large-scale dataset constructed based on the 'New York Times' news corpus using a distant supervision method. This method automatically aligns the knowledge base and text to generate large-scale training data. The NYT dataset contains a total of 66,194 sentences and 24 types of relation categories. Among them, 56195 sentences are used as the training set, 4999 sentences are used as the verification set, and the remaining 5000 sentences are used as the test set. The data of WebNLG is derived from articles in Wikipedia. The standard datasets are constructed according to the manual annotation by the annotator. It contains a total of 6222 sentences and 246 types of relation categories. The statistics of these two datasets are shown in Table 1.

In addition to statistics on the overall distribution of the datasets, we also divide the two datasets into three categories based on the overlap of the triples, namely: Normal, SEO, EPO. This is to show how our model can be used to deal the problem of overlapping triples. If none of the triples in a sentence has overlapping entities, the sentence belongs to the Normal class. If the entity pairs

Table 1 Statistics of the datasets

Dataset	NYT	WebNLG
Training set	56195	5019
Test set	5000	500
Dev set	4999	703
Relations	24	246

of two triples are the same but the relation is different, the sentence will be added to the EPO category. If some triples of a sentence have overlapping entities, but these triples do not have overlapping entity pairs, then the sentence belongs to the SEO category. It should be noted that a sentence in the EPO category may contain multiple Normal and SEO triples. The specific classification of the data is shown in Table 2.

4.2 Environment settings

Our experiment runs on the Windows 10 operating system, uses the Keras 2.24 framework, and uses Python for model building and training. For BERT, We used a pre-trained BERT model and set it according to the configuration file provided by [38]. It has a 12-layer Transformer structure and 110M parameters. The sentence embedding dimension is 768. The relation embedding dimension is 200. We use the Adam optimizer to train the model. The batch size is 8. The dropout rate is 0.5. The initial learning rate is set to 1e−5. The learning rate attenuation is 0.96^n.

Precision, recall, and F1 score are used to evaluate our model. In the process of triple evaluation, only when the head entity, tail entity, and relation are exactly the same as the gold standard annotations, the extracted triple is correct. When the development set achieves the best results, the corresponding results on the test set are recorded.

Table 2 Overlapping triple data statistics

Category	NYT		WebNLG	
	Train	Test	Train	Test
Normal	37013	3266	1596	182
SEO	14735	1297	3406	318
EPO	9782	978	227	16
All	56195	5000	703	500

4.3 Experimental results

4.3.1 Performance comparison

On two publicly available datasets, the performance of our model is compared with the following models:

- **NovelTagging (2017)** [10]: This model proposes an entity relation extraction method based on joint decoding
- **CopyRe (2018)** [15]: This model proposed a sequence labeling method for entity copy mechanism, which predicts whether each label in the original sentence will be copied (1 or 0), which can solve the problem of triple overlap. We present the experimental results using Multi-Decoder.
- **GraphRel(2019)** [43]: This model constructs a complete word graph for each sentence and uses GCN to predict the relation between all word pairs.
- **DpModule(2020)** [44]: This model proposes an end-to-end model with a dual pointer module that can extract the entire entity and relation together.
- **WDec (2020)** [45]: This model designed a new representation scheme and used the seq2seq model to generate triples with the entire boundary.
- **HBT(2020)** [32]: This model uses a segmentation extraction and decoding method to solve the extraction of overlapping triples, which can effectively solve the problem of triple SPO.
- **CASREL (2020)** [34]: This model proposes a new framework to extract triples. It models the relation between the subject and the function of the object in the sentence, thereby naturally solving the overlap problem.

The experimental results are shown in Table 3. According to the data in the table, our model always obtains better performance in the two datasets. Compared with the earlier joint extraction method using NovelTagging, the F1 score of our model on the NYT and WebNLG datasets increased 48.7% and 63.5%, respectively. We

Table 3 Results of models compared on NYT and WebNLG

Model	NYT			WebNLG		
	Precision	Recall	F1	Precision	Recall	F1
NovelTagging	62.4	31.7	42.0	52.5	19.3	28.3
CopyRe	61.0	56.6	58.7	37.7	36.4	37.1
GraphRel	63.9	60.0	61.9	44.7	41.1	42.9
DpModule	72.8	69.0	70.9	38.7	37.5	38.1
HBT	85.5	71.7	78.0	84.3	82.0	83.1
WDec	88.1	76.1	81.7	84.8	64.9	73.5
CASREL	89.7	89.5	89.6	**93.4**	90.1	91.8
Ours(LSTM)	84.4	73.4	78.5	89.3	80.5	84.7
Ours	**91.5**	**90.0**	**90.7**	91.4	**92.2**	**91.8**

Bold highlights the superiority of the indicator

analyze the reasons for the large experimental gap. First of all, our model uses the BERT encoder based on the transformer structure. Compared with the traditional feature extractor based on a recurrent neural network, our model captures more text semantic information and gets a better representation of the text. Since NovelTagging is glove-based model, comparisons based on LSTM have also been added. LSTM is used as the network structure for feature extraction, using glove embedding instead of BERT encoder in the embedding layer. The experimental results still show that our model has better performance. However, the comparison of the LSTM part of the experiments can reveal a poor effect boost on NYT. LSTM has a limited ability to extract sentence features. In the dataset NYT, there are a large number of triple entities in the training set. After a relation is extracted by the model, some unrelated entities are identified based on the relation. These entities generate a lot of entities based on the recent matching principle. Secondly, the NovelTagging model uses a joint decoding scheme to extract overlapping triples of sentences, and the nearest matching principle in the decoding process is used. However, since the entity in the sentence has only one label, this solution is unable to solve the problem of overlapping entities in the sentence. According to our statistics on the two public datasets, they both contain a large number of overlapping triples. Finally, our proposed model extracts corresponding entities based on different relations, which solves the SEO and EPO problems well, and therefore obtains a higher F1 score.

The relatively new joint extraction methods are HBT, WDec and CasREL. Compared with HBT and WDec, the F1 value of our model on the two datasets increased by 12.7%, 9.0%, 8.7%, 18.3%, respectively. Both HBT and WDec have low recall rates. Due to the segmented labeling structure, HBT cannot extract EPO type triples. WDec uses the seq2seq model to generate triples and removes repeated triples and fragment triples through post-processing. Therefore, WDec achieves high precision on both datasets, but the recall is relatively low. CASREL, like our model, also uses a BERT encoder based on the transformer structure. This model expresses the relation as a mapping function between the subject and the object in the sentence, thereby solving the problem of overlapping triples. Our model extracts triples based on relation decomposition. When there are a large number of relations in a sentence, the model needs to learn more features to correctly classify the relations. When the number of triples in a sentence increases, the number of relations also increases. Therefore CASREL performs better on more triples.

Table 4 An ablation study of model on the NYT dataset

Model	Precision	Recall	F1
Base model	91.5	90.0	90.7
-Attention	90.6	89.3	89.9
-CLN	89.7	88.1	88.9

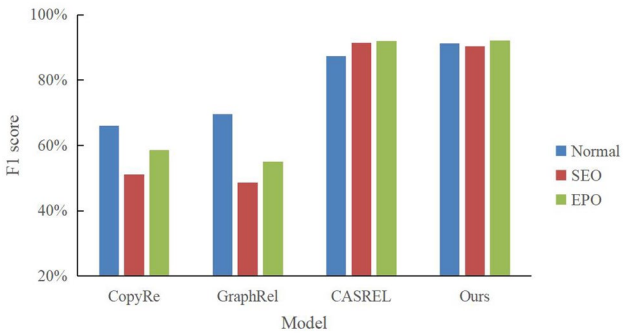

Fig. 4 Comparison of different sentence types according to the degree of overlap(NYT)

4.3.2 Ablation study

To prove the effectiveness of the proposed entity and relation detection module, we conduct the ablation study. As shown in the below Table 4, the introduction of the attention mechanism can better encode sentences, thereby improving the effect of relation and entity recognition. in addition, the connection method of CLN is better than ordinary concentration, which can integrate more sentence information and have more good presentation skills.

4.3.3 Analysis of different sentence types

To verify the performance of our model on the problem of overlapping triples, we conducted further experiments on the NYT dataset. According to the experimental results disclosed by each model, we have selected the following three models as the baseline model for comparison, namely: CopyRe, GraphRel and CASREL. Figures 4 and 5 show the experimental results of each model on Normal, SPO and EPO. Our model achieved better experimental results in dealing with overlapping problems, which is much better than CopyRe and GraphRel. According to the experimental results, CASREL is better than our model in some tests. But from the overall F1 value, our model is still slightly better than CASREL. The reason why GraphRel achieves poor results is that it predicts the relation of all word pairs. The relational classifier is overwhelmed by redundant candidates, resulting in the problem of entity pair redundancy CopyRe uses a seq2seq model to decode overlapping

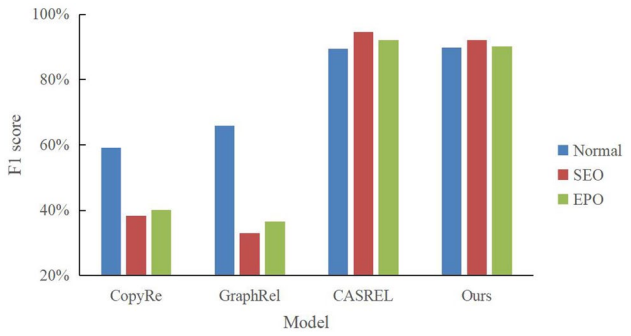

Fig. 5 Comparison of different sentence types according to the degree of overlap(WebNLG)

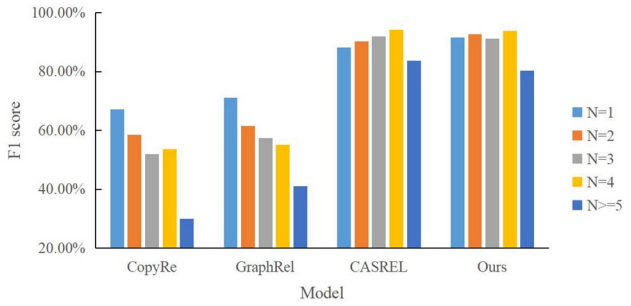

Fig. 6 Comparison of different sentence types according to the degree of overlap(NYT)

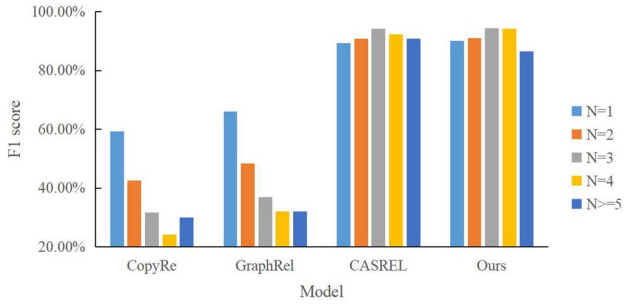

Fig. 7 Comparison of different sentence types according to the degree of overlap(WebNLG)

relations. However, it can only decode the first word entity of multiple words, while our model can detect the complete word sequence.

According to the size of the number of triples contained in the sentence, we divide the dataset into 5 categories. Each category contains sentences with 1, 2, 3, 4, 5, or more than 5 triples. We experimented with the above three models on different types of datasets. As shown in Figs. 6 and 7, our model has achieved better experimental results in most cases of the two datasets. However, when the model processes sentence larger than 5 triples, the performance dropped significantly. Through analysis, we found the following two reasons First of all, as the number of triples in a sentence increases, the overlap problem becomes more complicated. There may be overlaps between each triple. Secondly, sentences with more than 5 triples in the datasets only occupy a small part of the overall dataset. Therefore, the model is difficult to fully learn.

Table 5 Results on NYT-multi dataset

Model	Precision	Recall	F1
CoType	42.3	51.1	46.3
CASREL	39.6	65.6	49.4
Ours	41.7	61.4	49.7

Table 6 Results on relational triple elements

Elements	NYT	WebNLG
E1	94.8	**97.6**
E2	94.9	95.9
R	**95.5**	95.3
(E1, R)	93.0	93.3
(R, E2)	92.8	93.8
(E1, E2)	91.3	93.4

Bold highlights the superiority of the indicator

In addition to reporting on the "silver" quality test data, we also add a manual gold standard evaluation to evaluate the true quality of the model. The same NYT dataset as Ren et al. [7] were used to test the model and name it NYT-multi. The training corpus consists of 1.18 million sentences, which are taken from about 294k 1987–2007 "New York Times" news articles. The author of [46] manually annotated 395 sentences to build test data. After processing, the NYT-multi dataset contains 56,336 training sets, 5000 validation sets and 395 test sets. The experimental results are shown in the Table 5.

4.3.4 Error analysis

In order to explore the factors that affect the triples extraction by the model, we show the predicted F1 scores of different elements in the triples in Table 6. E1 represents the head entity, E2 represents the tail entity, and R represents the relation. In the experiment, if the start and end positions of the entity are correctly predicted, the entity is considered correct. When E1 and E2 are correct, (E1, E2) is correct. (E1, R) and (R, E2) respectively represent the prediction of the combination of the head entity and the tail entity and the relation. Only when the two entities and the relation corresponding to a triplet are correct, the triplet is considered to be correct.

It can be seen from Table 5 that for NYT, the recognition precision of the head entity and the tail entity recognition is higher, but when two entities are recognized as matching, the F1 score drops. The recognition results on E1 and E2 are consistent with the results on (E1, R) and (R, E2), which proves the effectiveness of our proposed model in recognizing head and tail entities. In addition, there is only a small gap between the F1 score of (E1, E2) and (E1, R, E2), but the gap between (E1, R, E2) and (E1, R) and (R, E2) is large. It means that relation extraction is easier than recognizing the entities in the triples. Compared with NYT, the F1 score of the head and tail entities in the WebNLG dataset are higher. This is mainly because the head and tail entities in the WebNLG dataset are mostly composed of a vocabulary, and the entity boundaries are easier to determine. The F1 score of its R is lower than that of NYT, mainly because the number of relations contained in the two datasets is different.

5 Conclusions

In this paper, we propose an end-to-end sequence labeling framework based on the joint extraction of entities and relations based on relation decomposition. Experimental results show that our model can extract multiple triples from sentences at the same time, and effectively solve the problem of overlapping triples. We conducted a large number of experiments on two publicly used datasets to verify the effectiveness of the proposed model. The experimental results show that our model is definitely better than the latest baseline in different situations, especially when extracting overlapping relational triples. In future work, we hope to propose a better extraction scheme for a larger number of relations and triples in sentences. In addition, we hope to make progress in other information extraction tasks, such as event extraction.

Acknowledgements This work is supported by the National Natural Science Foundation of China under Grant No. 61862063, 61502413, 61262025, 62002310. the National Social Science Foundation of China under Grant No. 18BJL104; the Natural Science Foundation of Key Laboratory of Software Engineering of Yunnan Province under Grant No. 2020SE301. Yunnan Science and Technology Major Project under Grant No. 202002AE090010, 202002AD080002-5. the Data Driven Software Engineering Innovative Research Team Funding of Yunnan Province under Grant No. 2017HC012.

References

1. Zelenko D, Aone C, Richardella A (2003) Kernel methods for relation extraction. J Mach Learn Rese 3:1083–1106
2. Chan YS, Roth D (2011) Exploiting syntactico-semantic structures for relation extraction. In: Proceedings of the 49th annual meeting of the association for computational linguistics, pp. 551–560
3. Nadeau D, Sekine S (2007) A survey of named entity recognition and classification. Lingvist Investig 30:3–26
4. Bach N, Badaskar S (2007) A review of relation extraction. Lit Rev Lang Stat 2:1–15
5. Li Q, Ji H (2014) Incremental joint extraction of entity mentions and relations. In: Proceedings of the 52nd annual meeting of the association for computational linguistics, pp. 402–312
6. Miwa M, Sasaki Y (2014) Modeling joint entity and relation extraction with table representation. In: Proceedings of the 2014 conference on empirical methods in natural language processing, pp. 1858–1869
7. Ren X, Wu Z, He W, Qu M, Voss CR, Ji H, (2017) Cotype: joint extraction of typed entities and relations with knowledge bases. In: Proceedings of the 26th international conference on world wide web, pp. 1015–1024
8. Yu X, Lam W (2010) Jointly identifying entities and extracting relations in encyclopedia text via a graphical model approach. In: Coling 2010: posters, pp. 1399–1407
9. Huang Z, Xu W, Yu K (2015) Bidirectional lstm-crf models for sequence tagging. arXiv preprint arXiv:1058.01991
10. Zheng S, Wang F, Bao H, Yue XH, Peng Z, Bo X (2017) Joint extraction of entities and relations based on a novel tagging scheme. arXiv preprint arXiv:1706.05075
11. Miwa M, Bansal M (2016) End-to-end relation extraction using lstms on sequences and tree structures. In: Proceedings of the fifty-fourth annual meeting of the association for computational linguistics, p. 1105-1116
12. Li F, Zhang M, Fu G, Ji D (2017) A neural joint model for entity and relation extraction from biomedical text. Physica A 18:1–11
13. Feng Y, Zhang H, Hao W, Chen G (2017) Joint extraction of entities and relations using reinforcement learning and deep learning. Comput Intell Neurosci. https://doi.org/10.1155/2017/7643065
14. Katiyar A, Cardie C (2017) 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, pp. 917–928
15. Zeng XR, Zeng DJ, He SZ, Liu K, Jun Z (2018) 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, pp. 506–514
16. Zhong Z, Chen D (2021) A frustratingly easy approach for entity and relation extraction. In: Proceedings of the 2021 conference of the North American chapter of the Association for Computational Linguistics: human language technologies
17. Lafferty J, McCallum A, Pereira FCN (2001) Conditional random fields: probabilistic models for segmenting and labeling sequence data. In: ICML '01: Proceedings of the eighteenth international conference on machine learning. ACM, pp 282–289
18. Lample G, Ballesteros M, Subramanian S (2016) Neural architectures for named entity recognition. In: Proceedings of the 2016 conference of the North American chapter of the Association for Computational Linguistics: human language technologies
19. Xu K, Feng Y, Huang S, Zhao D (2015a) Semantic relation classification via convolutional neural networks with simple negative sampling. In: Proceedings of the 2015 conference on empirical methods in natural language processing, p. 536–540
20. Zeng D, Liu K, Lai S, Zhou G, Zhao J (2014) Relation classification via convolutional deep neural network. In: Proceedings of the twenty-fifth COLING international conference, p. 2335-2344
21. Xu Y, Mou L, Li G, Chen Y, Peng H, Jin Z (2015b) Classifying relations via long short term memory networks along shortest dependency paths. In: Proceedings of the 2015 conference on empirical methods in natural language processing, p. 1785–1794
22. Zhou P, Shi W, Tian J, Qi Z, Li B, Hao H (2016) Attention-based bidirectional long short-term memory networks for relation classification. In: Proceedings of the 54th annual meeting of the association for computational linguistics, pp. 207–212

23. Zhang T, Lin H, Tadesse M, Ren Y, Duan X, Xu B (2020) Chinese medical relation extraction based on multi-hop self-attention mechanism. Int J Mach Learn Cybern 12:355–363
24. Zhou Y, Huang L, Guo T, Hu S, Han J (2019) An attention-based model for joint extraction of entities and relations with implicit entity features. In: Proceedings of the 2019 world wide web conference, p. 729-737
25. Meng Z, Tian S, Yu L, Lv Y (2020) Joint extraction of entities and relations based on character graph convolutional network and multi-head self-attention mechanism. J Exp Theor Artif Intell 33(2):349–362
26. Luo L, Yang Z, Cao M, Wang L, Zhang Y (2020) A neural network-based joint learning approach for biomedical entity and relation extraction from biomedical literature. J Biomed Inform 103:103384
27. Bekoulis G, Deleu J, Demeester T (2018) Joint entity recognition and relation extraction as a multi-head selection problem. Expert Syst Appl 114:34–45
28. Huang W, Cheng X, Wang T, Chu W (2019) Bert-based multi-head selection for joint entity-relation extraction. In: CCF international conference on natural language processing and chinese computing, pp. 713–723
29. Zeng X, He S, Zeng D (2019) Learning the extraction order of multiple relational facts in a sentence with reinforcement learning. In: Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing, pp. 367–377
30. Dai D, Xiao X, Lyu Y, Dou S, She Q, Wang H (2019) Joint extraction of entities and overlapping relations using position-attentive sequence labeling. In: Proceedings of the AAAI conference on artificial intelligence, pp. 6300–6308
31. Zeng D, Zhang H, Liu Q (2020) Copymtl: Copy mechanism for joint extraction of entities and relations with multi-task learning. In: Proceedings of the AAAI conference on artificial intelligence, pp. 9507–9514
32. Yu B, Zhang Z, Shu X (2020) Joint extraction of entities and relations based on a novel decomposition strategy. In: Proceedings of ECAI
33. Liu J, Chen S, Wang B, Zhang J, Li N, Xu T (2020) Attention as relation: Learning supervised multi-head self-attention for relation extraction. In: Proceedings of the twenty-ninth international joint conference on artificial intelligence, pp. 1294–1298
34. Wei Z, Su J, Wang Y, Tian Y, Chang Y (2020) A novel cascade binary tagging framework for relational triple extraction. In: Proceedings of the 58th annual meeting of the association for computational linguistics, pp. 1476–1488
35. Ye H, Zhang N, Deng S, Chen M, Tan C, Huang F, Chen H (2021) Contrastive triple extraction with generative transformer. In: Proceedings of the AAAI conference on artificial intelligence
36. Su JL (2019) Conditional layer normalization-based conditional text generation. https://spaces.ac.cn/archives/7124. Accessed 30 Sept 2019
37. Devlin J, Chang MW, Lee K, Toutanova K (2018) Bert: pre-training of deep bidirectional transformers for language understanding, In: Proceedings of the 58th annual meeting of the association for computationa. l Linguistics. p. 1810.04805
38. Vaswani A, Shazeer N, Parmar N (2017) Attention is all you need. Adv Neural Inf Process Syst 30:5998–6008
39. Lin M, Chen Q, Yan S (2013) Network in network. In: Processing of the 2th international conference on learning representations
40. Lin Z, Feng M, Santos CN (2017) A structured self-attentive sentence embedding. In: Processing of the 5th international conference on learning representations
41. Riedel S, Yao L, McCallum A (2010) Modeling relations and their mentions without labeled text. In: joint European conference on machine learning and knowledge discovery in databases, pp. 148–163
42. Takanobu R, Zhang T, Liu J, Huang M (2019) A hierarchical framework for relation extraction with reinforcement learning. In: Proceedings of the AAAI conference on artificial intelligence, p. 7072–7079
43. Fu TJ, Li PH, Ma WY (2019) 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, p. 1409-1418
44. Bai C, Pan L, Luo S, Wu Z (2020) Joint extraction of entities and relations by a novel end-to-end model with a double-pointer module. Neurocomputing 377:325–333
45. Nayak T, Ng HT (2020) Effective modeling of encoder-decoder architecture for joint entity and relation extraction. In: Proceedings of the AAAI conference on artificial intelligence, p. 1911.09886
46. Hoffmann R, Zhang C, Ling X, Zettlemoyer L, Weld DS (2011) Knowledge-based weak supervision for information extraction of overlapping relations. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, pp. 541–550