



REVIEW

# A Review of Joint Extraction Techniques for Relational Triples Based on NYT and WebNLG Datasets

Chenglong Mi<sup>1</sup>, Huaibin Qin<sup>1,\*</sup>, Quan Qi<sup>1</sup> and Pengxiang Zuo<sup>2</sup>

<sup>1</sup>School of Information Science and Technology, Shihezi University, Shihezi, 832000, China

<sup>2</sup>School of Medicine, Shihezi University, Shihezi, 832000, China

\* Corresponding Author: Huaibin Qin. Email: qhb\_inf@shzu.edu.cn

Received: 08 October 2024; Accepted: 11 December 2024; Published: 06 March 2025

**ABSTRACT:** In recent years, with the rapid development of deep learning technology, relational triplet extraction techniques have also achieved groundbreaking progress. Traditional pipeline models have certain limitations due to error propagation. To overcome the limitations of traditional pipeline models, recent research has focused on jointly modeling the two key subtasks-named entity recognition and relation extraction-within a unified framework. To support future research, this paper provides a comprehensive review of recently published studies in the field of relational triplet extraction. The review examines commonly used public datasets for relational triplet extraction techniques and systematically reviews current mainstream joint extraction methods, including joint decoding methods and parameter sharing methods, with joint decoding methods further divided into table filling, tagging, and sequence-to-sequence approaches. In addition, this paper also conducts small-scale replication experiments on models that have performed well in recent years for each method to verify the reproducibility of the code and to compare the performance of different models under uniform conditions. Each method has its own advantages in terms of model design, task handling, and application scenarios, but also faces challenges such as processing complex sentence structures, cross-sentence relation extraction, and adaptability in low-resource environments. Finally, this paper systematically summarizes each method and discusses the future development prospects of joint extraction of relational triples.

**KEYWORDS:** Relation triplet extraction; joint extraction methods; named entity recognition; relation extraction

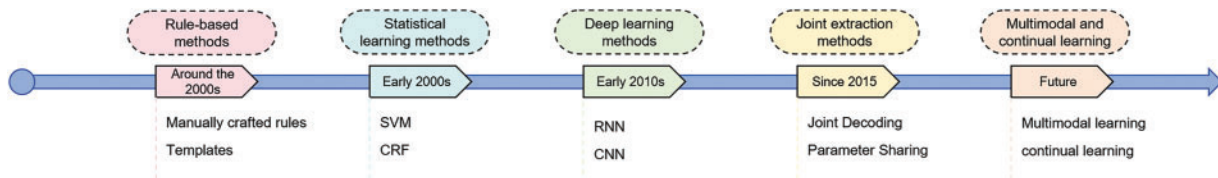
## 1 Introduction

The relationship triplet is regarded as a crucial component in knowledge graphs, typically represented in the structure (h, r, t), where h denotes the head entity, t signifies the tail entity, and r describes the relationship between the two entities. For instance, the triplet (Paris, capital\_of, France) conveys that “Paris is the capital of France.” To construct knowledge graphs effectively, the extraction of relationship triplets is identified as a vital step, which involves the task of extracting entities and their interrelations from unstructured natural language text. This extraction process is generally divided into two key steps. The first step is named entity recognition [1,2], which involves the identification of all specific entities within the text, including names of people, locations, and organizations. The accuracy of this step is critical, as it relies on the ability to extract entities from diverse and complex text data, providing the foundational elements necessary for subsequent relationship extraction. Following this, the second step is relationship extraction [3,4], wherein the relationships existing between the identified entities are analyzed and categorized. The challenge inherent in this phase is the accurate identification of the semantics of relationships, particularly when relationships



are implied or expressed in various forms. Consequently, by integrating the recognized entities with their corresponding relationships, a complete triplet is constructed, which is further utilized for the expansion and refinement of the knowledge graph.

This comprehensive process not only enhances the quality of structured information within knowledge graphs but also establishes a solid foundation for automated information inference and knowledge discovery. Overall, relationship triplet extraction has consistently been a prominent research topic in the field of natural language processing. Fig. 1 illustrates the timeline of the development of the relation triplet extraction task. In earlier studies, pipeline models [5] were predominantly employed to accomplish the triplet extraction task. Within this model framework, named entity recognition and relationship extraction were systematically decomposed into two independent subtasks. Initially, in the named entity recognition phase, potential entity pairs were identified from the text; subsequently, in the relationship extraction phase, attempts were made to determine and classify the relationships between the recognized entity pairs. Although this sequential processing approach simplifies the complexity of the task, it simultaneously introduces several challenges and limitations, such as the following:



**Figure 1:** Timeline of the relation triplet extraction task

1. **Error Propagation:** The main flaw of the pipeline model lies in the information loss resulting from the independence of the subtasks. The lack of effective context and interaction information transfer between named entity recognition and relationship extraction can lead to the accumulation of errors. For instance, if named entity recognition incorrectly labels an entity, this error will prevent the relationship extraction phase from accurately identifying or classifying the relationship between that entity and others. As the tasks are processed sequentially, errors gradually amplify, negatively impacting the effectiveness of the final relationship triplet extraction.

2. **Lack of Interaction:** Because the two subtasks work independently, the model cannot share information or leverage the potential synergies between entity recognition and relationship classification. For instance, the existence of certain relationships could provide valuable clues for identifying specific entity types, but this information cannot be effectively utilized due to the independent nature of the tasks.

3. **Information Loss:** In each independent subtask, some useful contextual information might be ignored or lost. This loss of information across tasks can negatively impact the overall extraction performance.

4. **Independent Optimization:** The pipeline model usually optimizes each subtask independently, meaning that each step's model is trained and optimized separately without considering the global optimal solution. This independent optimization may result in a decrease in the model's overall performance.

5. **Limited Ability to Handle Complex Relationships:** Since each step can only handle local information, it is difficult to capture the global syntactic and semantic structure. As a result, entities and relationships in complex sentences may not be accurately extracted or parsed.

**6. High Time and Computational Costs:** Each independent subtask requires separate model training and execution, leading to higher time and computational costs for the pipeline model, making it relatively less efficient overall.

In recent years, with a deepening understanding of the limitations of the pipeline model [6], joint modeling of entity recognition and relationship extraction has gradually become a focal point of research. This strategy of joint modeling aims to achieve the extraction of relationship triplets through a single model framework, thereby enhancing overall processing efficiency and accuracy while reducing errors in information transfer. The joint extraction model emerges as a transformative approach, markedly enhancing extraction accuracy and efficiency through the simultaneous recognition of entities and extraction of relationships. This method, while adeptly minimizing computational resource consumption, facilitates a more nuanced understanding of contextual information. By fostering synergistic effects between tasks, it enables the model to exhibit greater flexibility and precision throughout the information extraction process.

In particular, the ability to share information across multiple tasks plays a crucial role in reducing error rates during extraction [7]. Furthermore, the incorporation of rich feature representations not only amplifies the model's generalization capabilities but also allows it to adapt seamlessly to diverse data scenarios. Thus, the joint extraction strategy not only streamlines the data processing workflow but also presents a compelling solution for applications in natural language processing and knowledge graph construction [8]. Ultimately, this approach underscores the profound insights and practical significance inherent in cutting-edge research, paving the way for future advancements in the field.

In summary, the contributions of this review can be articulated as follows:

1. The paper systematically categorizes joint extraction models for relationship triplets into two distinct types based on their architectural frameworks: joint decoding methods and parameter sharing methods. Additionally, it provides a succinct overview of commonly utilized datasets and evaluation criteria relevant to the task of relationship triplet extraction, thereby establishing a foundational understanding for researchers.

2. A thorough analysis and comparative study of relationship triplet extraction models that employ joint decoding methods is presented, which are further classified into three categories: table filling methods, tagging methods, and sequence-to-sequence methods. This classification not only highlights the diversity of approaches but also elucidates their respective strengths and applications.

3. Furthermore, the paper delves into a comparative analysis of models utilizing parameter sharing methods, engaging in a comprehensive discussion of related work. This exploration serves to illuminate the nuances and advancements in this area.

4. Finally, a brief yet insightful overview of the strengths and weaknesses of existing joint extraction models for relationship triplets is provided, alongside an exploration of their application domains. This culminates in a forward-looking perspective on future developments in the field, aiming to inspire further research and innovation.

## 2 Task Description

Due to the possibility of relationship triplets sharing one or two entities, this overlap phenomenon complicates the extraction task. Based on the characteristics of entity overlap [9], sentences can be categorized into three types: (i) **No Entity Overlap (NEO)**: This type of sentence contains one or more triplets, but there are no shared entities among them. (ii) **Entity Pair Overlap (EPO)**: Sentences of this kind feature multiple triplets, where at least two triplets share the same entities, which may be in the same order or in reverse order. (iii) **Single Entity Overlap (SEO)**: In this case, a sentence contains multiple triplets, with at least two triplets

sharing one common entity. The primary objective of relationship triplet extraction is to identify all existing relationship triplets present in the sentence.

Traditional relationship triplet extraction techniques typically divide this task into two independent phases: first, entity recognition is performed, followed by the analysis of the relationships between these entities based on contextual relationships. However, employing a joint modeling approach allows for the simultaneous execution of entity recognition and relationship extraction, thereby forming an integrated model that simplifies this process. This joint modeling strategy enhances the synergy between entity recognition and relationship extraction by sharing contextual information. Consequently, this information-sharing mechanism significantly improves the accuracy and overall efficiency of the extraction process.

### 3 Datasets

There are many datasets available for relational triplet extraction. For instance, WikiEvents offers a wealth of event descriptions along with their associated entities and relationships, making it particularly suitable for understanding event-driven relations. Meanwhile, TACRED serves as a comprehensive benchmark dataset that encompasses various types of texts and relationships, thus becoming a crucial reference for evaluating model performance. Additionally, the SemEval [10] series of competitions presents multilingual datasets that broaden the application scenarios for relation extraction. Lastly, the ACE [11] dataset is dedicated to extracting events and relationships from news articles and other texts, thereby enriching the research materials available in this domain. However, this paper will focus on detailing the two most widely used datasets in recent years: NYT and WebNLG.

#### 3.1 NYT

The NYT [12] dataset is a textual resource primarily sourced from news articles published in The New York Times, encompassing a rich collection of natural language text. This dataset contains information from various domains, including text content, entities, and relationships, making it widely applicable in natural language processing, particularly in the extraction of relationship triplets.

The dataset includes examples with multiple relationships and overlapping entities, annotated with information from the Freebase knowledge graph to ensure accuracy and high quality of the annotations. The NYT dataset comprises a total of 66,194 sentences, covering 24 types of relationship categories. Specifically, 56,195 sentences are allocated to the training set, 4999 sentences are designated as the validation set, while the remaining 5000 sentences are used for testing.

When extracting relationship triplets from the NYT dataset, a variety of challenges regarding annotation quality arises, primarily concerning consistency, noise, and ambiguity. Moreover, contextual dependency and domain specificity significantly influence annotation accuracy, further contributing to the complexity of the extraction process and thereby limiting model performance.

Furthermore, the difficulty intensifies with certain relationship types, such as implicit relationships, multiple relationships, temporal and spatial relationships, and sentiment or attitude relationships. These complexities not only demand advanced reasoning capabilities but also require precise judgment within diverse contexts.

#### 3.2 WebNLG

The WebNLG [13] dataset was originally developed by INRIA (the French National Institute for Research in Computer Science and Automation) and contains a variety of triplets along with their corresponding human-written natural language descriptions. It is primarily used to study how to generate fluent natural

language descriptions from knowledge graphs. The data in the WebNLG dataset is mainly sourced from DBpedia, a knowledge base that extracts structured information from Wikipedia.

The WebNLG dataset comprises a total of 6222 sentences and 246 relationship categories. Each instance in this dataset consists of a set of triplets and several human-written reference sentences, with each reference sentence containing all the triplets for that instance. For detailed statistics on these two datasets, please refer to [Table 1](#).

**Table 1:** Statistics of the datasets.  $N$  is the number of triples in a sentence

Dataset	NYT	WebNLG
Training set	56195	5019
Test set	5000	500
Dev set	4999	703
Relations	24	246
NEO	3222	239
EPO	969	6
SPO	1273	448
$N = 1$	3240	256
$N = 2$	1047	175
$N = 3$	314	138
$N = 4$	290	93
$N > 5$	109	41
Triples	8120	1607

In the WebNLG dataset, the challenges surrounding annotation quality primarily manifest in the diversity and consistency between generated natural language descriptions and the triplets. While the overarching goal is to transform structured data into fluent text, the reality is that a single triplet can be expressed in multiple valid linguistic forms, leading to significant complexity in evaluating the generated text. Moreover, annotations frequently lack essential contextual information, resulting in generated texts that may be semantically inaccurate or incomplete. Furthermore, the generated texts are not infrequently plagued by grammatical errors or unclear expressions, which, in turn, adversely affects the overall quality and usability of the data.

#### 4 Evaluation Metrics

In the process of relationship triplet extraction, it is crucial to accurately identify the boundaries and types of entities, as well as the relationship between the head entity and the tail entity. Only when these elements are correctly identified can the extracted relationship triplets be considered valid. For this task, several commonly used evaluation metrics [14] are as follows:

**Precision** (Prec.): This refers to the ratio of the number of correctly identified relationship triplets to the total number of triplets identified by the model. The calculation formula is as follows:

$$Precision = \frac{TP}{TP + FP} \quad (1)$$

where  $TP$  (true positive): denotes the identification of a completely correct relational triad;  $FP$  (false positive): denotes that it is not the relational triad but the model determines it to be a relational triad.

**Recall (Rec.):** This measures the proportion of relationship triplets successfully identified by the model among all the relationship triplets that actually exist. The calculation formula is as follows:

$$Recall = \frac{TP}{TP + FN} \quad (2)$$

where FN (false negative): indicates that it should have been recognized but the model did not actually recognize it.

**F1 score (F1):** This metric is the harmonic mean of precision and recall, providing a comprehensive assessment of performance in both aspects. A higher F1 score reflects the overall effectiveness of the model.

$$F_1 = \frac{2Precision \times Recall}{Precision + Recall} \quad (3)$$

**Micro-average:** An overall evaluation of all relational triplets, calculating global *Precision*, *Recall*, and  $F_1$ . This approach treats each instance equally and combines the contributions of all categories into a single set of metrics, giving equal weight to each instance.

$$Precision_{micro} = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FP_i} \quad (4)$$

$$Recall_{micro} = \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FN_i} \quad (5)$$

$$micro - F_1 = 2 \times \frac{Precision_{micro} \times Recall_{micro}}{Precision_{micro} + Recall_{micro}} \quad (6)$$

**Macro-average:** As an evaluation method, this computes the *Precision*, *Recall*, and  $F_1$  for each category separately, and then averages these results. In this process, all categories are treated equally, regardless of the number of samples in each category, ensuring that the performance of smaller categories receives appropriate attention and is not overshadowed by larger categories.

$$Precision_{macro} = \frac{1}{n} \sum_{i=1}^n Precision_i \quad (7)$$

$$Recall_{macro} = \frac{1}{n} \sum_{i=1}^n Recall_i \quad (8)$$

$$macro - F_1 = 2 \times \frac{Precision_{macro} \times Recall_{macro}}{Precision_{macro} + Recall_{macro}} \quad (9)$$

**Top-k Evaluation Metrics:** When a system can output multiple candidate triplets, Top-k evaluation metrics such as Top-1, Top-5, Top-10, etc., are used to assess the model's performance. Specifically, the essence of the Top-k metric lies in its ability to assess the proportion of correct results among the top k outcomes returned by a model. This metric enjoys widespread popularity due to its simplicity and ease of understanding; however, it is crucial to acknowledge its limitations. Notably, the Top-k metric focuses solely on the top k results, which may lead to the oversight of the overall quality of the results. This characteristic becomes particularly pronounced in the context of relationship triplet extraction tasks. Consequently, the Top-k metric is often best employed in conjunction with other evaluation metrics to achieve a more comprehensive assessment of model performance.

By integrating multiple evaluation dimensions, researchers are better positioned to gain a nuanced understanding of how models perform on complex tasks, thereby avoiding the pitfalls of relying on a singular

metric that could yield a skewed perspective. Therefore, adopting a diversified set of evaluation standards in the assessment of relationship triplet extraction performance is instrumental in uncovering both the strengths and weaknesses of models, ultimately fostering advancement within this field.

## 5 Joint Extraction

### 5.1 Joint Decoding

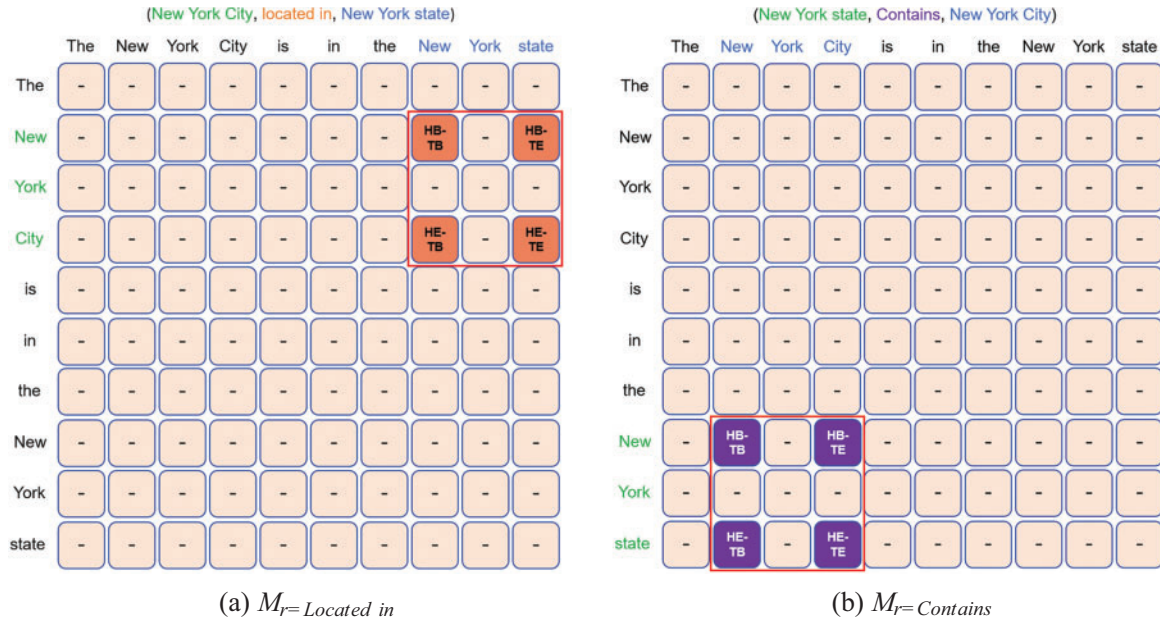
To overcome certain limitations of traditional pipeline models, researchers have proposed a joint decoding strategy. Currently, research on joint decoding in the field of relationship triplet extraction primarily focuses on three directions: (i) **Table Filling**: This method creates a table or matrix where the rows and columns represent different entities in the sentence, and each cell in the matrix indicates the relationship between the two entities. (ii) **Tagging**: This method treats relationship triplet extraction as a sequence labeling problem, introducing relationship information into the labeling system. (iii) **Sequence-to-Sequence**: This method takes unstructured text as input and directly generates relationship triplets, presenting them in a sequential output format.

#### 5.1.1 Table Filling

The table filling approach typically maintains a separate table for each relationship, and the triplets are extracted based on the populated relationship tables. TPLinker [15] and UniRE [16] are among the most effective table filling models. TPLinker treats the joint extraction problem as a token-pair linking task, constructs a single-step model that avoids interdependent steps, and introduces a handshake tagging scheme that efficiently and accurately extracts relational triplets through a linking mechanism, addressing exposure bias and error accumulation issues. However, the model has high tagging complexity, resulting in redundant operations and information. UniRE, on the other hand, proposes an innovative unified label space approach, treating relational triplet extraction as a table filling problem. It uses a unified classifier to predict the labels for each cell in the table, simplifying task learning and enhancing the interdependence between tasks. Hu et al. [17] introduce a hybrid deep relational matrix bidirectional method, combining relationship matrix representation and bidirectional processing to achieve joint extraction of relational triplets. GRTE [18] enhances model performance by incorporating a global feature-oriented strategy and an improved matrix-filling strategy, effectively capturing complex dependencies in sentences. Yan et al. [19] propose an innovative partition filtering mechanism that effectively captures fine-grained information and enhances overall task performance through joint learning. Liu et al. [20] view relationships as attention distributions, using attention weights to represent relationships between entities, enabling the model to simultaneously identify multiple entities and capture the relationships between them. Fig. 2 shows an example of a table filling model.

OneRel [21] introduced the Rel-Spec Horns tagging strategy, reducing the required number of label matrices from  $2N + 1$  to  $N$ , with each relationship having a single matrix. This reduction in redundancy allowed for the efficient extraction of relational triplets using a single module. Meanwhile, Wang et al. [22] combined the strengths of table filling and sequence encoding in an end-to-end approach for joint extraction. Similarly, Ning et al. [23] borrowed the object detection concept from computer vision, treating entities and relationships in text as “objects” to be detected through a single-stage framework, simultaneously identifying and extracting entities and their relationships. Additionally, Zhang et al. [24] used relationship prompts as part of the input to guide the model in accurately recognizing and extracting entities and relationships within the text. Wang et al. [25] further improved joint extraction by locating potential entity spans and using a relational graph to capture relationships between these entities. To better handle complex sentence structures and overlapping relationships, Han et al. [26] adopted span representations, while Tian et al. [27] enhanced the understanding of entities and relationships through a multi-view information

fusion strategy. Finally, Tang et al. [28] proposed the UniRel model, which utilizes the Transformer’s self-attention mechanism to model interactions between entity-entity and entity-relationship pairs within a single interaction map. In contrast, Wang et al. [29] employed tensor decomposition techniques to effectively model complex dependencies between entities and relationships, demonstrating excellent performance in processing intricate texts. Table 2 show the current state-of-the-art in table filling models.



**Figure 2:** Example of a table filling method

**Table 2:** Current state-of-the-art in table filling models

Model	NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1
TPLinker <sub>LSTM</sub> [15]	86.0	82.0	84.0	91.9	81.6	86.4
TPLinker <sub>BERT</sub> [15]	91.4	92.6	92.0	88.9	84.5	86.7
OneRel [21]	93.2	92.6	92.9	91.8	90.3	91.0
GRET <sub>LSTM</sub> [18]	86.2	87.1	86.6	88.0	86.3	87.1
GRET <sub>BERT</sub> [18]	93.4	93.5	93.4	92.3	87.9	90.0
PFN [19]	–	–	92.4	–	–	93.6
Document [20]	88.1	78.5	83.0	89.5	86.0	87.7
Span-RG [22]	83.4	68.1	74.9	66.5	62.8	64.6
DMBRE [17]	92.1	93.5	92.8	89.0	88.8	88.9
OD-RTE [23]	94.2	93.6	93.9	92.8	92.1	92.5
RPSS [24]	93.5	93.2	93.3	94.7	95.1	94.9
SMHS [26]	97.5	82.0	89.1	–	–	–
StereoRel [27]	92.0	92.3	92.2	91.6	92.6	92.1
UniRel [28]	93.5	94.0	93.7	94.8	94.6	94.7
TLRel [29]	88.5	85.2	86.8	91.8	92.7	92.2



The table filling approach necessitates that the model enumerate all potential entity pairs. While this process demonstrates commendable performance in managing local entities and relationships, it encounters significant challenges when the distance between entities and relationships increases. In such cases, the model's ability to learn the corresponding positions within the matrix becomes compromised. Moreover, as longer sentences are processed, the size of the matrix expands rapidly, leading to a marked increase in the demand for computational resources, including memory and storage. Consequently, this computational complexity becomes particularly pronounced when handling long texts, often resulting in diminished model efficiency. It is essential to underscore that in the pursuit of both accuracy and comprehensiveness, optimizing the use of computational resources remains a critical challenge for this method. This consideration not only highlights the intricacies involved in algorithm design but also points the way toward future research endeavors aimed at developing more efficient model architectures and methodologies.

### 5.1.2 Tagging

The tagging approach processes input text as a linear sequence by assigning a corresponding label to each word and gradually extracting the required triples by processing each element in the sequence. PRGC [30] is a more advanced tagging model that breaks down the joint extraction of triples into three subtasks: relation judgment, entity extraction, and head-tail entity alignment. It enhances overall extraction accuracy by predicting potential relationships in the text through a latent relationship module combined with a global alignment mechanism. However, it still suffers from issues such as error propagation and exposure bias. Additionally, Cheng et al. [31] propose an innovative cascaded double-decoder architecture, which effectively reduces the negative impact of entity recognition errors on the relationship classification task, thereby better capturing the dependencies between entities and relationships. Moreover, Qiao et al. [32] leverage the powerful representation capabilities of BERT to construct a model that jointly learns entity recognition and relation extraction tasks. Furthermore, CasRel [33] adopts a cascaded binary labeling strategy, breaking down the complex relation extraction task into simpler subtasks by first extracting the subject entity, and then simultaneously extracting the relation and its corresponding object entity, thereby reducing the model's learning difficulty. Lastly, Yuan et al. [34] assign different attention weights to each type of relationship, allowing the model to accurately capture entities related to specific relationships. Fig. 3 shows an example of a Tagging model.

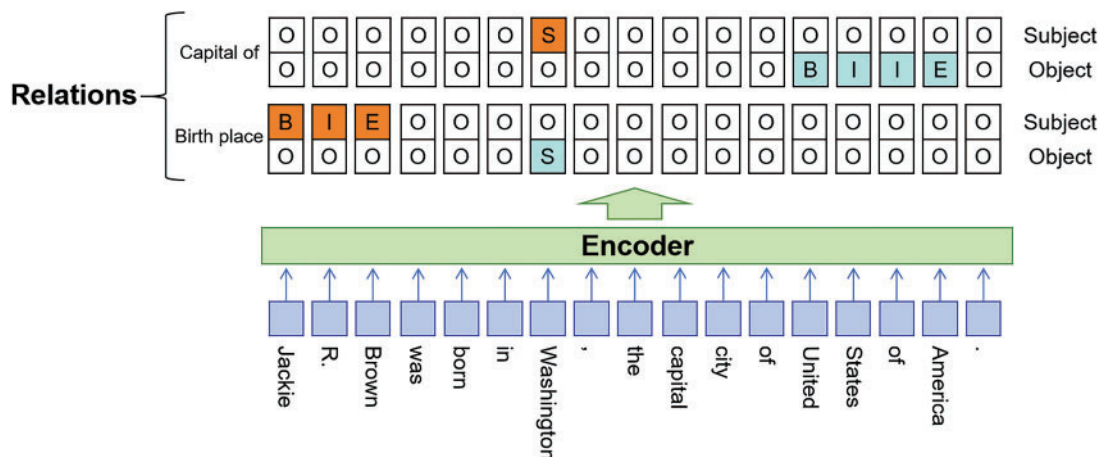


Figure 3: Example of a tagging method

Ren et al. [35] proposed a bidirectional extraction framework that enhances the overall performance and accuracy of the model by processing the text from left to right and right to left. Similarly, Xu et al. [36] innovatively employed a joint entity labeling approach, significantly improving the effectiveness of information extraction. Furthermore, Chen et al. [37] enhanced the model's ability to identify entities and relationships by introducing a position-aware attention mechanism and relationship embedding techniques. Hang et al. [38] identified entities in the text using a multi-label annotation method and employed a relationship alignment mechanism to clarify the relationships between these entities. Meanwhile, Zhang et al. [39] utilized the characteristics of capsule networks to capture information in both forward and backward directions and processed complex entity and relationship extraction tasks through a cascading structure. Finally, Wu et al. [40] achieved synchronous extraction of entities and relationships while employing a cross-type attention mechanism to capture the interdependencies between entities and relationships. Table 3 shows the current state-of-the-art in Tagging models.

The tagging approach serves as a straightforward and efficient method for extracting relational triplets, which renders it particularly well-suited for handling short texts and tasks characterized by relatively simple structures. However, when confronted with overlapping entities, nested entities, multiple relationships, or long-distance dependencies, this method may encounter significant challenges. Given that it cannot assign multiple distinct labels to the same entity, there arises a need to design more complex labeling structures or to introduce more sophisticated model architectures to effectively address these issues. This reality not only highlights the limitations of the tagging method in specific contexts but also underscores the necessity of exploring more flexible strategies and solutions in the processing of more complex texts.

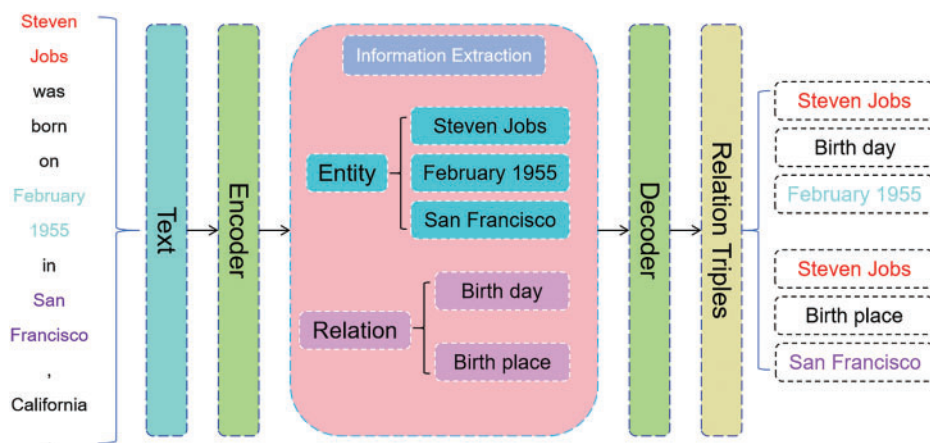
**Table 3:** Current state-of-the-art in Tagging models

Model	NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1
PRGC [30]	93.5	91.9	92.7	89.9	87.2	88.5
Document [31]	89.9	91.4	90.6	88.0	88.9	88.4
Document [32]	61.0	51.3	55.7	–	–	–
CASREL [33]	89.7	89.5	89.6	93.4	90.1	91.8
RSAN [34]	85.7	83.6	84.6	80.5	83.8	82.1
BiRTE [35]	91.9	93.7	92.8	89.0	89.5	89.3
GraphJoint [36]	88.7	83.8	86.2	88.3	87.7	87.9
PARE-Joint [37]	92.9	91.4	92.1	93.4	90.8	92.1
MLRA-LSTM-CRF [38]	74.9	45.3	56.5	–	–	–
CBCapsule [39]	88.4	87.2	87.8	90.9	89.2	90.0
SDN [40]	94.2	91.5	92.8	92.7	89.6	91.1

### 5.1.3 Sequence-to-Sequence

The basic concept of the sequence-to-sequence method is to transform the input sequence into a fixed-length vector representation, after which a decoder is used to generate a new sequence from that representation. Nayak et al. [41] effectively addressed the interdependence between entity recognition and relationship extraction by leveraging the advantages of the encoder-decoder structure, providing a viable joint extraction strategy for both tasks. Zeng et al. [42] proposed an end-to-end neural network model that combines generation and copying mechanisms, significantly improving accuracy when handling

complex entities and relationships. Sui et al. [43] employed a set prediction network to jointly model entity recognition and relationship extraction, enabling the model to simultaneously handle multiple entities and relationships. Zhao et al. [44] designed a joint extraction model based on heterogeneous graph neural networks, which iteratively integrated global information to enhance the precision of the extraction task. Zhang et al. [45] introduced a bias minimization strategy aimed at improving the accuracy of the joint extraction model by reducing exposure bias. Additionally, the BTDM model [46] utilized a bidirectional translation decoding mechanism, allowing the model to better leverage contextual information to ensure the accuracy of entity and relationship extraction. Chang et al. [47] incorporated techniques such as constrained decoding, representation reuse, and fusion to ensure that the generated entity and relationship triples adhere to syntactic and semantic rules. Finally, Ye et al. [48] combined generative Transformers with contrastive learning to enhance the model's discriminative capability, enabling it to handle more complex sentence structures. Fig. 4 shows an example of a sequence-to-sequence model.



**Figure 4:** Example of a sequence-to-sequence method

Wang et al. [49] enhanced the model's capability to recognize entities and relationships by integrating various semantic features, enabling effective handling of texts that contain complex structures and rich semantic information. Chen et al. [50] successfully addressed the joint extraction problem of entities and relationships using an enhanced binary pointer network, while identifying implicit relationships in the text through reasoning patterns. Simultaneously, Tan et al. [51] introduced a query mechanism and instance differentiation method to better distinguish complex contexts and similar instances in relationship extraction tasks, thereby improving the accuracy of entity and relationship extraction. Zhang et al. [52] combined external knowledge resources with contextual information to enhance the accuracy and robustness of entity and relationship extraction tasks. Additionally, Shang et al. [53] simplified the extraction process by simultaneously executing all extraction operations within a unified model. Li et al. [54] emphasized the approach of first identifying relationships within the text and then inferring and filling in the corresponding entities to complete the extraction of triples. Lai et al. [55] proposed a neural network model that incorporates a multi-head attention mechanism, effectively capturing the dependencies between entities and relationships in the text by introducing relationship-aware multi-head attention mechanisms. Furthermore, Liang et al. [56] employed a standard sequence-to-sequence architecture, treating the input text as a sequence and generating an output sequence that represents entity and relationship triples. Finally, Li et al. [57] proposed a decoding scheme called TDEER, which views relationships as transformations from the subject to the object,

effectively addressing the issue of overlapping relationship triples. Table 4 show the current state-of-the-art in sequence-to-sequence models.

**Table 4:** Current state-of-the-art in sequence-to-sequence models

Model	NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1
PNDec [41]	89.3	78.8	83.8	–	–	–
CopyRE [42]	61.0	56.6	58.7	37.7	36.4	37.1
SPN [43]	92.5	92.2	92.3	93.1	93.6	93.4
RIFRE [44]	93.6	90.5	92.0	93.3	92.0	92.6
Seq2UMTree [45]	79.1	75.1	77.1	–	–	–
BTDM [46]	93.1	92.4	92.7	90.9	90.1	90.5
Document [47]	92.8	93.1	92.9	90.4	92.4	91.4
Document [49]	93.6	91.7	92.6	94.9	92.3	93.5
CGT [48]	94.7	84.2	89.1	92.9	75.6	83.4
R-BPtrNet [50]	94.0	92.9	93.5	94.3	93.3	93.8
QIDN [51]	93.4	92.6	93.0	94.1	93.7	93.9
REKnow [52]	93.1	94.1	93.6	90.4	87.9	89.1
DirectRel [53]	93.6	92.2	92.9	91.0	89.0	90.0
RFBN [54]	93.7	93.6	93.6	91.5	89.4	90.4
RMAN [55]	87.1	83.8	85.4	83.6	85.3	84.5
Seq2Seq-RE [56]	88.3	77.3	82.5	–	–	–
TDEER [57]	93.0	92.1	92.5	93.8	92.4	93.1

The sequence-to-sequence approach naturally excels at handling nested entities, overlapping entities, and complex multiple relationships, as its decoding process allows for output generation at any position rather than being constrained by a fixed set of labels. However, a notable distinction arises during training, where the decoder typically relies on the ground truth label from the preceding word, while in the inference phase, it must depend on the labels it generates itself. This discrepancy can lead to a decline in model performance during inference.

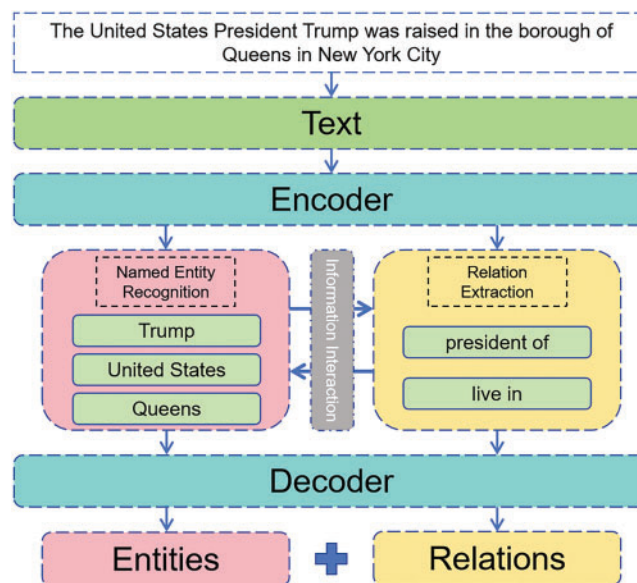
Moreover, the model may still encounter challenges when generating long sequences, particularly when tasked with extracting relationships that span multiple sentences, where accuracy is likely to diminish. Consequently, although the sequence-to-sequence method possesses significant potential in addressing complex structures, the critical difference between training and inference phases, along with the difficulties associated with long sequence generation, highlights the limitations of the model in practical applications.

## 5.2 Parameter Sharing

Unlike the joint decoding approach, the parameter-sharing method adopts a multi-module, multi-step process to extract relational triplets. Essentially, parameter-sharing still divides the relational triplet extraction task into two subtasks: named entity recognition and relation extraction. By sharing the parameters of the encoding layer in a joint model, it enables joint learning, thus facilitating interdependency between entity recognition and relation extraction. Existing parameter-sharing methods mainly rely on joint decoding or

multi-task learning techniques, enhancing the interaction between these two subtasks by sharing underlying parameters, ultimately achieving joint extraction of relational triplets.

Geng et al. [58] enhanced the model's ability to understand complex relationships within texts by integrating various semantic features, resulting in improved performance when handling multiple entities and longer sentences. Similarly, Gao et al. [59] proposed a relation decomposition-based triple extraction model. This model generates sentence-level vector representations by merging the word vectors of the input text, classifying the relationships within the text first, and then identifying the corresponding entities. Building on CopyRE, Zeng et al. [60] employed a multi-task learning strategy to generate entity and relationship labels for extraction, improving the accuracy of the generated results through a copying mechanism. Li et al. [61] introduced a joint extraction model based on a decomposition strategy, enhancing overall extraction performance by sharing information between tasks. Yu et al. [62] simplified the complex entity-relationship extraction task into several sub-tasks and achieved end-to-end joint extraction. Sun et al. [63] utilized a recursive mechanism to enable continuous information exchange, optimizing entity recognition and relationship extraction to improve the accuracy of the extraction process. Zhang et al. [64] proposed a coarse-to-fine extraction framework, initially performing rough extraction followed by detailed refinement. Finally, Wang et al. [65] adopted a decoupling and aggregation strategy, processing entity recognition and relationship extraction tasks separately and then integrating the results in subsequent stages, thereby enhancing the flexibility of the extraction process. Fig. 5 shows an example of a Parameter Sharing model.



**Figure 5:** Example of a parameter sharing method

Yang et al. [66] incorporated relationship information into the attention mechanism, enabling the model to more effectively focus on entities and their contexts related to specific relationships. Meanwhile, Duan et al. [67] developed an adaptive mechanism that flexibly adjusts processing strategies based on the features of entities and relationships within the context, thereby enhancing extraction accuracy. The triple relationship network mechanism proposed by Wang et al. [68] significantly improved the accuracy and efficiency of joint extraction by directly modeling the complex interactions between entities and relationship triples. Furthermore, Wang et al. [69] leveraged BERT's powerful pre-training capabilities in conjunction with a decomposition strategy to comprehensively enhance the performance of entity and relationship

extraction. Yang et al. [70] adopted a bidirectional relationship-guided attention mechanism that integrates semantics and knowledge, enabling the model to better tackle the challenges posed by complex texts. Gao et al. [71] enhanced the model's ability to handle long sentences by combining stage-wise processing with a global entity docking mechanism. In addition, Sun et al. [72] implemented progressive multi-task learning, achieving more precise entity and relationship extraction by gradually learning each sub-task and effectively controlling the flow of information. Huang et al. [73] utilized transformation rules in knowledge graph embeddings to learn relationship representations, proposing a special relationship called NA and dynamically selecting suitable relationships through a bias loss function. Zhuang et al. [74] designed a priority-driven joint extraction model that prioritizes relationship extraction to guide entity recognition and triple generation. Finally, Chen et al. [75] improved the joint extraction capabilities of entities and relationships by transforming and reinforcing the relationships between text and knowledge graphs. Table 5 shows the current state-of-the-art in Parameter Sharing models.

**Table 5:** Current state-of-the-art in Parameter Sharing models

Model	NYT			WebNLG		
	Prec.	Rec.	F1	Prec.	Rec.	F1
RS-Joint [58]	80.3	57.2	66.8	–	–	–
Document [59]	91.5	90.0	90.7	91.4	92.2	91.8
CopyMTL [60]	75.7	68.7	72.0	58.0	54.9	56.4
Document [61]	86.5	73.2	79.3	85.3	83.1	84.2
ETL-Span [62]	85.5	71.7	78.0	84.3	82.0	83.1
RIN [63]	83.9	85.5	84.7	77.3	76.8	77.0
C2FERE [64]	93.7	92.6	93.1	91.5	89.2	90.3
BiDArtER [65]	–	–	92.6	–	–	94.1
RGAM [66]	90.6	92.0	91.3	93.5	91.9	92.6
Document [67]	81.3	76.7	79.4	67.4	65.1	66.3
TRN [68]	93.0	92.3	92.6	93.5	92.7	93.1
Document [69]	87.0	85.1	86.0	–	–	–
BRASK [70]	93.0	91.5	92.2	94.8	92.2	93.5
ERGM [71]	93.3	91.5	92.4	94.2	91.2	92.7
PMEI [72]	88.4	88.9	88.7	80.8	82.8	81.8
TransRel [73]	90.1	93.9	92.0	92.7	94.9	93.8
RFTE [74]	88.9	90.5	89.7	91.3	92.5	91.9
MTG [75]	95.6	93.1	94.3	94.8	95.1	94.9

In joint extraction models, parameter sharing techniques effectively reduce the number of parameters by allowing certain parameters to be shared between the tasks of entity recognition and relationship extraction. This approach enhances training efficiency and improves the model's generalization capability. However, when significant differences exist between these tasks, parameter sharing can have detrimental effects, ultimately weakening overall performance. The adjustment of shared parameters affects all tasks simultaneously, which often leads to difficulties in fine-tuning specific tasks.

Consequently, this limitation not only restricts the flexibility and adaptability of the parameter-sharing model but also raises concerns regarding its performance in diverse task settings. Therefore, exploring how to

balance the benefits of parameter sharing with the unique requirements of individual tasks becomes essential. This inquiry not only offers potential avenues for enhancing the performance of joint extraction models but also lays the groundwork for future research in this area.

## 6 Experiments

In order to validate the reproducibility of current state-of-the-art relational triplet extraction techniques and address the lack of performance comparisons across different models under uniform conditions, this paper will conduct small-scale comparative experiments under consistent hardware settings. These experiments will utilize the same datasets and evaluation metrics to ensure fairness and reliability. Moreover, additional small-scale experiments will be carried out, focusing on the number of relational triplets present in sentences and the types of overlapping triplets. Through these analyses, the paper aims to evaluate and compare the performance of various extraction models across different datasets, providing a deeper understanding of their strengths and limitations in diverse scenarios.

### 6.1 Datasets and Evaluation Metrics

To provide a clear and intuitive comparison of the strengths and weaknesses of various methods in practical tasks, this paper employs the NYT and WebNLG datasets, as introduced in [Section 3](#), for the comparative experiments. In order to further investigate how different approaches handle complex scenarios, sentences containing varying types of overlapping triplets and sentences with different numbers of triplets are used to test all models.

For the evaluation, precision (Prec.), recall (Rec.), and F1 score (F1) are adopted as the primary metrics, allowing for a comprehensive assessment of the models' performance through the combined analysis of these various indicators. This approach ensures a robust and nuanced evaluation of the methods under different conditions.

### 6.2 Implementation Details

To facilitate a systematic comparison of the discussed models under uniform conditions, all training processes in our experiments are conducted on a workstation equipped with an Intel Xeon Gold 6133 CPU, 256 GB memory, an RTX 4090 GPU, and running Ubuntu 20.04. The model parameters used are directly derived from those provided in the respective papers, ensuring consistency with the original experimental settings.

### 6.3 Baselines

To validate the reproducibility of the models and compare the strengths and weaknesses across different types, we have selected a number of advanced models from each category for the comparative experiments. The models chosen include: (1) **Table filling**: OneRel [21], TPLinker [15], and GRTE [18]; (2) **Tagging**: PRGC [30], CASREL [33], and BiRTE [35]; (3) **Sequence-to-sequence**: BTDM [46], R-BPtrNet [50], and RFBFN [54]; (4) **Parameter sharing**: C2FERE [64], ERGM [71], and MTG [75]. The code for all these models is sourced from the publicly available implementations provided in the respective papers. This selection ensures a representative and comprehensive evaluation of the models' performance across various extraction tasks.

## 6.4 Results and Analysis

From Table 6, it can be observed that the best-performing model is MTG, which demonstrates exceptional handling of overlapping relational triplets in sentences, particularly excelling with an F1 score of 98.0% on the EPO-type data in the WebNLG dataset. In addition, the table filling model shows generally strong performance across both datasets; however, its performance on the WebNLG dataset, particularly with SPO-type data, is slightly lower than its performance on the NYT dataset. This discrepancy highlights a limitation in the table filling model's ability to effectively handle the more complex sentence structures and natural language expressions present in WebNLG. The tagging model, while performing at an average level across all models, effectively leverages contextual information, enabling it to exhibit stable performance across various sentence types. Moreover, it demonstrates good generalization ability, consistently performing well on different datasets.

**Table 6:** F1 scores on sentences with different overlapping patterns

Model	NYT			WebNLG		
	NEO	EPO	SPO	NEO	EPO	SPO
OneRel [21]	90.6	95.1	94.8	<b>91.9</b>	95.4	94.7
TPLinker [15]	90.1	94.0	93.4	87.9	95.3	92.5
GRTE [18]	91.1	95.0	94.4	90.6	94.5	<b>96.0</b>
PRGC [30]	91.0	94.5	94.0	90.4	95.9	93.6
CASREL [33]	87.3	92.0	91.4	89.4	94.7	92.2
BiRTE [35]	91.4	94.2	94.7	90.1	94.3	95.9
BTDM [46]	90.8	94.7	94.9	91.0	94.3	93.5
R-BPtrNet [50]	90.4	95.2	94.4	89.5	96.1	93.9
RFBFN [54]	91.2	95.6	95.2	91.0	96.5	94.6
C2FERE [64]	<b>91.9</b>	95.1	94.5	91.8	96.3	94.7
ERGM [71]	90.9	94.1	93.6	90.3	96.0	93.8
MTG [75]	91.1	<b>96.7</b>	<b>95.7</b>	90.0	<b>98.0</b>	94.5

As for the sequence-to-sequence model, its overall performance on EPO-type triplets is relatively stable across both datasets, showcasing sensitivity to entities and relations in complex sentences. This is particularly important, as errors in entity recognition have a much larger impact than errors in relation classification for EPO-type triplets. However, the sequence-to-sequence model's performance on the SPO-type data in the WebNLG dataset lags behind its performance on the NYT dataset, possibly due to suboptimal utilization of contextual information. Finally, the parameter-sharing model shows superior performance on the EPO-type data in the WebNLG dataset compared to the NYT dataset, reflecting the model's strong capabilities under low-resource conditions. This suggests that, in scenarios with limited resources, the parameter-sharing model can still perform remarkably well.

Table 7 presents the extraction performance of different models when handling sentences with varying numbers of triplets. When  $N < 3$ , C2FERE demonstrates the best performance, indicating its strong ability to handle simpler sentence structures. In contrast, when  $N > 3$ , MTG outperforms all other models, highlighting its superior understanding of more complex sentence structures. Overall, the parameter-sharing model excels in handling sentence structures with varying numbers of triplets, which can be attributed to its



unique construction approach. By employing a multi-task learning framework that combines named entity recognition and relation extraction, the model enhances extraction performance.

**Table 7:** F1 scores on sentences with different triple numbers

Model	NYT					WebNLG				
	N = 1	N = 2	N = 3	N = 4	N > 5	N = 1	N = 2	N = 3	N = 4	N > 5
OneRel [21]	90.5	93.4	93.9	96.5	<b>94.2</b>	91.4	<b>93.0</b>	95.9	95.7	94.5
TPLinker [15]	90.0	92.8	93.1	96.1	90.0	88.0	90.1	94.6	93.3	91.6
GRTE [18]	90.8	93.7	94.4	96.2	93.4	90.6	92.5	96.5	95.5	94.4
PRGC [30]	91.1	93.0	93.5	95.5	93.0	89.9	91.6	95.0	94.8	92.8
CASREL [33]	88.2	90.3	91.9	94.2	83.7	89.3	90.8	94.2	92.4	90.9
BiRTE [35]	91.5	93.7	93.9	95.8	92.1	90.2	92.9	95.7	94.6	92.0
BTDM [46]	90.7	93.4	94.2	96.2	94.0	90.8	92.5	96.1	95.4	92.7
R-BPtrNet [50]	89.5	93.1	93.5	96.7	91.3	88.5	91.4	96.2	94.9	94.2
RFBN [54]	91.4	93.8	<b>94.8</b>	96.4	93.9	90.8	92.6	<b>96.6</b>	94.7	94.5
C2FERE [64]	<b>91.7</b>	<b>94.2</b>	94.2	96.1	93.6	<b>91.7</b>	92.4	96.2	95.3	94.5
ERGM [71]	90.9	93.4	93.1	95.7	90.1	90.3	91.5	95.1	94.0	92.5
MTG [75]	90.6	93.6	94.4	<b>97.8</b>	92.4	89.2	92.0	96.5	<b>95.9</b>	<b>95.4</b>

Additionally, the table filling model shows results just below the parameter-sharing model, which can be explained by its need to enumerate all possible entity-relation pairs and construct a relation matrix for each triplet. While this approach performs well in more complex environments, it inevitably increases computational demand, thus reducing efficiency. In the case of sentences with fewer triplets, the tagging model demonstrates relatively stable performance; however, when  $N > 5$ , its performance declines significantly. This suggests that, in scenarios involving multiple triplets, the model struggles to fully utilize contextual information, revealing limitations when processing complex sentences. As for the sequence-to-sequence model, it generally performs well, with the RFBN model standing out in particular when handling  $N < 3$ . This model demonstrates superior performance compared to others, which underscores its advantage in more complex extraction scenarios.

## 7 Current State-of-the-Art & Trends

Despite the significant advancements achieved in accuracy and efficiency by triplet joint extraction models, their limitations remain noteworthy. Primarily, the issue of computational complexity often restricts their practical application. Table 8 shows the computational complexity and applicable scenarios of different relation triplet extraction techniques; particularly when handling large-scale data, the training and inference processes of these models can consume substantial computational resources, resulting in extended response times. Additionally, concerns regarding scalability arise, as many existing methods struggle to maintain consistent performance in the face of an ever-increasing variety of entities and relation types. More critically, in resource-constrained environments, the performance of these models may decline markedly, given that their updates typically necessitate extensive annotated data and computational resources.

In addition, as the complexity of relation triplet extraction models continues to increase, the computational resources required for training these models, along with their environmental impact, have become

increasingly critical concerns. The training of large-scale deep learning models typically relies on high-performance computing hardware, such as GPUs or TPUs, which consume substantial amounts of electricity and contribute significantly to carbon emissions. Particularly when dealing with vast amounts of textual data, the training process can extend over several days or even weeks, further escalating energy consumption and the environmental burden [76]. Moreover, in order to achieve high model performance, researchers often depend on large datasets and distributed computing frameworks, which not only place greater demands on computational power but also increase energy consumption associated with data storage and transmission. Given the growing importance of sustainability, finding ways to optimize the use of computational resources while reducing environmental impact, without compromising model performance, has emerged as a key research direction in the field.

**Table 8:** Computational complexity and applicable scenarios of relation triplet extraction techniques,  $n$  is the number of entities in the text

Methods	Computational complexity	Applicable scenarios
Table filling	$O(n^2)$	Structured output, suitable for text data with fixed formats or structures.
Tagging	$O(n) \sim O(n^2)$	Suitable for tasks that extract relationships step by step, ideal for simple relation extraction tasks.
Sequence-to-sequence	$O(n^2) \sim O(n^3)$	Applicable to joint extraction of long texts and complex relationships.
Parameter sharing	$O(n^2) \sim O(n^3)$	Sharing features in multi-task learning to enhance the overall performance of the model.

In recent years, Transformer-based architectures such as BERT [77] and GPT-3 have demonstrated immense potential in the field of relational triplet extraction. BERT, with its bidirectional encoder and pretraining-finetuning framework, effectively captures contextual information, thereby enhancing the accuracy and generalization capability of relation extraction tasks. On the other hand, GPT-3 leverages its powerful generative capacity and few-shot learning advantage, allowing it to excel in complex and low-resource environments. Both models combine robust language modeling abilities and exhibit unique strengths when dealing with intricate sentence structures and multiple relations. Furthermore, the integration of technologies like Graph Neural Networks (GNN) can further enhance model performance, thereby positioning Transformer-based architectures as highly promising and valuable in the context of relational triplet extraction tasks, with significant potential for both application and research advancement.

On the application front, the potential of joint relation triplet extraction technology is vast. It significantly improves answer accuracy in intelligent question-answering systems, thereby serving as a crucial foundation for information extraction and knowledge graph construction. In personalized recommendation systems, it enhances user experience, enabling more precise recommendations. Furthermore, in dialogue systems, it bolsters context comprehension, resulting in more natural and fluid exchanges. In the realm of social network analysis, this technology aids in identifying interactions among users, providing valuable data support for marketing strategies [78]. In biomedical fields, it contributes to drug development by extracting key relations that drive scientific discoveries. Collectively, these applications not only accelerate the rapid advancement of relation triplet extraction technology but also lay a vital technological foundation for the intelligent transformation of various industries. As the field evolves, an increasing array of innovative applications is anticipated, highlighting its profound societal value and academic significance.

## 8 Outlook of Future Works

The future development prospects of joint relation triplet extraction undoubtedly brim with vast potential. On the technical front, the growing emphasis on interpretability and transparency is increasingly vital; this focus not only fosters user trust in the decision-making processes of models but also deepens the understanding of deep learning mechanisms. Simultaneously, the emergence of multimodal learning enables models to integrate diverse data sources such as text, images, and videos, thus enhancing their performance in navigating complex scenarios [79]. Furthermore, the incorporation of continuous learning capabilities is poised to become a crucial factor, allowing models to self-update and adapt flexibly to the rapidly changing landscape of knowledge and environments. In addition, combining few-shot learning and self-supervised learning with relation triplet joint extraction techniques has the potential to significantly enhance the generalization capabilities of relation extraction systems. This is particularly true in low-resource scenarios, where such approaches can improve the scalability of models, enabling them to perform effectively even when labeled data is scarce or difficult to obtain. Additionally, the adaptation to cross-domain and cross-language contexts is likely to emerge as a focal point of research, thereby broadening the applicability of these models across various fields and languages [80].

In terms of applications, the technology of relation triplet extraction is set to intertwine more closely with the construction of knowledge graphs, facilitating ongoing advancements in intelligent question-answering, personalized recommendations, social network analysis, and biomedical information extraction. This synergy is expected not only to optimize user experiences and enhance the efficiency of information retrieval and decision support but also to drive the intelligent transformation of sectors such as corporate decision-making, marketing, and scientific research. As technology continues to evolve and application scenarios diversify, relation triplet extraction is positioned to reveal its profound social value while igniting new avenues for exploration and innovation within academic research, thus underscoring its pivotal role in advancing the construction of a smart society.

## 9 Conclusion

This paper provides a brief review of joint extraction technologies for relational triples, categorizing them into two major approaches: joint decoding and parameter sharing. Joint decoding methods can be further divided into three types: table filling, tagging, and sequence-to-sequence. While joint decoding is effective for extracting triples from short sequence texts and allows for easier overall task model adjustments, its performance diminishes when dealing with long sequence texts and complex texts. On the other hand, parameter sharing methods enable the two subtasks of relational triple extraction to share some or all of the parameters, thereby reducing redundancy in model parameters. However, modifying shared parameters affects all tasks, thus limiting the model's flexibility. With the continuous advancement of deep learning and natural language processing technologies, future research directions primarily focus on optimizing both the performance of joint decoding and parameter sharing methods across different text scenarios. Additionally, enhancing the extraction of cross-document relational triples and exploring few-shot learning and transfer learning will also be crucial for improving adaptability in low-resource languages and domains.

**Acknowledgment:** The authors are grateful to all the editors and anonymous reviewers for their comments and suggestions and thank all the members who have contributed to this work with us.

**Funding Statement:** The research leading to these results received funding from Key Areas Science and Technology Research Plan of Xinjiang Production And Construction Corps Financial Science and Technology Plan Project under Grant Agreement No. 2023AB048 for the project: Research and Application Demonstration of Data-driven Elderly Care System.

**Author Contributions:** The authors confirm contribution to the paper as follows: study conception and design: Chenglong Mi; data collection: Chenglong Mi; analysis and interpretation of results: Chenglong Mi; draft manuscript preparation: Chenglong Mi; manuscript guidance and revision: Huaibin Qin, Quan Qi, Pengxiang Zuo. All authors reviewed the results and approved the final version of the manuscript.

**Availability of Data and Materials:** Not applicable.

**Ethics Approval:** Not applicable.

**Conflicts of Interest:** The authors declare no conflicts of interest to report regarding the present study.

## References

1. Tjong Kim Sang EF, De Meulder F. Introduction to the CoNLL-2003 shared task: language-independent named entity recognition. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003; 2003. p. 142–7.
2. Ratinov L, Roth D. Design challenges and misconceptions in named entity recognition. In: Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009); 2009 Jun; Boulder, CO, USA. p. 147–55.
3. Zelenko D, Aone C, Richardella A. Kernel methods for relation extraction. *J Mach Learn Res.* 2003 Feb;3(3):1083–106. doi:10.3115/1118693.1118703.
4. Bunescu R, Mooney R. A shortest path dependency kernel for relation extraction. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing; 2005 Oct; Vancouver, BC, Canada. p. 724–31.
5. Nadeau D, Sekine S. A survey of named entity recognition and classification. *Lingvist Investig.* 2007;30(1):3–26. doi:10.1075/li.30.1.03nad.
6. Chan YS, Roth D. Exploiting syntactico-semantic structures for relation extraction. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies; 2011 Jun; Portland, OR, USA. p. 551–60.
7. Li Q, Ji H. Incremental joint extraction of entity mentions and relations. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014); 2014; Baltimore, MD, USA. p. 402–12.
8. Yin B, Sun Y, Wang Y. Entity relation extraction method based on fusion of multiple information and attention mechanism. In: 2020 IEEE 6th International Conference on Computer and Communications (ICCC); 2020. p. 2485–90.
9. Nayak T, Majumder N, Goyal P, Poria S. Deep neural approaches to relation triplets extraction: a comprehensive survey. *Cognit Comput.* 2021;13(5):1215–32. doi:10.1007/s12559-021-09917-7.
10. Hendrickx I, Kim SN, Kozareva Z, Nakov P, Séaghdha DÓ, Padó S, et al. SemEval-2010 Task 8: multi-way classification of semantic relations between pairs of nominals. In: Proceedings of the 5th International Workshop on Semantic Evaluation: Recent Achievements and Future Directions (SEW-2009); 2009; Boulder, CO, USA. p. 94–9.
11. Walker C, Strassel S, Medero J, Maeda K. ACE 2005 multilingual training corpus. Philadelphia, PA, USA: Linguistic Data Consortium; 2006.
12. Riedel S, Yao L, McCallum A. Modeling relations and their mentions without labeled text. In: Machine Learning and Knowledge Discovery in Databases, European Conference, ECML PKDD 2010; 2010; Barcelona, Spain. p. 148–63.
13. Gardent C, Shimorina A, Narayan S, Perez-Beltrachini L. Creating training corpora for nlg micro-planners. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2017. p. 179–88.
14. Liu P, Guo Y, Wang F, Li G. Chinese named entity recognition: the state of the art. *Neurocomputing.* 2022;473(1):37–53. doi:10.1016/j.neucom.2021.10.101.

15. Wang Y, Yu B, Zhang Y, Liu T, Zhu H, Sun L. TPLinker: single-stage joint extraction of entities and relations through token pair linking. In: Proceedings of the 28th International Conference on Computational Linguistics; 2020 Dec; Barcelona, Spain. p. 1572–82.
16. Wang Y, Sun C, Wu Y, Zhou H, Li L, Yan J. UniRE: a unified label space for entity relation 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 Aug; Online. p. 220–31.
17. Hu Y, Zhao J, Li M, Yuan L, Zou Q, Zhang R. Hybrid deep relation matrix bidirectional approach for relational triple extraction. In: 2024 27th International Conference on Computer Supported Cooperative Work in Design (CSCWD); 2024. p. 1728–33.
18. Ren F, Zhang L, Yin S, Zhao X, Liu S, Li B, et al. A novel global feature-oriented relational triple extraction model based on table filling. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing; 2021 Nov; Punta Cana, Dominican Republic. p. 2646–56.
19. Yan Z, Zhang C, Fu J, Zhang Q, Wei Z. A partition filter network for joint entity and relation extraction. In: Proceedings of the 30th International Conference on Information and Knowledge Management (CIKM); 2021; Punta Cana, Dominican Republic. p. 185–97.
20. Liu J, Chen S, Wang B, Zhang J, Li N, Xu T. Attention as relation: learning supervised multi-head self-attention for relation extraction. In: Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence; 2021. p. 3787–93.
21. Shang Y-M, Huang H, Mao X. OneRel: joint entity and relation extraction with one module in one step. Proc AAAI Conf Artif Intell. 2022;36(10):11285–93. doi:10.1609/aaai.v36i10.21379.
22. Wang J, Lu W. Two are better than one: joint entity and relation extraction with table-sequence encoders. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020 Nov. p. 1706–21.
23. Ning J, Yang Z, Sun Y, Wang Z, Lin H. OD-RTE: a one-stage object detection framework for relational triple extraction. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); 2023 Jul; Toronto, ON, Canada. p. 11120–35.
24. Zhang Z, Liu H, Yang J, Li X. Relational prompt-based single-module single-step model for relational triple extraction. J King Saud Univ-Comput Inf Sci. 2023;35:101748. doi:10.1016/j.jksuci.2023.101748.
25. Wang X, Wang D, Ji F. A span-based model for joint entity and relation extraction with relational graphs. In: 2020 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom); 2020; Exeter, UK. p. 513–20.
26. Han D, Zheng Z, Zhao H, Feng S, Pang H. Span-based single-stage joint entity-relation extraction model. PLoS One. 2023 Feb;18(2):1–14. doi:10.1371/journal.pone.0281055.
27. Tian X, Jing L, He L, Liu F. 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 (Volume 1: Long Papers); 2021 Aug. p. 4851–61.
28. Tang W, Xu B, Zhao Y, Mao Z, Liu Y, Liao Y, et al. Unified representation and interaction for joint relational triple extraction. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing; 2022 Dec; Abu Dhabi, United Arab Emirates. p. 7087–99.
29. Wang Z, Nie H, Zheng W, Wang Y, Li X. A novel tensor learning model for joint relational triplet extraction. IEEE Trans Cybern. 2024 Apr;54(4):2483–94. doi:10.1109/TCYB.2023.3265851.
30. Zheng H, Wen R, Chen X, Yang Y, Zhang Y, Zhang Z, et al. 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 Aug. p. 6225–35.
31. Cheng J, Zhang T, Zhang S, Ren H, Yu G, Zhang X, et al. A cascade dual-decoder model for joint entity and relation extraction. IEEE Trans Emerg Top Comput. Intell. 2024 Jun;1–13. doi:10.1109/TETCI.2024.3406440.

32. Qiao B, Zou Z, Huang Y, Fang K, Zhu X, Chen Y. A joint model for entity and relation extraction based on BERT. *Neural Comput Appl.* 2022 Mar;34(5):3471–81. doi:10.1007/s00521-021-05815-z.
33. Wei Z, Su J, Wang Y, Tian Y, Chang Y. A novel cascade binary tagging framework for relational triple extraction. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*; 2020 Jul. p. 1476–88.
34. Yuan Y, Zhou X, Pan S, Zhu Q, Song Z, Guo L. A relation-specific attention network for joint entity and relation extraction. In: *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence*; 2021. p. 4054–60.
35. Ren F, Zhang L, Zhao X, Yin S, Liu S, Li B. A simple but effective bidirectional extraction framework for relational triple extraction. *arXiv:2112.04940v2*. 2021.
36. Xu M, Pi D, Cao J, Yuan S. A novel entity joint annotation relation extraction model. *Appl Intell.* 2022;52(11):12754–70. doi:10.1007/s10489-021-03002-0.
37. Chen T, Zhou L, Wang N, Chen X. Joint entity and relation extraction with position-aware attention and relation embedding. *Appl Soft Comput.* 2022;119:108604. doi:10.1016/j.asoc.2022.108604.
38. Hang T, Feng J, Yan L, Wang Y, Lu J. Joint extraction of entities and relations using multilabel tagging and relational alignment. *Neural Comput Appl.* 2022;34(8):1–16. doi:10.1007/s00521-021-06685-1.
39. Zhang N, Deng S, Ye H, Zhang W, Chen H. Robust triple extraction with cascade bidirectional capsule network. *Expert Syst Appl.* 2022;187:115806. doi:10.1016/j.eswa.2021.115806.
40. Wu H, Shi X. Synchronous dual network with cross-type attention for joint entity and relation extraction. In: *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*; 2021 Nov; Punta Cana, Dominican Republic. p. 2769–79.
41. Nayak T, Ng HT. Effective modeling of encoder-decoder architecture for joint entity and relation extraction. *Proc AAAI Conf Artif Intell.* 2020 Apr;34(5):8528–35. doi:10.1609/aaai.v34i05.6374.
42. Zeng X, Zeng D, He S, Liu K, Zhao J. Extracting relational facts by an end-to-end neural model with copy mechanism. In: *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*; 2018 Jul; Melbourne, Australia. p. 506–14.
43. Sui D, Zeng X, Chen Y, Liu K, Zhao J. Joint entity and relation extraction with set prediction networks. *IEEE Trans Neural Netw Learn Syst.* 2024 Sep;35(9):12784–95. doi:10.1109/TNNLS.2023.3264735.
44. Zhao K, Xu H, Cheng Y, Li X, Gao K. Representation iterative fusion based on heterogeneous graph neural network for joint entity and relation extraction. *Knowl Based Syst.* 2021;219:106888. doi:10.1016/j.knsys.2021.106888.
45. Zhang RH, Liu Q, Fan AX, Ji H, Zeng D, Cheng F, et al. Minimize exposure bias of Seq2Seq models in joint entity and relation extraction. In: *Findings of the Association for Computational Linguistics: EMNLP 2020*; 2020 Nov. p. 236–46.
46. Zhang Z, Yang J, Liu H, Hu P. BTDM: a bi-directional translating decoding model-based relational triple extraction. *Appl Sci.* 2023;13(7):4447. doi:10.3390/app13074447.
47. Chang H, Xu H, van Genabith J, Xiong D, Zan H. JoinER-BART: joint entity and relation extraction with constrained decoding, representation reuse and fusion. *IEEE/ACM Trans Audio Speech Lang Process.* 2023;31:3603–16. doi:10.1109/TASLP.2023.3310879.
48. Ye H, Zhang N, Deng S, Chen M, Tan C, Huang F, et al. Contrastive triple extraction with generative transformer. *Proc AAAI Conf Artif Intell.* 2021 May;35(16):14257–65. doi:10.1609/aaai.v35i16.17677.
49. Wang T, Yang W, Wu T, Yang C, Liang J, Wang H, et al. Joint entity and relation extraction with fusion of multi-feature semantics. *J Intell Inform Syst.* 2024;61(5):1–22. doi:10.1007/s10844-024-00871-y.
50. Chen Y, Zhang Y, Hu C, Huang Y. Jointly extracting explicit and implicit relational triples with reasoning pattern enhanced binary pointer network. In: *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*; 2021 Jun. p. 5694–703.
51. Tan Z, Shen Y, Hu X, Zhang W, Cheng X, Lu W, et al. Query-based instance discrimination network for relational triple extraction. In: *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*; 2022 Dec; Abu Dhabi, United Arab Emirates. p. 7677–90.
52. Zhang S, Ng P, Wang Z, Xiang B. REKnow: enhanced knowledge for joint entity and relation extraction. *arXiv:2206.05123*. 2022.

53. Shang Y-M, Huang H, Sun X, Wei W, Mao X-L. Relational triple extraction: one step is enough. In: Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22; 2022 Jul. p. 4360–6.
54. Li Z, Fu L, Wang X, Zhang H, Zhou C. RFBFN: a relation-first blank filling network for joint relational triple extraction. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop; 2022 May; Dublin, Ireland. p. 10–20.
55. Lai T, Cheng L, Wang D, Ye H, Zhang W. RMAN: relational multi-head attention neural network for joint extraction of entities and relations. *Appl Intell.* 2022;52(3):3132–42. doi:10.1007/s10489-021-02600-2.
56. Liang Z, Du J. Sequence to sequence learning for joint extraction of entities and relations. *Neurocomputing.* 2022;501(1):480–8. doi:10.1016/j.neucom.2022.05.074.
57. Li X, Luo X, Dong C, Yang D, Luan B, He Z. TDEER: an efficient translating decoding schema for joint extraction of entities and relations. In: Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing; 2021 Nov; Punta Cana, Dominican Republic. p. 8055–64.
58. Geng Z, Zhang Y, Han Y. Joint entity and relation extraction model based on rich semantics. *Neurocomputing.* 2021;429(10):132–40. doi:10.1016/j.neucom.2020.12.037.
59. Gao C, Zhang X, Liu H, Yun W, Jiang J. A joint extraction model of entities and relations based on relation decomposition. *Int J Mach Learn Cybern.* 2022;13(7):1833–45. doi:10.1007/s13042-021-01491-6.
60. Zeng D, Zhang H, Liu Q. CopyMTL: copy mechanism for joint extraction of entities and relations with multi-task learning. *Proc AAAI Conf Artif Intell.* 2020 Apr;34(5):9507–14. doi:10.1609/aaai.v34i05.6495.
61. Li R, La K Lei J, Huang L, Ouyang J, Shu Y, et al. Joint extraction model of entity relations based on decomposition strategy. *Sci Rep.* 2024;14(1):59. doi:10.1038/s41598-024-51559-w.
62. Yu B, Zhang Z, Shu X, Liu T, Wang T, Wang B, et al. Joint extraction of entities and relations based on a novel decomposition strategy. In: *ECAI 2020*; 2020. p. 2282–9.
63. Sun K, Zhang R, Mensah S, Mao Y, Liu X. Recurrent interaction network for jointly extracting entities and classifying relations. In: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP); 2020 Nov. p. 3722–32.
64. Zhang M, Wang J, Qu J, Li Z, Liu A, Zhao L, et al. A coarse-to-fine framework for entity-relation joint extraction. In: 2024 IEEE 40th International Conference on Data Engineering (ICDE); 2024. p. 1009–22.
65. Wang Y, Liu X, Kong W, Yu H-T, Racharak T, Kim K-S, et al. A decoupling and aggregating framework for joint extraction of entities and relations. *IEEE Access.* 2024;12:103313–28. doi:10.1109/ACCESS.2024.3420877.
66. Yang Y, Li X, Li X. A relation-guided attention mechanism for relational triple extraction. In: 2021 International Joint Conference on Neural Networks (IJCNN); 2021. p. 1–8.
67. Duan G, Miao J, Huang T, Luo W, Hu D. A relational adaptive neural model for joint entity and relation extraction. *Front Neurobot.* 2021;15:13195. doi:10.3389/fnbot.2021.635492.
68. Wang Z, Yang L, Yang J, Li T, He L, Li Z. A triple relation network for joint entity and relation extraction. *Electronics.* 2022;11(10):1535. doi:10.3390/electronics11101535.
69. Wang C, Li A, Tu H, Wang Y, Li C, Zhao X. An advanced bert-based decomposition method for joint extraction of entities and relations. In: 2020 IEEE Fifth International Conference on Data Science in Cyberspace (DSC); 2020. p. 82–8.
70. Yang Y, Zhou S, Liu Y. Bidirectional relation-guided attention network with semantics and knowledge for relational triple extraction. *Expert Syst Appl.* 2023;224:119905. doi:10.1016/j.eswa.2023.119905.
71. Gao C, Zhang X, Li LY, Li JH, Zhu R, Du KP, et al. Ergm: a multi-stage joint entity and relation extraction with global entity match. *Knowl Based Syst.* 2023;271:110550. doi:10.1016/j.knosys.2023.110550.
72. Sun K, Zhang R, Mensah S, Mao Y, Liu X. Progressive multi-task learning with controlled information flow for joint entity and relation extraction. *Proc AAAI Conf Artif Intell.* 2021 May;35(15):13851. doi:10.1609/aaai.v35i15.17632.
73. Huang H, Shang Y-M, Sun X, Wei W, Mao X. Three birds, one stone: a novel translation based framework for joint entity and relation extraction. *Knowl Based Syst.* 2022;236:107677. doi:10.1016/j.knosys.2021.107677.

74. Zhuang C, Zhang N, Jin X, Li Z, Deng S, Chen H. Joint extraction of triple knowledge based on relation priority. In: 2020 IEEE International Conference on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom); 2020. p. 562–9.
75. Chen Y, Zhang Y, Huang Y. Learning reasoning patterns for relational triple extraction with mutual generation of text and graph. In: Findings of the Association for Computational Linguistics: ACL 2022; 2022 May; Dublin, Ireland. p. 1638–47.
76. Lin Y, Ji H, Huang F, Wu L. A joint neural model for information extraction with global features. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics; 2020 Jul. p. 7999–8009.
77. Devlin J, Chang M-W, Lee K, Toutanova K. BERT: pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies; 2019 Jun; MN, USA. Vol. 1, p. 4171–86.
78. Shen Y, Ma X, Tang Y, Lu W. A trigger-sense memory flow framework for joint entity and relation extraction. In: Proceedings of the Web Conference 2021; 2021; New York, NY, USA. p. 1704–15.
79. Zhao S, Hu M, Cai Z, Liu F. Modeling dense cross-modal interactions for joint entity-relation extraction. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20; 2020 Jul. p. 4032–8.
80. Zhong Z, Chen D. 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; 2021 Jun. p. 50–61.