A Cryptography-Domain Relation Extraction Method Based on Retrieval Augmentation and Chain of Thought

Leer Bao, Wei Zhang

Abstract— *Abstract*—**Relation extraction is a crucial task in natural language processing, aimed at identifying entities and their relationships from text, and holds particular significance in the cryptography domain due to its involvement with complex multi-entity interactions. This paper addresses the challenges of relation extraction in cryptography, such as intricate terminology, implicit relationships, data scarcity, and the demand for high precision, by proposing a method based on retrieval augmentation and chain-of-thought optimization. By integrating chain-of-thought prompt optimization, example retrieval enhancement, and dependency analysis techniques, this approach significantly improves extraction performance in low-resource environments. Additionally, a high-quality cryptography-domain dataset containing 24 entity relationship categories is constructed, providing robust support for related research.**

Tailored to the unique characteristics of the cryptography domain, this paper first designs a chain-of-thought-based prompt template with a self-checking mechanism to guide the model through step-by-step reasoning, enhancing the accuracy of extracting complex relationships. Second, it employs S-BERT embeddings along with a weighted ranking strategy using cosine similarity and the DICE coefficient to dynamically retrieve and inject the most relevant examples into the prompt template, improving the model's adaptability in data-scarce scenarios. Then, a rule-based retrieval method precisely injects necessary background knowledge. Finally, StanfordCoreNLP is utilized to extract dependency relationships as structured knowledge, further optimizing the model's understanding of implicit relationships.

Experimental results demonstrate that this method, implemented on the ChatGLM4-Plus model, achieves an F1-score of 70.62%, significantly outperforming traditional methods and unoptimized large language model baselines. Ablation studies confirm the effectiveness of each module and their synergistic contributions, underscoring the method's value in enhancing the precision and robustness of relation extraction in the cryptography domain. This approach not only provides an efficient tool for knowledge graph construction and information retrieval in cryptography but also offers a replicable optimization strategy for relation extraction tasks in other highly specialized domains.et

Index Terms—**Relation Extraction; Large Language Model; Chain-of-Thought; Retrieval Augmentation**

I. INTRODUCTION

Relation extraction is an important task in the field of natural language processing, aiming to extract entities and the relationships between them from textual content. Each relationship is represented as a triplet (head entity, relationship, tail entity), consisting of two entities and the relationship between them. Relation extraction helps to extract structured information from large volumes of unstructured text, thereby uncovering hidden value within the data. It can be applied to many downstream applications [1][2], such as knowledge graph completion [3] and alignment[4], question-answering systems[5], and information retrieval[6].

In recent years, the development of deep neural networks and pre-trained language model has significantly improved the performance of relation extraction. These methods can primarily be divided into two categories: pipeline-based relation extraction methods[7]and joint relation extraction methods[8]-[10]. Pipeline-based methods extract entities and relationships between entities from unstructured text in two separate stages. First, entities are identified from the text, and then the relationships between entity pairs are detected. In the early stages, pipeline-based relation extraction methods [11]-[13]mainly used named entity recognition (NER) tools to extract entities, and then applied supervised learning algorithms with feature engineering to classify the relationships between entity pairs. Such relation extraction methods[14][15] typically assume that the target entities have already been identified, and the relation extraction model only needs to predict the relationship between entity pairs. However, because the entity recognition and relation extraction processes are separate, pipeline-based methods are often susceptible to error propagation, meaning that the relationship classification may be influenced by errors introduced during the entity recognition process.

On the other hand, joint extraction methods aim to efficiently extract entities and their relationships by jointly modeling entity recognition and relation classification within a unified framework. This method requires the system to accurately identify and distinguish the various entities and relationships involved. Compared to pipeline-based methods, joint extraction can more effectively alleviate error propagation for the following reasons: First, the joint training allows the model to optimize the representations of both tasks simultaneously, reducing interference between tasks caused by errors from one task. Second, joint extraction methods directly optimize for the overall goal (e.g., the probability of obtaining the correct triplet). Finally, joint learning enables the model to utilize complementary information between the two tasks to correct potential errors from earlier stages. Currently, joint methods have developed into various technical approaches, including span-based methods[16][17], sequence-to-sequence methods[18], machine reading comprehension-based methods [19], and sequence labeling methods[20][21].

Manuscript received April 06, 2025

Leer Bao, College of Computer Science, Beijing Information Science and Technology University, BeiJing, China.

Wei Zhang, College of Computer Science, Beijing Information Science and Technology University, BeiJing, China.

Relation extraction holds particular significance in the field of cryptography, where cryptographic systems involve complex interactions among multiple entities (such as cryptographic devices, algorithms, protocols, etc.). Accurately identifying these relationships is crucial for both the research and application of cryptographic technologies. However, relation extraction in the cryptography domain faces numerous challenges.

First, the terminology in the cryptography domain is intricate, featuring a plethora of specialized vocabulary with diverse expressions, including abbreviations, compound words, and context-dependent variants. This complexity makes it difficult for traditional relation extraction methods to accurately identify and match relevant concepts. Second, the entity relationships within cryptographic systems are highly intricate, encompassing various types such as protection and support relationships. These relationships are often implicitly expressed through passive voice, nested structures, or cross-sentence dependencies, posing additional difficulties for relation extraction tasks. Furthermore, traditional statistical and machine learning-based approaches typically rely on large-scale annotated datasets. However, the specialized nature of the cryptography domain results in a scarcity of high-quality data, leading to suboptimal performance of these methods in this context. Most critically, the cryptography domain demands extremely high accuracy and security of information. Any erroneous relationship identification could compromise the security of cryptographic systems. Existing methods still face bottlenecks in low-resource, highly specialized domains, struggling to meet the stringent requirements for high precision and robustness.

To address these challenges, this chapter proposes a relation extraction method tailored for the cryptography domain, leveraging retrieval augmentation and chain-of-thought prompt optimization. By incorporating knowledge enhancement, dependency analysis, and example retrieval techniques, this approach effectively improves the accuracy and stability of relation extraction in the cryptography domain. Specifically, the main contributions of this study are as follows:

A cryptography-domain relation extraction method based on chain-of-thought(CoT)[22] prompt optimization: This method employs structured chain-of-thought prompts and a self-checking mechanism to guide large-scale models in step-by-step reasoning, ensuring logical consistency in the relationship inference process and enhancing the accuracy and robustness of extraction.

An example retrieval augmentation[23] mechanism designed for cryptography-domain relation extraction: Utilizing S-BERT to embed task text and example text, combined with cosine similarity and DICE coefficient calculations, this mechanism retrieves the most relevant examples from a pre-constructed cryptography-domain example library. This enhances the adaptability of large-scale models to specialized tasks and improves relation extraction performance under data-scarce conditions.

A dependency-based semantic enhancement mechanism: By using StanfordCoreNLP to extract dependency relationships from text and injecting them as structured knowledge into the chain-of-thought prompt template, this mechanism helps the model better understand implicit relationships in complex texts with greater accuracy.

Construction of a high-quality cryptography-domain relation extraction dataset containing 24 entity relationship categories: Experimental results on this dataset demonstrate that the proposed method, built upon ChatGLM4-Plus, achieves an F1-score of 70.62%, outperforming traditional methods and large-scale model baselines, significantly enhancing the model's relation extraction capabilities.

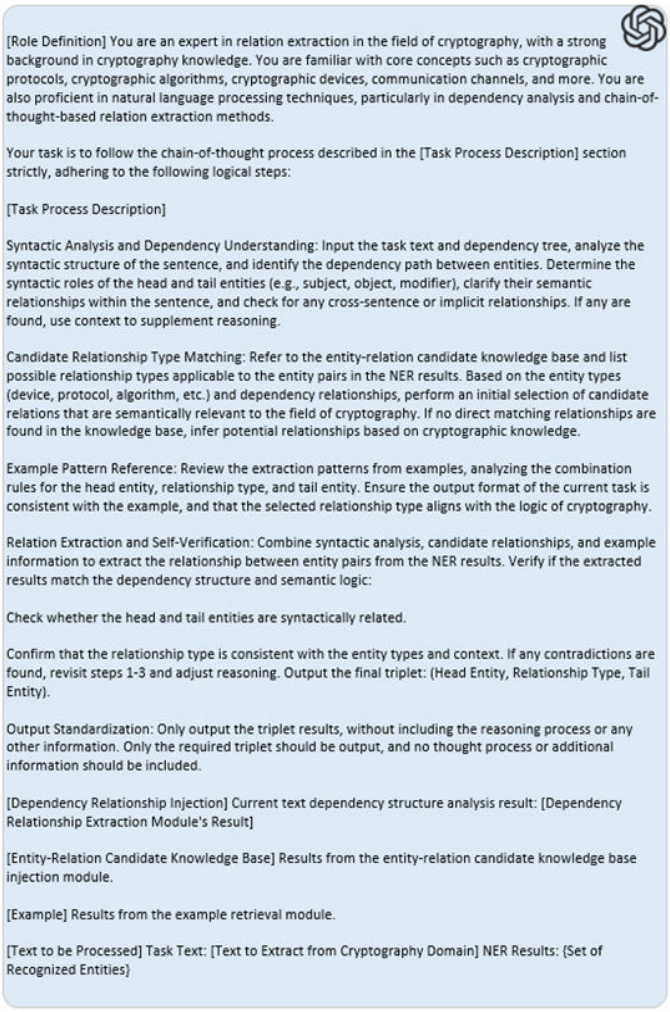
II. Relation Extraction Technology in the Cryptography Domain Based on Retrieval Augmentation and Chain-of-Thought

To address the numerous challenges faced by relation extraction in the cryptography domain, this study proposes a prompt fine-tuning method based on chain-of-thought technology and self-checking techniques. This method leverages CoT prompt techniques to guide the model in step-by-step reasoning, integrating precise task-specific knowledge, dependency relationships, and optimal example injections into the CoT prompt template. This ensures consistency and accuracy in the reasoning process, thereby improving the precision and robustness of relation extraction. As illustrated in Figure 1, the overall framework of the proposed method consists of four components: a CoT prompt template tailored for cryptography-domain relation extraction tasks, a precise entity relationship knowledge injection module, a dependency relationship extraction module, and an example retrieval module.

First, a CoT prompt template is designed specifically for relation extraction tasks in the cryptography domain. By incorporating structured reasoning and a self-checking mechanism, this template enhances the stability of the model's reasoning process, thereby improving the accuracy of relation extraction. Second, an entity relationship knowledge injection module is introduced. Given the complexity and specialized nature of relationships in the cryptography domain, this module dynamically matches and selects the most relevant knowledge from a pre-constructed cryptography-domain entity relationship knowledge base. For instance, relationships such as (cryptographic device, cryptographic algorithm, support, encryption machine executing RSA-2048 algorithm) are injected into the prompt template based on the entity types present in the task text, enhancing the model's understanding of domain-specific relationships. Third, a dependency relationship extraction module is incorporated, which extracts dependency relationships from the task text and injects them as structured knowledge into the prompt template. This improves the large-scale model's precision in reasoning about implicit relationships. Finally, the example retrieval module employs an algorithm combining cosine similarity and the DICE coefficient for similarity ranking. It retrieves the most relevant examples from a pre-constructed cryptography-domain relation extraction example library and injects them into the prompt template. This provides contextual support for low-resource scenarios, optimizing the model's ability to recognize diverse entity relationships.

A. Chain-of-Thought Prompt Template for Relation Extraction Tasks in the Cryptography Domain

The particular difficulties of relation extraction in the cryptography domain lie in the complexity of terminology, the implicit nature of relationships, nested and cross-sentence dependencies, and data scarcity. To tackle these issues, a prompt template based on chain-of-thought (CoT) technology is designed, as shown in Figure 2. The introduction of CoT technology aims to guide the model in gradually understanding and reasoning through complex tasks, ensuring that the reasoning process follows a clear logical chain. Combined with a self-checking mechanism, this approach helps the model verify the results of each reasoning step in real time, reducing the error rate during inference.


[Role Definition] You are an expert in relation extraction in the field of cryptography, with a strong background in cryptography knowledge. You are familiar with core concepts such as cryptographic protocols, cryptographic algorithms, cryptographic devices, communication channels, and more. You are also proficient in natural language processing techniques, particularly in dependency analysis and chain-of-thought-based relation extraction methods.

Your task is to follow the chain-of-thought process described in the [Task Process Description] section strictly, adhering to the following logical steps:

[Task Process Description]

Syntactic Analysis and Dependency Understanding: Input the task text and dependency tree, analyze the syntactic structure of the sentence, and identify the dependency path between entities. Determine the syntactic roles of the head and tail entities (e.g., subject, object, modifier), clarify their semantic relationships within the sentence, and check for any cross-sentence or implicit relationships. If any are found, use context to supplement reasoning.

Candidate Relationship Type Matching: Refer to the entity-relation candidate knowledge base and list possible relationship types applicable to the entity pairs in the NER results. Based on the entity types (device, protocol, algorithm, etc.) and dependency relationships, perform an initial selection of candidate relations that are semantically relevant to the field of cryptography. If no direct matching relationships are found in the knowledge base, infer potential relationships based on cryptographic knowledge.

Example Pattern Reference: Review the extraction patterns from examples, analyzing the combination rules for the head entity, relationship type, and tail entity. Ensure the output format of the current task is consistent with the example, and that the selected relationship type aligns with the logic of cryptography.

Relation Extraction and Self-Verification: Combine syntactic analysis, candidate relationships, and example information to extract the relationship between entity pairs from the NER results. Verify if the extracted results match the dependency structure and semantic logic:

Check whether the head and tail entities are syntactically related.

Confirm that the relationship type is consistent with the entity types and context. If any contradictions are found, revisit steps 1-3 and adjust reasoning. Output the final triplet: (Head Entity, Relationship Type, Tail Entity).

Output Standardization: Only output the triplet results, without including the reasoning process or any other information. Only the required triplet should be output, and no thought process or additional information should be included.

[Dependency Relationship Injection] Current text dependency structure analysis result: [Dependency Relationship Extraction Module's Result]

[Entity-Relation Candidate Knowledge Base] Results from the entity-relation candidate knowledge base injection module.

[Example] Results from the example retrieval module.

[Text to be Processed] Task Text: [Text to Extract from Cryptography Domain] NER Results: {Set of Recognized Entities}

Figure 2: Chain-of-Thought Prompt for Cryptography Domain Relation Extraction Task

Specifically, the task workflow is first described in detail using CoT technology. This description is intended to guide the model through a series of clear reasoning steps to progressively extract entity relationships from the input text. The template reserves slots for knowledge injection from the subsequent three modules, allowing the enhanced results from these modules to be incorporated into designated positions. This ensures that each reasoning step of the model is augmented with dependency relationships, precise entity relationship knowledge, and the most relevant task-specific examples, thereby improving the model's relation extraction performance. On the other hand, the self-checking mechanism employs rule-based validation. After each reasoning step, predefined rules are applied to check the results. If an error is detected, the model revisits the previous reasoning steps to ensure the accuracy of the extracted relationships.

B. Precise Entity Relationship Knowledge Injection Module

In the cryptography domain, entities exhibit complex multi-level relationships, such as "support" or "invocation" relationships between cryptographic devices and algorithms. These entity relationships are highly context-dependent; for instance, a communication channel may be associated with either a cryptographic device or a protocol depending on the scenario. Additionally, there are long-range dependency issues, such as the authorization of a cryptographic certificate to a system, which requires forming a semantic chain across multiple nodes. To address these challenges, this section designs a dynamic retrieval-augmented method that selects and injects the most relevant entity relationship knowledge based on the task input, assisting large-scale models in performing relation extraction efficiently and accurately.

In this component, entity relationship knowledge specific to the cryptography domain is injected into the prompt template through retrieval augmentation. As shown in Table 1, this knowledge includes entities, the relationships between them, and contextual examples illustrating these relationships. This injection method enables the model to more precisely identify entity relationships within the task text. Furthermore, as new tasks and data are introduced, the knowledge base can be dynamically updated through incremental learning, ensuring the timeliness and accuracy of background knowledge and allowing the model to continuously adapt to the latest developments in the cryptography domain.

Table 1: Example of Entity-Relation Candidate Knowledge

No	Content
1	([System], [Affiliated with], [Unit], [Cryptography xx system is affiliated with xx management bureau])
2	([System], [Deployed in], [Server Room], [Electronic signature system is deployed in the financial institution's server room])
3	([System], [Includes], [Cryptographic Device], [The system includes HSM encryption devices])
4	([System], [Calls], [Cryptographic Algorithm], [The system calls the SM4 algorithm])
5	([System], [Implements], [Cryptographic Protocol], [The system implements the IPSec protocol])
6	([System], [Uses], [Communication Channel], [The system uses the HTTPS channel to transmit encrypted data])
7	([Server Room], [Stores], [Cryptographic Device], [The xx server room stores quantum key distribution devices])
8	([Cryptographic Device], [Operates in], [Server Room], [The server encryption device operates in the bank's encryption zone])
9	([Cryptographic Device], [Executes], [Cryptographic Algorithm], [The encryption device executes the RSA-2048 algorithm])
10	([Cryptographic Device], [Supports], [Cryptographic Protocol], [The gateway supports the TLS 1.3

	protocol])
11	([Cryptographic Device], [Stores], [Cryptographic Certificate], [The mobile payment terminal contains the root CA certificate])
12	([Cryptographic Device], [Uses], [Communication Channel], [The encryption gateway transmits data via a VPN channel])
13	([Cryptographic Protocol], [Uses], [Cryptographic Algorithm], [The ECDH protocol uses elliptic curve algorithms])
14	([Cryptographic Protocol], [Protects], [Communication Channel], [The Kerberos protocol protects the secure channel for ticket transmission])
15	([Cryptographic Protocol], [Validates], [Cryptographic Certificate], [The TLS protocol validates the validity of X.509 certificates])
16	([Communication Channel], [Supports], [System], [SD-WAN channel supports distributed enterprise system communication])
17	([Cryptographic Certificate], [Binds], [Cryptographic Device], [The digital ID certificate is bound to a specific UKey hardware])
18	([Cryptographic Certificate], [Authenticates System], [System], [SSL server certificates authenticate the Web system's identity])
19	([Cryptographic Certificate], [Authenticates Unit], [Unit], [SSL certificates display the enterprise's legal person registration information])
20	([Unit], [Manages], [System], [The xx management bureau manages the cryptographic management system])
21	([Unit], [Manages], [Server Room], [The xx bank manages the financial data center])
22	([Unit], [Manages], [Cryptographic Device], [The unit centrally manages all encryption gateway devices])
23	([Unit], [Issues], [Cryptographic Certificate], [The CA can issue certificates])
24	([System], [Implements], [Cryptographic Protocol], [The system implements the IPSec protocol])

Specifically, a set of quadruples is constructed in the format (Entity 1, Relationship, Entity 2, Contextual Example), encompassing all required entity relationships. The task input, which includes the target text, its entities, and their types, is used to retrieve relevant quadruples from this set based on entity types. A subset of candidate quadruples related to the input is then filtered and injected into the CoT prompt template, addressing challenges such as incomplete entity information and contextual understanding.

Compared to static approaches that inject all knowledge at once, which may lead to information overload, making it difficult for the model to extract key relationships and introducing irrelevant noise that affects final judgments, the method in this module filters the most relevant relationship information based on the entities present in the task text. This reduces interference from irrelevant noisy knowledge and minimizes confusion, thereby enhancing the model's extraction precision.

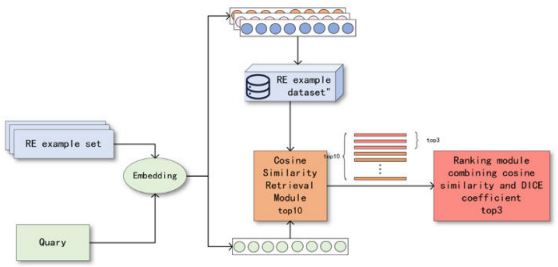
C. Dependency Relationship Extraction Module

To address these issues and more accurately identify complex entity relationships in the cryptography domain, dependency analysis technology is introduced. This technique aims to assist the model in deeply understanding the hierarchical structure of sentences and clarifying the relationships between words, thereby enhancing its ability to recognize entity associations. Dependency analysis, also known as dependency parsing or dependency structure analysis, is a key technique in natural language processing. It focuses on analyzing the dependency relationships between words in a sentence, using a tree-like structure to describe these dependencies. The introduction of this method provides syntactic information support for relation extraction in the cryptography domain, improving the model's accuracy and robustness when handling complex, specialized data. To this end, a dependency relationship extraction module is designed to extract dependency relationships from the target task text, aiding the model in parsing syntactic structures more accurately, understanding the text more clearly, and uncovering relationships between entities. In this chapter, dependency relationships in sentences are analyzed and extracted using StanfordCoreNLP, developed by Stanford University.

D. Example Retrieval Module for Relation Extraction Tasks in the Cryptography Domain

Large-scale models possess strong in-context learning (ICL) capabilities, and the injection of high-quality examples can enable these models to perform well on tasks even in low-resource scenarios. However, injecting low-quality examples can degrade model performance. Traditional methods that statically inject fixed examples are ineffective for relation extraction tasks in the cryptography domain. This is because fixed samples struggle to capture the diversity and complexity of relationships, offering limited knowledge coverage in low-resource environments. Moreover, they lack task adaptability, exhibit low relevance to the task text, and are prone to introducing noise interference.

To address this, this chapter proposes an example retrieval method tailored for relation extraction tasks in the cryptography domain. The overall structure and workflow of this method are illustrated in Figure 3. The core objective of this approach is to dynamically retrieve the most relevant examples from a pre-constructed cryptography-domain relation extraction example library for each task input. These examples are then used to construct few-shot prompts, guiding the large language model in completing the relation extraction task. The example library is based on the cryptography-domain relation dataset pre-constructed in this chapter, containing 1,536 annotated samples covering 24 entity relationship categories (carefully selected as the most representative data from a total of 2,292 entries). The retrieval process employs a weighted ranking strategy that combines cosine similarity and the DICE coefficient. By comprehensively evaluating semantic similarity and entity overlap, this strategy ensures that the retrieved examples are not only semantically close to the input text but also preserve key terminology and entity characteristics specific to the cryptography domain. Ultimately, the top-3 retrieved



examples are injected into the prompt template, enhancing the model's relation extraction capabilities.

Figure 3: Example Retrieval Method

First, the construction of the example library serves as the foundation for example retrieval, providing high-quality reference samples for relation extraction tasks. The example library is built based on the cryptography-domain relation extraction dataset, containing carefully filtered and annotated text samples that cover various entity relationship patterns. To transform the text into semantic representations, this method employs the S-BERT model to generate embedding vectors. Compared to the original BERT, S-BERT optimizes sentence-level embeddings, efficiently capturing semantic similarity and producing fixed-size high-dimensional vectors. These vectors exhibit greater distinctiveness in semantic space, clustering semantically similar sentences together in the vector space, which facilitates subsequent retrieval and ranking. This makes S-BERT particularly suitable for complex, low-resource scenarios like the cryptography domain. The generated embedding vectors are stored in the ChromaDB vector database, enabling efficient similarity computation and retrieval in subsequent steps. To maintain the timeliness and adaptability of the example library, its contents are periodically updated with new data and task requirements through incremental learning, reflecting the latest developments in the domain and ensuring structured, efficient support for relation extraction tasks.

Next, the example retrieval module comes into play. When an input text requiring relation extraction is received, it is first transformed into a high-dimensional vector using S-BERT. Similarity calculations are then performed between this vector and the example vectors stored in ChromaDB. Traditional methods such as cosine similarity, Euclidean distance, and Manhattan distance have limitations: cosine similarity relies solely on the angle between vectors, overlooking contextual differences and being insensitive to redundancy in long texts, while Euclidean and Manhattan distances are sensitive to vector magnitude and neglect directionality, performing poorly in high-dimensional spaces. To address the shortcomings of these traditional similarity computation methods in the context of cryptography-domain relation extraction tasks, this chapter adopts a text similarity computation approach that combines cosine similarity and the DICE coefficient. This ensures that example retrieval considers not only semantic similarity but also matches the distinctive entity relationship features specific to the cryptography domain.

Specifically, the process begins by vectorizing the query text $q$and the example text d_i,converting them into high-dimensional vectors $\mathbf{v}_q$ and $\mathbf{v}_{d_i}$,respectively. Then, the cosine similarity $\mathrm{Cosine}(q, d_i)$.between the two is calculated according to Equation (1). All examples are ranked in descending order based on cosine similarity, and the top 10 examples with the highest similarity scores are selected as a coarse recall set. This ensures that the initially filtered examples are semantically close to the current task text.

$$\mathrm{Cosine}(q,d_i)=\frac{\mathbf{v}_q\cdot\mathbf{v}_{d_i}}{\|\mathbf{v}_q\|\|\mathbf{v}_{d_i}\|}\#(1)$$

Since the range of cosine similarity is [-1,1], to combine it with the DICE coefficient score for an overall similarity calculation, it is mapped to the interval [0,1].according to Equation (2). This yields the cosine similarity score$S_{cosine}(d_i)$。

$$S_{cosine}(d_i)=\frac{\mathrm{Cosine}(q,d_i)+1}{2}\#(2)$$

Next, for the task input text and the filtered examples, the sets of entity types they contain, E_q and E_{d_i}, are extracted respectively. Then, according to Equation (3),the DICE coefficient $DICE(E_q,E_{d_i})$ is calculated to obtain the DICE score $S_{DICE}(d_i)$ · This score measures the similarity between the entity types contained in the task input and the candidate examples, ensuring that the entity types in the examples align with those in the current task.

$$S_{DICE}(d_i)=DICE(E_q,E_{d_i})=\frac{2\times|E_q\cap E_{d_i}|}{|E_q|+|E_{d_i}|}\#(3)$$

Then, to integrate the information from both cosine similarity and the DICE coefficient, the scores from these two similarity calculations are combined according to Equation (4) to compute the comprehensive ranking score$S_{\mathrm{total}}(d_i)$ for each example.

$$S_{\mathrm{total}}(d_i)=S_{\mathrm{cosine}}(d_i)+S_{\mathrm{DICE}}(d_i)\#(4)$$

Finally, the Top-10 examples recalled in the initial screening are sorted according to $S_{\mathrm{total}}(d_i)$,and the top 3 examples with the highest rankings are selected as the final retrieval results.

By using the above method, it is possible to dynamically and flexibly retrieve the most suitable examples from the example library and inject them into the chain-of-thought prompt templates for the cryptography-related relation extraction task. This forms a complete input with high contextual relevance, reducing the negative impact of noise, allowing the model to fully leverage its ICL capabilities, and thereby improving the performance of the relation extraction task.

III. Experiments

If you are using *Word,* use either the Microsoft Equation Editor or the *Math Type* add-on (http://www.mathtype.com) for equations in your paper (Insert | Object | Create New | Microsoft Equation *or* Math Type Equation). "Float over text" should *not* be selected.

A. Dataset

This paper focuses on the characteristics of data and tasks in cryptography-related relation extraction. Using real-world business data from the cryptography field, books, and teaching materials as data sources, and through discussions with cryptography experts, 24 types of entity relationships were identified. Entity relationships were

annotated using a multi-person, multi-round cross-annotation process and expert proofreading. A cryptography-related relation extraction dataset was constructed, covering 24 types of relationships and containing 2,292 high-quality data entries. The dataset was split into training, validation, and test sets at a ratio of 8:1:1 for model training, parameter tuning, and performance evaluation.

B. Evaluation Metrics

Precision：Precision measures the proportion of actual positive samples among all samples predicted as positive.

$$precision = \frac{Num_{TP}}{Num_{TP} + Num_{FP}} \#(5)$$

Recall：Recall measures the proportion of correctly predicted positive samples among all actual positive samples.

$$recall = \frac{Num_{TP}}{Num_{TP} + Num_{FN}} \#(6)$$

F1-score：The F1 score is the harmonic mean of precision and recall.

$$F1-score = \frac{precision \times recall \times 2}{precision + recall} \#(7)$$

C. Experimental Environment

The experimental environment consists of 120GB of memory, a GPU with NVIDIA GeForce RTX 4090, a CPU of Xeon(R) Gold 6430, 24GB of VRAM, CUDA 12.4, and the runtime environment is Python 3.10. The parameters are shown in Table 3.

Table 3: Experimental Parameter Settings

Parameters	value
Learning Rate	1e-5
Epoch	5
Batchsize	16
LoRA-rank	16
LoRA-alpha	32

D. Comparison Experiments

To comprehensively evaluate the performance of the cryptography-related relation extraction method based on retrieval augmentation and chain-of-thought, three types of models were selected for comparison: traditional neural network models, deep learning models, and large-scale pre-trained language models. These models represent different technical approaches to relation extraction tasks and enable the assessment of the effectiveness of the proposed method from multiple dimensions.

The traditional neural network method was a mainstream technology in early relation extraction tasks, relying on sequence modeling or local feature extraction. In this study, CNN and BiGRU were selected as baseline models to compare the performance differences between traditional methods and the proposed approach in the cryptography-related relation extraction task.

CNN: CNN extracts local contextual features through convolution kernels and excels at capturing patterns in short texts. However, due to its reliance on fixed window sizes, CNN struggles to handle long-distance dependencies, which limits its performance in relation extraction tasks.

BiGRU: BiGRU, using a bidirectional GRU structure, can simultaneously model the forward and backward dependencies of a text, offering advantages in contextual understanding compared to CNN. However, due to limitations in sequence modeling, BiGRU still falls short in modeling long texts and cross-sentence relationships.

The selection of CNN and BiGRU aims to reveal the limitations of traditional neural networks in highly specialized, low-resource scenarios, providing a benchmark for future method improvements.

With the development of deep learning, end-to-end relation classification models have become key technologies in natural language processing. These methods combine domain knowledge with neural networks to automatically extract deep features, offering advantages over traditional models. This study uses R-BERT and DRPC (Dependency Relation Prediction and Control) as representative models.

R-BERT: Based on the BERT pre-trained model, R-BERT enhances the semantic representation ability of relation classification by introducing entity markers. R-BERT integrates entity boundary information into the extraction process, excelling at handling explicit relationships in short sentences. However, it performs limitedly in modeling long-distance or implicit relationships.

DRPC: DRPC employs a multi-task learning framework to simultaneously perform relation extraction and dependency relation prediction, improving the model's understanding of text structure. It introduces a self-attention mechanism to enhance its ability to capture long-distance dependencies. Additionally, an information flow control mechanism allows adaptive focusing on key information, reducing irrelevant noise and improving relation extraction accuracy.

The selection of R-BERT and DRPC aims to evaluate the performance of deep neural network methods in low-resource cryptography-related tasks and to verify whether the proposed optimization strategy can further improve performance.

Large-scale pre-trained language models, with their outstanding contextual understanding abilities, have performed exceptionally well in NLP tasks, especially in relation extraction, where they have demonstrated groundbreaking progress. This study selects the ChatGLM, Qwen, and ChatGPT series models to assess their adaptability in cryptography-related tasks and explore the optimization effects of the proposed method on large language models.

ChatGLM series: Developed by Zhipu Huazhang, the ChatGLM series is optimized for Chinese NLP tasks. The Air version is lightweight, suitable for simpler tasks, while the Plus version has stronger reasoning capabilities, making it suitable for high-precision requirements. It is ideal for evaluating the performance of LLMs in Chinese relation extraction tasks.

Qwen series: Developed by Alibaba, the Qwen series performs excellently on Chinese tasks. The Plus version balances efficiency and performance, while the Max version

has stronger reasoning capabilities, allowing for comparisons of different LLMs' performance on specialized tasks.
ChatGPT-4.0: Developed by OpenAI, ChatGPT-4.0 is one of the most advanced general-purpose large models, with powerful reasoning capabilities. It serves as a baseline for evaluating the performance of large language models.
The selection of these models aims to analyze the baseline performance of LLMs in cryptography-related tasks and verify whether the proposed method can further enhance the accuracy and robustness of relation extraction on top of LLMs.

The results of the comparison experiments are shown in Table 4, with all data rounded to two decimal places.

Table 4: Comparison of Experimental Results

Models	P	R	F1-score
CNN	24.73	27.28	25.94
BiGRU	32.43	35.26	33.79
R-BERT	38.54	42.71	40.52
DRPC	40.18	45.24	42.56
ChatGLM-Air	41.32	47.64	44.26
ChatGLM-Air+our method	48.14	55.87	51.72
Qwen-Plus+our method	52.70	55.21	53.93
ChatGPT-4.0+our method	63.84	66.97	65.37
ChatGLM-0520+our method	64.34	67.52	65.89
Qwen-Max+our method	65.87	68.97	67.38
ChatGLM-Plus+our method	**68.76**	**72.58**	**70.62**

The performance of the traditional neural network models CNN and BiGRU was poor, with F1-scores of 25.94% and 33.79%, respectively. These results were significantly lower than those of the other two types of models. CNN relies on local feature extraction and struggles to capture the complex long-distance dependencies in cryptography-related texts, leading to the lowest F1-score. The bidirectional gated recurrent unit (BiGRU) captures both forward and backward dependencies in the context, offering advantages over CNN in understanding the domain-specific data. The F1-score for BiGRU improved by 7.85 percentage points compared to CNN, but it was still limited by the inability of traditional neural networks to generalize in specialized, low-resource scenarios. This demonstrates that traditional neural network models lack deep semantic understanding and the ability to handle complex relationships, leading to poor performance in cryptography-related relation extraction tasks and difficulty in meeting high-accuracy demands.

The deep learning models R-BERT and DRPC showed a significant improvement in F1-score compared to the traditional neural networks, with scores 6.73 and 8.77 percentage points higher than BiGRU, respectively. R-BERT, which uses the BERT pre-trained model and fine-tunes it on the training set constructed in this study, adapted to cryptography-related relation extraction tasks. R-BERT achieved an F1-score of 40.52% on this task, showing significant performance improvement compared to CNN and BiGRU. This improvement is mainly due to the strong text understanding ability of the BERT pre-trained model, which enables it to capture richer contextual information and, to some extent, recognize entity relationships in the cryptography domain. However, despite the fine-tuning with cryptographic data, the small size of the training set led to insufficient generalization ability in low-resource environments. Additionally, R-BERT struggles with complex tasks involving multiple relationships and long-distance dependencies. Furthermore, since the BERT pre-training knowledge is derived from general texts and not specifically optimized for cryptography, it performs poorly when dealing with specialized terminology and implicit relationships. The DRPC model also trained on the same dataset achieved an F1-score of 42.56%, 2.04 percentage points higher than R-BERT. This improvement is attributed to DRPC's use of syntactic structure information to enhance its performance in relation extraction tasks, and the introduction of a self-attention mechanism helps better capture the complex relationships in cryptography. Overall, deep learning models outperform traditional neural networks in this task, but due to the limitations in training data and the inherent shortcomings of the models, their performance still cannot compete with that of optimized large models.

In this experiment, ChatGLM4-Air was selected as the base model and directly applied to cryptography-related relation extraction tasks. A comparison was also made with the same model optimized using the proposed method to verify its impact on model performance. As shown in Figure 4, ChatGLM4-Air, without optimization, achieved an F1-score of 44.26% on the cryptography-related relation extraction task, which is better than the performance of traditional deep learning models. This is mainly due to the powerful reasoning capabilities and rich pre-trained knowledge of large models. When optimized using the proposed method, ChatGLM4-Air's F1-score increased to 51.72%, a 7.46 percentage point improvement over the original model. This improvement is attributed to four main factors: the chain-of-thought prompt designed for cryptography-related relation extraction tasks significantly improved the model's logical reasoning ability and reduced the chances of reasoning errors; the precise injection of alternative knowledge about entity relationships provided sufficient knowledge to the model

while significantly reducing the negative impact of noise; dependency relations optimized the model's ability to extract complex relationships across sentences and long distances; and the example retrieval strategy effectively improved the model's relation extraction ability in low-resource environments. This set of comparison experiments strongly demonstrates the effectiveness of the proposed method in enhancing large models' ability to perform cryptography-related relation extraction tasks. It proves that the proposed method significantly improves large models' performance in cryptography-related relation extraction tasks, enabling them to achieve high accuracy even in the absence of large-scale labeled data, providing an effective optimization solution for the application of large models in cryptography-related relation extraction tasks.

The experiment applied the proposed method to optimize six different large language models. The results showed that all large models achieved significant improvements after using the method, with the optimized ChatGLM-Plus achieving the best performance, followed by the optimized Qwen-Max and ChatGLM-0520.

The optimized ChatGLM-Plus performed best in the experiment, achieving an F1-score of 70.62%, fully demonstrating the outstanding effect of the proposed method on cryptography-related relation extraction tasks. The optimized Qwen-Max achieved an F1-score of 67.38%, showing excellent performance, but slightly behind ChatGLM-Plus by 3.24 percentage points. Meanwhile, the optimized ChatGLM-0520 and ChatGPT-4.0 also achieved impressive scores of 65.89% and 65.37%, respectively. However, in comparison, the optimized Qwen-Plus and ChatGLM4-Air had the lowest F1-scores among the large models, with scores of only 53.93 and 51.72. These results suggest that while the proposed method can enhance the performance of large models in cryptography-related relation extraction tasks, the final outcome is still limited by the capabilities of the base models.

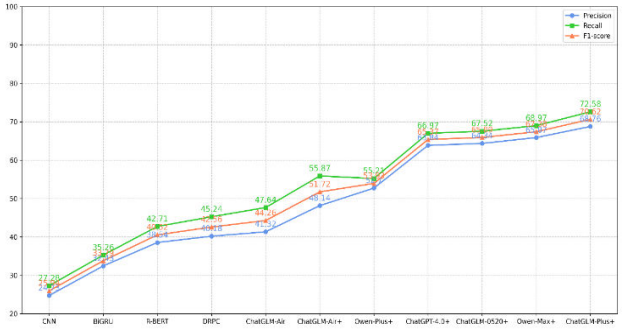


Figure 4 Comparison of Relation Extraction Experiments

The above experimental results show that the proposed method can significantly enhance the relation extraction capabilities of various large models in the cryptography domain, even under conditions of scarce labeled data. It provides an effective optimization strategy for large models in cryptography and other highly specialized, complex domains, offering significant academic and practical value.

Ablation Experiment

The ablation experiment aims to evaluate the specific contribution of each module to the performance of cryptography-related relation extraction tasks by progressively removing or adding the optimization components proposed in this study. This helps to verify the effectiveness and necessity of each component. The cryptography-related relation extraction method based on retrieval enhancement and chain-of-thought proposed in this chapter includes several key optimization modules, such as the chain-of-thought (CoT) prompt template for cryptography tasks, precise injection of entity relationship alternatives (PIJ), dependency relation enhancement (DP), and optimal example injection (OEJ). These modules work together to enhance model performance in tasks involving complex terminology, implicit relationships, and data scarcity scenarios. However, the specific role of each module and its impact on overall performance still requires further quantitative analysis.

The ablation experiment adjusts the method configuration under unified experimental conditions to observe changes in model performance, revealing each module's independent contribution and synergy in improving precision, recall, and F1-score. The experiment uses ChatGLM4-Plus as the base model and tests it on the cryptography-related relation extraction dataset constructed in Chapter 4 to ensure the reliability and domain relevance of the results.

To comprehensively assess the role of each module, the ablation experiment designed the following four configurations, progressively adding optimization components:

LLM+CoT: This is the baseline configuration for the ablation experiment. It enhances the ChatGLM4-Plus large model with the chain-of-thought (CoT) prompt template for cryptography-related relation extraction tasks. Additionally, all the required alternative entity relations are statically written into the prompt template, along with three static fixed examples. This configuration aims to test the base performance of the large model with only the CoT optimization, reflecting its initial ability to handle complex relationship reasoning.

LLM+CoT+PIJ: This configuration builds upon the baseline by adding the entity relationship alternative knowledge precise injection (PIJ) module. This module dynamically selects the most relevant knowledge from a pre-built entity relationship quadruple knowledge base and injects it into the prompt template to reduce noise interference and enhance the model's understanding of domain-specific relationships. This configuration is used to evaluate the role of the PIJ module in improving the model's performance.

LLM+CoT+PIJ+DP: This configuration further introduces the dependency relation enhancement (DP) module. The DP module extracts dependency relations between entities through syntactic parsing and injects them into the prompt template. This configuration tests the contribution of the DP module.

LLM+CoT+PIJ+DP+OEJ: This is the complete method proposed in this chapter. It adds the optimal example injection (OEJ) module to the previous configuration. This module dynamically retrieves Few-shot examples from a pre-built example library using a combination of cosine similarity and the DICE coefficient method, selecting examples that best match the task text and injecting them into the prompt template to improve model performance. This

configuration reflects the best performance achieved through the synergistic effect of all modules.

The results of the ablation experiment are shown in Table 5, with all data rounded to two decimal places.

Table 5 Ablation Experiment Results

Configuration	P	R	F1-score
LLM+CoT	58.26	65.51	61.67
LLM+CoT+PIJ	66.31	67.32	66.81
LLM+CoT+PIJ+DP	66.75	69.93	68.30
LLM+CoT+PIJ+DP+OEJ	**68.76**	**72.58**	**70.62**

LLM+CoT: This configuration uses CoT enhancement and statically writes all alternative entity relationships, as well as three manually selected fixed examples, into the prompt template. The model achieved an F1-score of 61.67%. This result indicates that the chain-of-thought (CoT) prompt template for cryptography-related relation extraction tasks significantly enhanced the model's reasoning ability through structured inference and self-check mechanisms. The recall rate of 65.51% suggests that the model is able to cover relationship instances well, but the precision is relatively low at only 58.26%, indicating that the static injection of all entity relationships introduced a lot of noise, leading to a higher false-positive rate.

LLM+CoT+PIJ: After adding the entity relationship alternative knowledge precise injection (PIJ) module, the F1-score increased to 66.81%, a 5.17 percentage point improvement. Unlike the previous configuration, the addition of PIJ significantly reduced noise interference, and the substantial improvement in precision indicates that the module effectively reduced misjudgments of relationships. This result validates the critical role of PIJ in enhancing model performance.

LLM+CoT+PIJ+DP: After further introducing the dependency relation enhancement (DP) module, the F1-score reached 68.30%, a 1.49 percentage point increase over the previous configuration. The recall rate improved by 2.61 percentage points, indicating that the DP module significantly enhanced the model's ability to handle complex relationships, particularly implicit relationships and long-distance dependencies common in cryptography-related texts. However, the improvement in precision was limited, which shows that while DP contributed significantly to improving recall, further improvements in precision rely on other modules.

LLM+CoT+PIJ+DP+OEJ: In the full configuration, the F1-score reached 70.62%, an increase of 2.32 percentage points over the previous configuration. Compared to the previous configuration, OEJ replaced the three static fixed examples in the prompt template with the best examples dynamically retrieved based on task input. This improvement ensures that the injected examples are highly relevant to the current task by using cosine similarity and the DICE coefficient for ranking. The simultaneous improvement in both precision and recall indicates that this module not only optimized the model's precise judgment of relationships but also enhanced its ability to cover a wide range of complex relationships, achieving the best overall performance in the experiment. This result highlights the important role of this module in improving the model's task performance.

Comprehensive Analysis: As shown in Figure 5, each module contributed to the optimization of the model's performance. Additionally, the synergy of these components led to excellent results in the cryptography-related relation extraction task, validating the rationality and practical value of the proposed method.

IV. SUMMARY

This paper addresses the challenges in cryptography-related relation extraction tasks, such as complex terminology, implicit relationships, data scarcity, and unstable reasoning. It proposes a cryptography-related relation extraction method based on retrieval enhancement and chain-of-thought , which enhances performance in low-resource environments. The method combines CoT prompt optimization, Retrieval-Augmented Generation (RAG), precise injection of entity relationship alternatives, and dependency relation enhancement technologies, significantly improving extraction accuracy and robustness. Additionally, a cryptography-related relation extraction dataset is constructed, providing strong support for related work in the cryptography domain.

First, in terms of dataset construction, the characteristics of data and tasks in cryptography-related relation extraction were considered. Real-world cryptography business data, books, and teaching materials were used as data sources. Through discussions with experts, 24 types of entity relationships in the cryptography domain were identified. Using multi-person, multi-round cross-annotation and review proofreading, a high-quality cryptography-related relation extraction dataset was constructed, laying the foundation for subsequent experiments.

Next, the chain-of-thought prompt template was designed, which breaks down relation extraction into logical steps and introduces a self-check mechanism, improving the model's ability to analyze complex relationships in specialized domains. Then, the optimal example retrieval module used cosine similarity and DICE coefficient-based weighted ranking to dynamically inject the most relevant Few-shot examples into the prompt template, mitigating issues related to data scarcity and suboptimal extraction performance. The precise entity relationship injection module pre-built an alternative knowledge base of entity relationships, dynamically selecting the most relevant relationships to inject into the prompt template, significantly reducing noise

interference. The dependency relation enhancement module extracted structured information via syntactic parsing and injected it into the template, optimizing the model's ability to understand complex cryptography-related texts.

Finally, experiments showed that the method, based on the ChatGLM4-Plus model, achieved an F1-score of 70.62%, outperforming both general large models and traditional baseline methods. Ablation experiments validated the contribution of each module, and these modules, working synergistically, improved the method's performance in cryptography-related relation extraction tasks.

REFERENCES

[1] Nayak T, Majumder N, Goyal P, et al. Deep neural approaches to relation triplets extraction: a comprehensive survey[J]. Cognitive computation, 2021, 13(5): 1215-1232.

[2] XU W, DENG Y, LEI W, et al. ConReader: Exploring implicit relations in contracts for contract clause extraction[C]. Association for Computational Linguistics, 2022.

[3] Chen X, Zhang N, Li L, et al. Hybrid transformer with multi-level fusion for multimodal knowledge graph completion[C]//Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval. 2022: 904-915.

[4] Zhang R, Trisedya B D, Li M, et al. A benchmark and comprehensive survey on knowledge graph entity alignment via representation learning[J]. The VLDB Journal, 2022, 31(5): 1143-1168.

[5] Luo K, Lin F, Luo X, et al. Knowledge base question answering via encoding of complex query graphs[C]//Proceedings of the 2018 conference on empirical methods in natural language processing. 2018: 2185-2194.

[6] Yang Z. Biomedical information retrieval incorporating knowledge graph for explainable precision medicine[C]//Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval. 2020: 2486-2486.

[7] Miwa M, Bansal M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, 2016.

[8] Nayak T, Ng H T. Effective modeling of encoder-decoder architecture for joint entity and relation extraction[C]//Proceedings of the AAAI conference on artificial intelligence. 2020, 34(05): 8528-8535.Y. Yuan, X. Zhou, S. Pan, et al. A relation-specific attention network for joint entity and relation extraction. In Proceedings of the International Joint Conference on Artificial Intelligence. International Joint Conference on Artificial Intelligence. 2021.

[9] Zhao T, Yan Z, Cao Y, et al. Asking effective and diverse questions: A machine reading comprehension based framework for joint entity-relation extraction[C]//Proceedings of the Twenty-Ninth International Conference on International Joint Conferences on Artificial Intelligence. 2021: 3948-3954.

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

[11] Miwa M, Bansal M. End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures[C]//Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). 2016: 1105-1116.

[12] Pawar S, Palshikar G K, Bhattacharyya P. Relation extraction: A survey[J]. arXiv preprint arXiv:1712.05191, 2017.

[13] Bhartiya A, Badola K. DiS-ReX: A Multilingual Dataset for Distantly Supervised Relation Extraction[C]//Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). 2022: 849-863.

[14] Chen X, Zhang N, Xie X, et al. KnowPrompt: Knowledge-aware Prompt-tuning with Synergistic Optimization for Relation Extraction[C]//Proceedings of the ACM Web Conference 2022. ACM, 2022: 2778-2788.

[15] Wu S, Fan K, Zhang Q. Improving distantly supervised relation extraction with neural noise converter and conditional optimal selector[C]//Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence. 2019: 7273-7280.

[16] Yan Z, Jia Z, Tu K. An empirical study of pipeline vs. joint approaches to entity and relation extraction[C]//Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers). 2022: 437-443.

[17] Zhao X, Yang M, Qu Q, et al. Exploring Privileged Features for Relation Extraction With Contrastive Student-Teacher Learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(8): 7953-7965.

[18] Zeng D, Liu Q, Zhang H. CopyMTL: Copy Mechanism for Joint Extraction of Entities and Relations with Multi-Task Learning[C]//Proceedings of the AAAI Conference on Artificial Intelligence. Association for the Advancement of Artificial Intelligence (AAAI), 2020, 34: 9507-9514.

[19] Li X, Yin F, Sun Z, et al. Entity-Relation Extraction as Multi-Turn Question Answering[C]//Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. 2019: 1340-1350.

[20] Hillebrand L P, Deußer T, Dilmaghani T, et al. KPI-BERT: A Joint Named Entity Recognition and Relation Extraction Model for Financial Reports[C]//International Conference on Pattern Recognition 2022. 2022.

[21] Yuan Y, Zhou X, Pan S, et al. A relation-specific attention network for joint entity and relation extraction[C]//International Joint Conference on Artificial Intelligence-Pacific Rim International Conference on Artificial Intelligence 2020. Association for the Advancement of Artificial Intelligence (AAAI), 2020: 4054-4060.

[22] Wei J, Wang X, Schuurmans D, et al. Chain-of-thought prompting elicits reasoning in large language models[J]. Advances in neural information processing systems, 2022, 35: 24824-24837.

[23] Lewis P, Perez E, Piktus A, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks[J]. Advances in neural information processing systems, 2020, 33: 9459-9474.