

An Adaptive RAG Knowledge Retrieval Method for Large Models Based on Domain Knowledge Enhancement

Hai-Yun Ma^{1*}, Ying-Long Ma², and Zhong-Lin Zhang³

¹ School of Electronic Information and Electrical Engineering, Tianshui Normal University,
Tianshui, 741001, China
tsmhy1999@163.com

² School of Control and Computer Engineering, North China Electric Power University,
Beijing 102206, China
yinglongma@ncepu.edu.cn

³ School of Electronic and Information Engineering, Lanzhou Jiaotong University,
Lanzhou 730070, China
zhangz1@mail.lzjtu.cn

Received 4 March 2025; Revised 24 July 2025; Accepted 4 August 2025

Abstract. Knowledge Q&A, a key application of LLMs, draws much attention. However, LLMs struggle with Chinese abbreviated technical terms due to insufficient domain knowledge, and existing power Q&A methods often neglect text semantic context, limiting performance. This paper proposes an LLM-adaptive RAG method with domain knowledge enhancement for Chinese power Q&A. It first presents an info extraction framework with domain term recognition enhancement, using dual-pointer and generative extraction. Second, a two-way recall RAG method is proposed, predicting entity/relationship links via vectors and keywords, generating canonical forms, and using LLMs to select answers. Experiments show it outperforms mainstream methods in power field and works well generally.

Keywords: knowledge question and answering (Q&A), domain term recognition enhancement, information extraction, prompt, retrieval-augmented generation (RAG), large language model (LLM)

1 Introduction

Knowledge question and answering (Q&A) [1] involves modeling a knowledge graph as a graph, where the user's question is treated as a search path and the answer is derived using graph structure information, such as the shortest path or graph traversal. The power industry [2], a backbone of modern social infrastructure, encompasses aspects such as power grid topologies, power equipment operation statuses, and new energy technologies, forming a vast and complex system. To effectively manage and utilize this system, the power knowledge graph [3] has emerged, structurally linking knowledge within the power domain and serving as a foundation for intelligent power grid management and decision-making. When dealing with power grid accident handling plans, establishing an efficient and fast power knowledge Q&A system can effectively enhance operational efficiency. Moreover, retrieval augmentation can retrieve information from existing databases to enhance the generation effect. Combining knowledge Q&A with retrieval augmentation is therefore pivotal for advancing power grid intelligence.

In recent years, methods based on deep learning models [4-6] have been widely adopted in knowledge Q&A, achieving significant research progress. Mainstream knowledge Q&A methods can be divided into two categories, based on semantic parsing [4] and information retrieval [5]. Semantic-parsing-based methods parse the question, extract entities and relationships using natural language processing techniques [7], and generate logical expressions [4] that can be executed on the knowledge graph. Long Short-Term Memory (LSTM) [7] or Convolutional Neural Networks (CNNs) [8] can be utilized to encode the semantics of the mined question,

* Corresponding Author

while attention mechanisms [9] can be introduced to more accurately identify the entities and relationships most relevant to the question. Information-retrieval-based methods are typically divided into two stages: indexing and ranking. These methods efficiently filter out answer information highly relevant to the question from the extensive structured data in the knowledge graph. The Multi-Channel Convolution Neural Network (MCCNN) [5] is utilized to obtain three feature types: the question–answer path, type, and context. It learns the embedded vector representation of candidate answers, which is used as the input for a scoring function to rank the candidate answers. Recent advancements in knowledge Q&A focus on researching the retrieval-augmented generation (RAG) framework [10], namely combining Q&A methods with large language models (LLMs) to enhance Q&A performance. For example, by employing LLMs to optimize the index [11] or directly utilizing LLMs for retrieval [12].

Although RAG-based knowledge Q&A methods [11, 12] can improve Q&A performance to a certain extent, many of these methods face challenges in recognizing specialized terms in professional fields, and most existing methods rely solely on embedded semantic vectors to mine answers [13]. Overemphasizing embedded semantic vectors while ignoring the text itself can lead to severe error propagation. Knowledge Q&A based on deep learning models typically learns domain knowledge by embedding external knowledge [14, 15]. While a few LLM-based knowledge Q&A methods [11, 12] can enhance domain-specific performance by injecting external knowledge, the fact that many advanced LLMs are not open source [16] makes it very challenging to inject external knowledge into these models.

The key to the power knowledge Q&A task lies in bridging the gap between natural language and the power domain knowledge graph’s structure, providing a user-friendly natural language interface and enabling the power domain knowledge graph’s convenient operation. Currently, to achieve this goal, researchers primarily adopt methods based on semantic parsing and information retrieval. The core idea of semantic-parsing-based methods is to deepen the understanding of the question asked by the user, enabling the mining of implicit semantic and syntactic information in the user’s question. A deeper understanding can provide more accurate information. This aids the model’s subsequent retrieval work on the knowledge graph.

Traditional semantic parsing methods typically use manual annotation to generate the corresponding logical form. Berant et al. [18] roughly mapped question phrases to entities and relationships in the knowledge base, enhancing performance through a log-linear model and generating additional predicates through bridge connection operations based on adjacent predicates. Bast et al. [19] proposed the *Aqqu* model, which maps the question to three templates: entity, relationship, and answer node. The model then matches the question with the knowledge base and instantiates the template. Zheng et al. [20] started with natural language questions and the knowledge base, proposing a method for automatically generating templates based on the similarity between the two. Abujabal et al. [21] proposed the *QUINT* model, which can automatically generate templates. When running the task, the question template is obtained through question mapping, dependency relationships in the parsed question are instantiated, and the results are obtained through scoring and ranking. Abujabal et al. [22] proposed the *NEQA* model, which completes knowledge base question answering (KBQA) through continuous learning. This model can automatically learn mapping templates from a small number of datasets to complete the mapping from a syntactic structure to a semantic structure. Additionally, it can continuously improve through user feedback. Although these methods provide strong interpretability, the manual workload is too significant.

With the continuous development of deep learning technology, knowledge Q&A methods based on deep learning are becoming increasingly prevalent. Compared to traditional methods, the performance of deep-learning-based methods has significantly improved. Golub et al. [4] selected the top ten candidate entities as the entity set, extracting its one-hop relationships in the knowledge graph as the candidate relationships. The corresponding vector encodings were obtained through LSTM and a CNN. Then, the cosine similarity between the entity vectors and relationship vectors was calculated to obtain the final result. Yavuz et al. [23] represented the context of the entity through Bidirectional Long Short-Term Memory (BiLSTM), with this context also helping the model accurately predict the entity type. Then, the entities were ranked by calculating the similarity to obtain the final result. Dong et al. [24] rewrote the question multiple times, obtaining the corresponding vector representation using BiLSTM, and calculated the similarity between the rewritten question and the initial question. Then, a second BiLSTM was used to encode the answer, and the similarity between the answer encoding and the rewritten question encoding was calculated. The final result was obtained through calculation. Models such as *NS-CQA* [25], *MRL-CQA* [26], *CIPITR* [27], and *SSRP* [28] convert the user’s input question into a corresponding program, which can be run directly on the knowledge graph to obtain the final answer. Gu et al. [29] performed entity linking on the entities in the question, then mined relevant information in the knowledge graph, obtaining the result using a sequence-to-sequence model based on Bert. *RnG-KBQA* [30] is similar to Gu’s method but directly uses the mined relevant information to combine candidate logical forms, which are then ranked using Bert. The top-ranked candidate logical forms are input into T5, and the final result is then obtained. *CBR-KBQA* [31]

is the first method that combines case-based reasoning with knowledge graph Q&A. After encoding the question, this method retrieves the dataset, extracts questions similar to it in the encoding vector space, and converts them, along with the corresponding logical forms, into training corpora, using BIGBIRD to obtain the final result.

Information-retrieval-based methods focus on using advanced retrieval techniques. They quickly find relevant information by matching keywords or phrases with entities and relationships in the knowledge graph. Their goal is to promptly provide users with accurate answers.

Yao et al. [32] proposed a knowledge graph Q&A method based on feature engineering, which primarily obtains the result through dependency syntactic analysis. This method extracts the corresponding features and adjusts the weights so that candidate feature graphs with higher relevance are assigned larger weights. Bordes et al. [32] proposed simultaneously mapping the question and the candidate entities in the knowledge graph to the vector space. Dong et al. [5] proposed the Multi-Column Convolution Neural Network (MCCNN) model, which focuses on three dimensions: the answer path, answer context, and answer type. Each of these dimensions corresponds to a trained convolutional network. Simultaneously, the word vectors of the triples and the question are trained and mapped to the same semantic space. Hao et al. [33] and Qu et al. [34] employed the attention mechanism to extract the implicit relevant information within the sentence. Naseri et al. [35] found that entity information plays a certain role and used related entity information to improve the representation of the entity. Lukovnikov et al. [36] enhanced performance through a hierarchical word and character set question encoder and were able to generate relatively independent entity–relationship representations. EmbedKGQA, however, [37] differs from these approaches. It found that the link performance in the graph embedding model is highly powerful and thus sought to improve the incompleteness of the knowledge graph by combining the graph embedding model with knowledge Q&A. Information-retrieval-based knowledge graph Q&A methods focus on establishing associations between user questions and the knowledge graph using efficient retrieval techniques. These methods first retrieve relevant entities and relationships from the knowledge graph using keyword or phrase matching and then generate answers. However, their limitation lies in their inability to handle semantically complex and deep-level questions, which may lead to poor performance in scenarios requiring a higher level of semantic understanding.

To address these challenges, this paper proposes a RAG-based Chinese power knowledge Q&A method enhanced by domain term recognition. First, the original text sequence is input into the LLM. The full names and abbreviations of the relevant professional terms in the power grid accident handling plan are also input into the LLM to rewrite the original text and mine its implicit information. Next, a prompt is added before the rewritten text to guide the model in performing information extraction, which is divided into two methods to adapt to different scenarios: dual-pointer extraction and generative extraction. Entities and relationships in the knowledge graph and the original text are then linked through vector recall and keyword recall. Vector recall utilizes the Moka Massive Mixed Embedding (M3E) model [17] to embed the entities and relationships in the original text and mine them in the knowledge base. Similarly, keyword recall measures the importance of keywords in the question and mines them in the knowledge base. The results of both recalls are reordered to obtain the final candidate entities and candidate relationships. Finally, the LLM receives executable logical form rules from the knowledge base, and both the original text sequence and the candidate entity relationship are input into the LLM to generate the corresponding logical form. After querying the knowledge base, the query results are fed back into the LLM to select the correct answer and generate a response.

The main contributions of this paper are as follows:

(1) A RAG knowledge Q&A framework enhanced by domain term recognition is proposed. First, external knowledge is input into the LLM to enhance the original text's representation and mine implicit information. Then, after information extraction and entity–relationship linking, the LLM generates a logical expression. Finally, the candidate answers are identified based on the logical expression, and the LLM generates a response.

(2) A Chinese power information extraction method enhanced by domain term recognition is proposed. The full names and abbreviations of the relevant professional terms in the power grid accident handling plan are input into the LLM to rewrite the original text. Then, the rewritten text is subjected to information extraction using either the dual-pointer method or the generative method, which effectively reduces model complexity and improves the model's generalization ability.

(3) A RAG power knowledge Q&A method based on two-way recall is proposed. To achieve entity linking and relationship linking, recall is performed in two dimensions: vectors and keywords. This method not only considers semantic information but also the text itself. Then, the LLM generates an executable logical form and queries the candidate triple set in the knowledge base. The candidate triple set and the original question text sequence are input into the LLM for final decision-making, generating a more natural language-like response.

Paper structure

(1) An information extraction framework incorporating domain term recognition enhancement. Technical terms and abbreviations relevant to power grid accident handling plans are pre-input into the LLM for learning, after which the LLM rewrites the original text. Prompts are then added before the text sequence to control the specific execution of the information extraction model. In the information extraction stage, two extraction methods, including dual-pointer extraction and generative extraction.

(2) A two-way recall RAG knowledge Q&A method. This method predicts entity and relationship links in vector and keyword dimensions, respectively. The LLM is provided with a canonical form executable on the knowledge base, and the original text sequence, candidate entities, and candidate relationships are input into the LLM to generate the corresponding canonical form. The query results are input into the LLM again to select the correct answer and generate a response.

(3) Experimental result. The LLM-adaptive RAG knowledge retrieval method, based on domain knowledge enhancement, not only outperforms popular methods in the power field but also demonstrates strong performance in the general field.

2 Establishment of the Power Knowledge Retrieval Model

2.1 Overall Framework

This paper proposes a RAG-based Chinese power knowledge Q&A framework enhanced by domain term recognition, as shown in Fig. 1. The framework is divided into four parts.

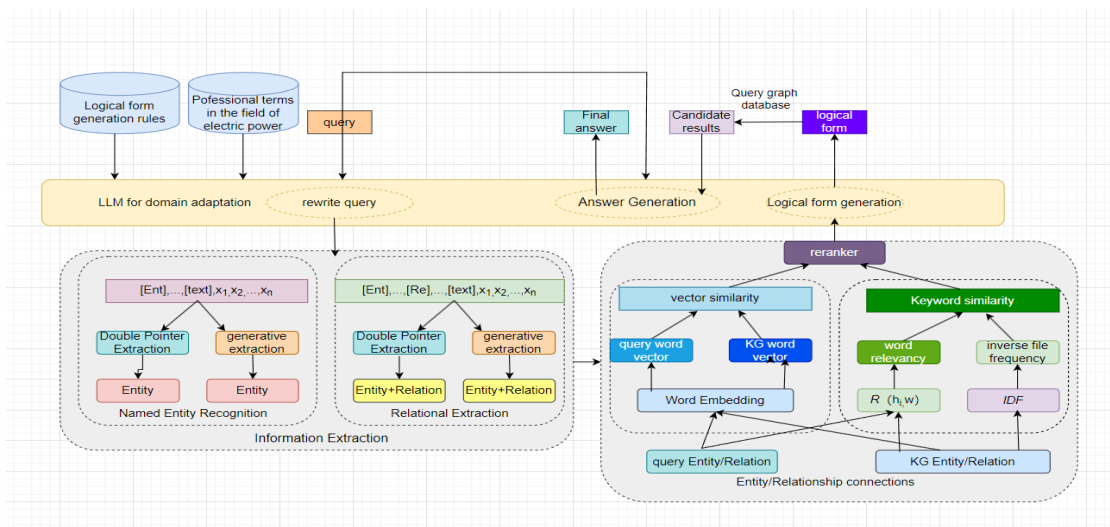


Fig. 1. Enhanced RAG knowledge quiz framework based on domain terminology recognition

First, domain term recognition enhancement: the full names and abbreviations of the professional terms in the power grid accident handling plan domain are input into the LLM for learning. The original text is rewritten to transform the implicit information into explicit information while maintaining the explicit information.

Second, the information extraction model: a prompt is added before the text sequence to guide the model in information extraction. Subsequently, when extracting entity relationships, two methods are adopted to adapt to different scenarios: dual-pointer extraction and generative extraction. Dual-pointer extraction can share the parameters of named entity recognition and relationship extraction, enabling the two tasks to interact and promote each other. Generative extraction encodes the text sequence, spliced with the prompt, into the large model and then decodes it, demonstrating stronger generalization ability.

Third, entity–relationship linking: this is divided into two recall methods, vector recall and keyword recall, for

entity linking and relationship linking. In the vector recall stage, the similarity of the embedded vectors is calculated to screen out the knowledge graph nodes or edges that best match the entities or relationships extracted from the original text. In the keyword recall stage, the similarity of the original text is calculated by comparing its keywords to find similar entities or relationships in the knowledge graph. The results of the two recalls are then comprehensively sorted. Vector recall focuses on semantic similarity, while keyword recall emphasizes surface text similarity. By comprehensively sorting the results of the two recalls, a more accurate and comprehensive recall effect can be achieved.

Fourth, answer generation: the comprehensively sorted results of the two recalls are input into the LLM. The LLM is provided with a logical form template to generate the corresponding logical form. A graph database query language is then generated based on the logical form to retrieve the answer from the knowledge graph. The retrieved triple set is subsequently input into the LLM again. The LLM selects the correct answer and generates a corresponding natural language answer based on the original question text sequence.

2.2 Domain Adaptation of the LLM

In the actual operation of the power grid accident handling plans, the original text often contains implicit information, which may prevent the model from effectively identifying important information. Additionally, the LLM lacks the rules for ultimately generating logical forms. To address these issues, the abbreviations of professional terms in the power grid accident handling plan domain and the Logical Form Generation Rules (LFGR) are first input into the LLM for learning. This enables the LLM to acquire the domain's professional knowledge. The LLM adopted in this paper is qwen-plus, with the specific formula illustrated in Equation 1:

$$qwen - plus = qwen - plus(f_1, f_2, \dots, f_n, lfgr) . \quad (1)$$

Where f_i represents the full names and abbreviations of the professional terms in the power grid accident handling plan domain, n represents the number of professional terms, and the output qwen-plus represents that it has learned the domain professional terms and logical form generation rules.

2.3 Information Extraction

The next step involves inputting the original text sequence into qwen-plus for rewriting, which transforms all implicit information into explicit information. The specific formula is presented in Equation 2:

$$x_1, x_2, \dots, x_n = qwen - plus(y_1, y_2, \dots, y_m) . \quad (2)$$

Where x_i represents the rewritten word and y_i represents the word in the original text. Since the purposes of named entity recognition and relationship extraction are different, the corresponding prompts are also different. If the task to be executed is named entity recognition, then [Ent]... is added before the original text sequence. If the task is to complete relationship extraction, then [Ent]...[Re]... must be added before the original text sequence. The specific formulas are presented in Equations 3 and 4:

$$t \oplus x = [[Ent], \dots, [text], x_1, x_2, \dots, x_n] . \quad (3)$$

$$t \oplus x = [[Ent], \dots, [Re], \dots, [text], x_1, x_2, \dots, x_n] . \quad (4)$$

Among them, t represents the added token, x represents the text sequence, and n represents the length of the text sequence. After splicing the text sequence and the prompt information, the spliced result is input into the encoder for encoding. Different information extraction tasks can be completed according to the prompt's guidance. The specific formula is presented in Equation 5:

$$U = \text{Ernie3.0}(t \oplus x) . \quad (5)$$

The encoder adopted is Ernie3.0, a pre-trained, large-scale, and knowledge-enhanced model that combines autoregressive and autoencoding networks. Therefore, by training this model, it can easily achieve zero-shot learning, few-shot learning, or fine-tuning for natural language understanding and generation tasks.

Dual-Pointer Extraction and Training. After embedding the text sequence into a vector U , this paper employs a softmax operation to process each vector separately. Through the flexible use of dual-pointers, the model can quickly and accurately, and with extremely high operational efficiency, locate the head and tail of the extracted word vectors. The specific formulas are presented in Equations 6 and 7:

$$U = \{u_1, u_2, \dots, u_n\} . \quad (6)$$

$$y = \text{soft max}(u_i), i \in 1, 2, 3, \dots, n . \quad (7)$$

Where u_i is the word vector after the text is embedded. The model determines the head and tail of the entity and relationship through pointers. The pointer's pointing position can be regarded as a multi-classification task, which includes four categories: entity head, entity tail, relationship head, and relationship tail. Since the cross-entropy loss function provides a clear direction in gradient descent optimization and is more suitable for classification problems, the loss function here adopts it. The specific formula is presented in Equation 8:

$$L(p, q) = -\sum_{i=1}^n p(x_i) \log(q(x_i)) . \quad (8)$$

Where p represents the one-hot encoding of the true label and q represents the predicted probability of the model for this category. The goal here is to minimize the cross-entropy between p and q .

Generative Extraction and Training. The target tasks and outputs of named entity recognition and relationship extraction differ. To make the model work and achieve a unified output structure, a unified structured output framework (Ent Name: Span (Re Name: Span)) is designed. Among them, Ent Name refers to the category of the entity segment, Re Name refers to the relationship category between two entity segments within the same sentence, and Span refers to the target segment in the original text. This ensures that the output after decoding is unified in this format.

In generative extraction, the hidden representation U , after splicing the original text sequence and the prompt information, is the linearized representation of the unified structure output. The token is then decoded. The specific formula is presented in Equation 9:

$$z_i, u_i^d = \text{Decoder}([U; u_1^d, u_2^d, \dots, u_{i-1}^d]) . \quad (9)$$

Where z_i is the i -th token in the sequence and u_i^d is the state of the decoder. The decoder is a transformer decoder, used to predict the conditional probability of z_i . The required information is then extracted from the linearized representation of the unified structure output.

Since the generative extraction method needs to output a unified structure in the end, this model must also consider the mapping and generation abilities from text to structure. In addition, the masked language can also improve the model's expression ability. To enable the model to possess the mapping ability from text to structure, it is pre-trained with words and word vectors. Specifically, the pre-training data is $D_p = \{(x, z)\}$, where the Ent type $s1$ and Re type $s2$ in z are extracted and calculated as $s = s1 \cup s2$. Subsequently, the model is pre-trained to ensure it acquires the desired mapping ability. The specific formula is presented in Equation 10:

$$L_1 = \sum_{(x,z) \in D_p} -\log p(z|x,s;\theta_e,\theta_d) . \quad (10)$$

Where θ_e and θ_d represent the parameters of the encoder and decoder, respectively. To ensure the validity of the structure generated by the model, the decoder of the model is also pre-trained to function as a structured language model. The specific formula is presented in Equation 11:

$$L_2 = \sum_{z \in D_e} -\log p(z_i|z_{<i};\theta_d) . \quad (11)$$

Where D_e represents a sample dataset of the unified structure. Through pre-training in this manner, the decoder can extract the rules of the unified structure. Furthermore, during pre-training to ensure the model's mapping ability from text to structure, the masked language model task is also employed on D_e . This task improves the model's expression ability and rationally utilizes damaged fragments, leveraging them to effectively alleviate potential catastrophic forgetting. The specific formula is presented in Equation 12:

$$L_3 = \sum_{x \in D_t} -\log p(x''|x';\theta_e,\theta_d) . \quad (12)$$

Where x' represents the damaged source text and x'' represents the damaged target fragment. The mapping from text to structure, the generation ability of the structure, and the expression ability of the model must be comprehensively considered to obtain the final loss function. Accordingly, a combination of these three loss functions forms the final loss function. The specific formula is presented in Equation 13:

$$L = L_1 + L_2 + L_3 . \quad (13)$$

2.4 Entity–Relationship Linking

After extracting the entities and relationships in the question, they need to be linked with those in the knowledge graph. A two-way recall strategy is thus adopted, primarily divided into two parts: vector recall and keyword recall. The vector recall stage focuses on semantic similarity, calculating the similarity of the embedded vectors to accurately match the entities or relationships in the knowledge graph. The keyword recall stage emphasizes surface text similarity, calculating the similarity of the original text to more comprehensively cover the potential answers in the knowledge graph. These two recall strategies work together to achieve a more accurate and comprehensive recall effect.

Vector Recall. In the vector recall stage, for the user's question and the relevant entities or relationships in the knowledge graph, the Moka Massive Mixed Embedding (M3E) model is first used to embed both into vectors. The specific formulas are presented in Equations 14 and 15:

$$h = M3E(h) . \quad (14)$$

$$w = M3E(w) . \quad (15)$$

Where h represents the entity or relationship in the user's question and w represents the entity or relationship in the knowledge graph. Then, the result is determined by comparing the similarity of the vectors. The specific formula is presented in Equation 16:

$$\text{cosine}(h,w) = \left\langle \frac{h}{\|h\|_2}, \frac{w}{\|w\|_2} \right\rangle . \quad (16)$$

Where $\text{cosine}(\mathbf{h}, \mathbf{w})$ is used to calculate the cosine similarity between the vector \mathbf{h} in the user's question and the vector \mathbf{w} in the knowledge graph; $\langle \mathbf{h}, \mathbf{w} \rangle$ is the dot product of \mathbf{h} and \mathbf{w} ; and $\|\mathbf{h}\|_2$ and $\|\mathbf{w}\|_2$ are the 2-norms of \mathbf{h} and \mathbf{w} , respectively.

Keyword Recall. Inverse Document Frequency (IDF) is an important indicator used to measure the importance of a certain word within an entire document collection. In the keyword recall of the knowledge graph Q&A system in this paper, IDF is used to adjust the weight of the entity to better reflect its information value. The specific formula is presented in Equation 17:

$$IDF(h_i) = \log \frac{N - n(h_i) + 0.5}{n(h_i) + 0.5} . \quad (17)$$

Where h_i represents the entity or relationship in the original text, $n(h_i)$ represents the number of nodes in the knowledge graph that contain the entity or the number of edges that contain the relationship, and N represents the total number of all entities or relationships in the knowledge graph. $IDF(h_i)$ is typically used as the weight related to h_i . A lower IDF value indicates that the entity is more common in the knowledge graph, while a higher IDF value indicates that although the entity or relationship is relatively rare in the knowledge graph, its importance in recall may be higher. This reflects the role of the IDF value, which helps the model distinguish the importance of different entities or relationships, thus improving recall accuracy.

Then, the correlation between the entity relationship in the original question text sequence and the entity relationship in the knowledge graph is calculated. The specific formula is presented in Equation 18:

$$R(h_i, w) = \frac{f_i \cdot (k_1 + 1)}{f_i + k_1 \cdot \left(1 - b + b \cdot \frac{wl}{avgwl}\right)} \cdot \frac{h_{f_i} \cdot (k_2 + 1)}{h_{f_i} + k_2} . \quad (18)$$

Where w represents the entity or relationship in the knowledge graph; f_i represents the frequency of occurrence of the entity or relationship in the knowledge graph; h_{f_i} represents the frequency of occurrence of the entity or relationship in the original question; wl represents the length of w ; $avgwl$ represents the average length; and k_1 , k_2 , and b are all adjustment factors. The default value of k_1 is 1.2, which controls the rate at which word frequency saturates. The smaller this value is, the faster the rate of saturation. Conversely, the larger this value is, the slower the rate of saturation. The default value of b is typically 0.75, which controls the field length normalization value. When this value is 0.0, normalization is disabled, and when this value is 1.0, full normalization is enabled. Since, in most cases of knowledge graph Q&A, h_i appears only once in the original question—that is, $h_{f_i}=1$ —the above formula can be simplified, as shown in Equation 19:

$$R(h_i, w) = \frac{f_i \cdot (k_1 + 1)}{f_i + k_1 \cdot \left(1 - b + b \cdot \frac{wl}{avgwl}\right)} . \quad (19)$$

Subsequently, by comprehensively considering the inverse document frequency, that is, the weight of the entity or relationship and the correlation between the entity or relationship, the final score can be calculated. The final score represents the correlation of the candidate entity or candidate relationship. The higher the score, the higher the correlation and the higher the ranking. The specific formula is presented in Equation 20:

$$Score(H, w) = \sum_{i=1}^n IDF(h_i) \cdot R(h_i, w) . \quad (20)$$

To make full use of the results of vector recall and keyword recall, comprehensive sorting is required. This section uses bge-reranker to comprehensively reorder the results. The specific formula is presented in Equation 21:

$$Candidate = bge - reranker(Candidate_{vector}, Candidate_{keyword}) . \quad (21)$$

Where *Candidate* represents the reordered candidate entity set or candidate relationship set, *Candidate*_{vector} represents the candidate entity set or candidate relationship set recalled by the vector, and *Candidate*_{keyword} represents the candidate entity set or candidate relationship set recalled by the keyword.

2.5 Query Based on Logical Form Generation

Logical Form Generation. After completing the comprehensive sorting of the candidate entities and candidate relationships, the model inputs the top ten candidate entities and candidate relationships into qwen-plus, which then generates the corresponding executable logical form. The specific formula is presented in Equation 22:

$$LF = qwen - plus(e_1, e_2, \dots, e_{10}, r_1, r_2, \dots, r_{10}) . \quad (22)$$

Where e_i represents the candidate entity, r_i represents the candidate relationship, and *LF* represents the generated candidate logical form set.

Query Generation. After obtaining the logical form, the logical form is used to query the knowledge graph. The candidate result, Ca_i , and the original question text sequence are then simultaneously input into qwen-plus, which selects the most likely correct answer and generates a fluent natural language. The specific formula is presented in Equation 23.

$$Answer = qwen - plus(query, Ca_1, Ca_2, \dots, Ca_{100}) . \quad (23)$$

3 Experiments and Results Analysis

3.1 Datasets

The simulation experiment datasets in this paper include the professional dataset Ppd (Power Project Dataset) from the power field, as well as the general domain datasets WebQSP (Web Question Sparse) and CWQ (Complex Web Question Answering). The data for Ppd are sourced from real grid accident handling plan information. It is a professional and confidential dataset containing 2,206 question-and-answer records for grid accident handling plans. WebQSP is a dataset focusing on the conversion from natural language questions to SPARQL queries, comprising 3,098 training sets and 1,639 test sets. For a fair comparison with the experimental results from Ppd, 2,206 data points are randomly extracted from WebQSP and translated into Chinese for use in this study. CWQ is a dataset in the research field for complex knowledge base question answering, containing 27,734 training sets, 3,480 validation sets, and 3,475 test sets. Similarly, for a fair comparison with the experimental results of Ppd, 2,206 data points are randomly extracted from CWQ and translated into Chinese. It should be noted that three random extraction experiments are carried out on WebQSP and CWQ in this paper, with the experiment exhibiting the best performance selected as the experimental result.

3.2 Baseline Methods

Several current popular models are selected as baselines in this paper, as follows:

(1) TPLinker [38]: A single-module, single-step joint extraction model is proposed. It shares the same decoder for named entity recognition and relationship extraction, extracting both entities and relationships simultaneously, effectively solving the problem of exposure bias.

(2) SPN [39]: This paper directly regards entity–relationship joint extraction as the prediction of sets, effectively solving the ranking problem of multiple candidate triples.

(3) CGT [40]: This paper proposes a training framework for the contrastive learning of triples, along with a batch dynamic attention mask and a triple calibration mechanism, to improve model performance.

(4) PRGC [41]: This paper divides the triple extraction task into three subtasks: relationship judgment, entity extraction, and subject–object alignment. It proposes a triple joint extraction framework based on potential relationships and global correspondence.

(5) STAGG [42]: A semantic parsing framework for a question-answering system based on a knowledge base is proposed. In this framework, a one-stage search question can be formed, simplifying semantic parsing into the generation of a query graph.

(6) AQGnet [43]: A two-stage formal query construction method is proposed. In the first stage, the proposed model can predict the query structure of the question. In the second stage, this method is used to rank candidate queries.

(7) QGG [44]: The proposed model can handle both constrained questions and questions with multi-hop relationships.

(8) ReTraCk [45]: This paper proposes ReTraCk, which includes a retriever for efficiently retrieving relevant knowledge base items, a converter for generating logically correct logical forms, and a checker for improving the transduction process.

(9) RnG-KBQA [46]: A ranking and generation-based method is proposed. This model improves the answer coverage by reasonably constructing a generation model and, therefore, has strong generalization ability.

(10) FC-KBQA [47]: A fine-to-coarse KBQA combination framework (FC-KBQA) is proposed to ensure the generalization ability and executability of logical expressions.

(11) KnowGPT [16]: A black-box knowledge injection framework for LLM-based question answering is proposed, KnowGPT, which uses deep reinforcement learning to extract relevant knowledge from the knowledge graph. It also employs a multi-armed bandit approach to construct the most appropriate prompt for each question.

(12) Adaptive-RAG [48]: An adaptive QA framework is proposed, which can dynamically select the most suitable strategy for the LLM, ranging from the simplest to the most complex, based on query complexity.

3.3 Evaluation Metrics

To prove the effectiveness of the model in the field of grid accident handling plans, evaluation criteria are formulated. Accuracy (*Acc*) measures the closeness between a given measurement value and its true value. Recall (*Rec*) represents the proportion of all positive samples that are correctly predicted. The F1 value can be regarded as the weighted average of precision and recall. In this paper, precision is treated as equivalent to accuracy, hence only accuracy is used. The overall performance of the model can be comprehensively evaluated using accuracy, recall, and F1. The specific formulas are shown in Formulas 24, 25, 26, and 27:

$$Acc = (TP + TN) / (TP + FN + TN + FP) . \quad (24)$$

$$Pre = TP / (TP + FP) . \quad (25)$$

$$Rec = TP / (TP + FN) . \quad (26)$$

$$F_1 = 2 Pre \times Rec / (Pre + Rec) . \quad (27)$$

In addition to accuracy and the F1 value, Hits@1 is also used as an evaluation metric. The Hits@n metric is commonly employed in link prediction tasks and is widely used in knowledge question answering. It represents the average proportion of triples ranked less than or equal to n in the link prediction results. The larger the Hits@n value, the more triples ranked at the top are predicted in the link prediction task, thus reflecting better model performance. Therefore, Hits@n is an important metric in evaluating the performance of knowledge question answering models. The specific calculation method is shown in Formula 28:

$$\text{Hits}@n = \frac{1}{|S|} \sum_{i=1}^{|S|} \mathbb{I}(\text{rank}_i \leq n) . \quad (28)$$

Where $\mathbb{I}(\bullet)$ is an indicator function. Specifically, if the condition is true, the function value is 1, otherwise it is 0. In this paper, $n = 1$.

3.4 Experimental Results Analysis

The data presented in Table 1 show that the model proposed in this paper demonstrates excellent performance for the professional dataset Ppd in the power field. Whether accuracy, the F1 value, or Hits@1, the proposed model exhibits significantly better performance than that of the baseline models. For the general dataset, although only one metric is higher than the baseline, the difference from the best performance is not substantial, demonstrating the overall balance of the model. This indicates that the approach adopted in this paper—performing information extraction after rewriting the original text, screening candidate entities through vector recall and keyword recall, and ultimately selecting and generating an answer using qwen-plus—effectively improves the overall performance of the knowledge graph question answering model. Specifically, the accuracy of the method in this paper is improved by 0.3% compared to the current most advanced method on the professional dataset in the power field. Moreover, the F1 value is improved by 0.6% and Hits@1 by 0.2%. Although the performance of the model in this paper is slightly reduced on the key datasets WebQSP and CWQ for knowledge question answering, the accuracy on WebQSP is only 0.2% lower than that of the most advanced baseline method. Moreover, the F1 value is reduced by 0.1% and Hits@1 is 0.1% higher than the optimal baseline method. On CWQ, the accuracy is 1.6% lower than that of the most advanced baseline method, the F1 value is reduced by 1.4%, and Hits@1 is reduced by 2.3%. Nevertheless, the model still demonstrates excellent performance on WebQSP and good performance on CWQ. The experimental results show that the model proposed in this paper is more suitable for knowledge graph question answering in the field of grid accident handling plans than the other baseline models, while also exhibiting good performance in the general field.

Table 1. Experimental results of the knowledge question answering model on Ppd, WebQSP, and CWQ (%)

Model	Ppd			WebQSP			CWQ			
	Acc	F1	Hits@1	Acc	F1	Hits@1	Acc	F1	Hits@1	
Deep learning	STAGG	70.3	72.1	63.4	/	50.5	49.3	/	40.6	38.6
	AQGnet	72.6	74.5	64.7	/	51.4	49.9	/	32.8	31.3
	QGG	80.6	82.1	64.5	/	71.9	57.1	/	39.7	36.8
	ReTrack	82.8	84.5	69.1	/	70.1	58.6	/	53.4	47.1
	RnG-KBQA	83.4	85.4	71.3	/	74.3	62.8	/	52.6	46.3
	FC-KBQA	83.5	85.4	71.2	/	75.9	62.9	/	49.3	42.9
R	KnowGPT	83.9	86.2	71.9	75.3	77.1	64.1	60.1	59.4	53.4
A	Adaptive-RAG	81.3	83.3	71.7	73.1	74.9	62.3	65.5	66.7	59.6
G		Ours-generation	81.9	83.7	71.4	72.9	74.9	62.7	63.7	65.1
	Ours-double pointer	84.2	86.8	72.1	75.1	77.0	64.2	63.9	65.3	57.3

As shown in Fig. 2, this is a performance comparison bar chart of knowledge question answering models on the Ppd dataset. List 10 models on the horizontal axis, covering 10 models such as STAGG and AQGnet; The vertical axis represents the performance indicator values, ranging from 0-80, represented by blue (accuracy Acc), orange (F1 score, comprehensive accuracy and recall), and green (Hits@1, Top-1 hit rate), distinguish between three types of indicators. Overall, for most models, $F1 > Acc > Hits@1$, this reflects that under the Ppd dataset, the comprehensive evaluation (F1) of various models performs better, and it is difficult to accurately hit the Top1. Specifically, early models such as STAGG performed relatively poorly, while Ours double pointer performed outstandingly, with an Acc of about 84, an F1 of about 87 and Hits@1 Around 72, KnowGPT and other models have also shown good performance. Reflecting the differences in knowledge question answering adaptability and

algorithm logic among different models, which can be used for subsequent model optimization (such as targeting Hits@1 Shortcomings) provide reference.

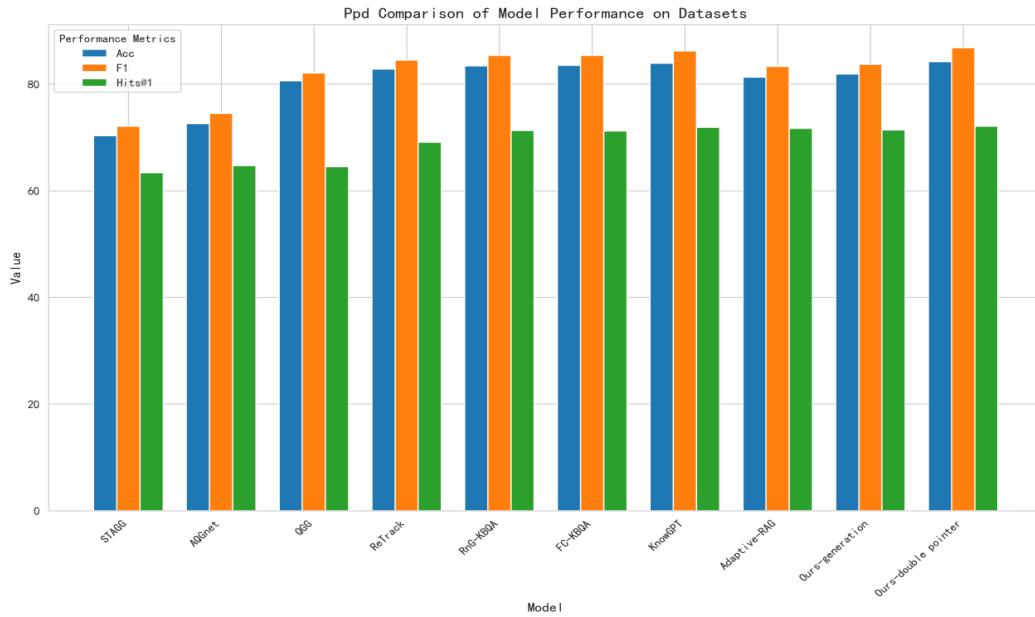


Fig. 2. Performance comparison chart of knowledge question answering models on Ppd dataset

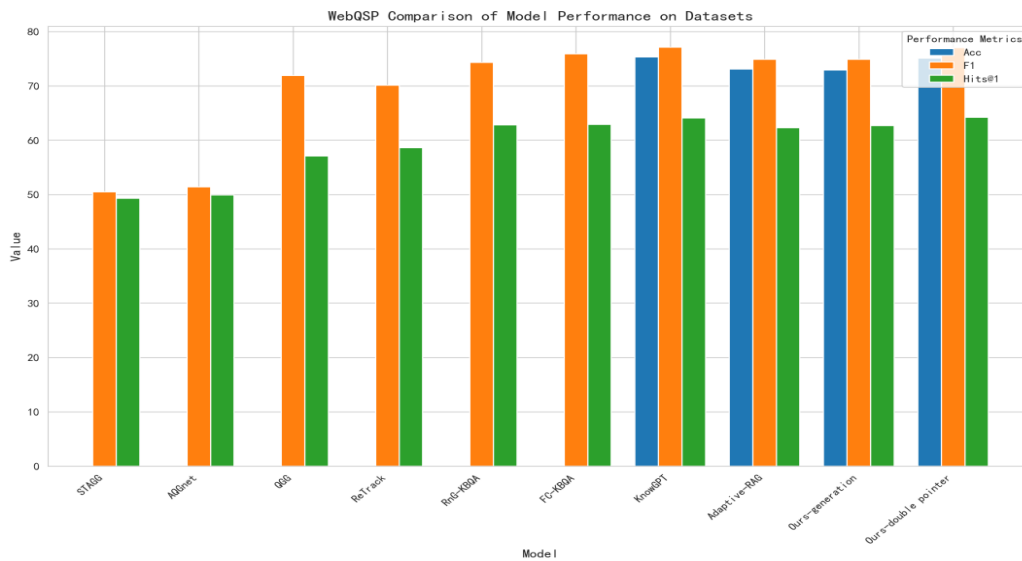


Fig. 3. Comparison of model performance on WebQSP dataset

As shown in Fig. 3, this is a bar chart comparing the performance of models on the WebQSP dataset. The horizontal axis lists 10 models including QGG and ReTrack, while the vertical axis shows performance values (0-80), with blue (Acc, accuracy), orange (F1, comprehensive evaluation), and green (Hits@1, Top -1 hit rate) 3 distinguishing indicators. For most models, F1 (orange) performs outstandingly, such as some models with F1 exceeding 70, Acc (blue) and Hits@1 (Green) Slightly lower. Reflecting that under the WebQSP dataset, the model’s comprehensive evaluation dimension is better, and it is difficult to accurately hit Top1. Models such as

“Ours double pointer” have relatively balanced and high numerical values for the three indicators, with high performance; Some early model indicators were low, reflecting the adaptability and algorithm differences of different models, providing direction for optimization.

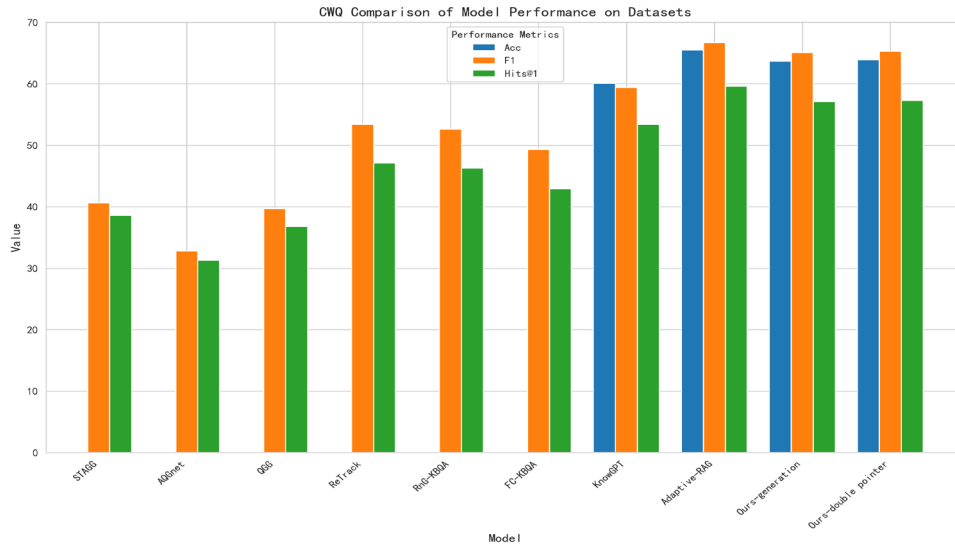


Fig. 4. Comparison of model performance on CWQ dataset

As shown in Fig. 4, this is a bar chart comparing the performance of models on the CWQ dataset. The horizontal axis represents 10 models including STAGG and AQNet, and the vertical axis represents performance values (0-70), with blue (Acc, accuracy), orange (F1, comprehensive evaluation), and green (Hits@1, Top -1 hit rate) 3 distinguishing indicators. Overall, most models have F1 (orange)>Acc (blue)> Hits@1 (Green). In the CWQ dataset, the overall performance (F1) of the model is relatively better, and it is difficult to accurately hit the Top1. Early models such as STAGG and AQNet had lower performance indicators, while KnowGPT, Adaptive RAG, Ours generation, Ours double pointer and other models had better performance, with three more prominent indicators. Reflecting the adaptability and algorithmic logic differences of different models in the CWQ dataset, Optimize for subsequent models (such as improving Hits@1 shortcomings) Provide reference.

Table 2. Experimental results of the information extraction model on Ppd (%)

Model	NER			RE		
	Acc	Rec	F1	Acc	Rec	F1
TPLinker	84.8	87.9	86.3	91.2	88.1	89.6
SPN	86.9	86.1	86.5	89.7	83.9	86.7
CGT	84.5	82.3	83.4	84.4	86.8	85.6
PRGC	88.3	83.6	85.9	82.7	85.5	84.1
Ours-generation	86.4	86.0	86.2	87.3	85.6	86.4
Ours-double pointer	87.9	85.6	86.7	91.1	89.5	90.2

The data presented in Table 2 show that the information extraction model proposed in this paper exhibits excellent performance for the professional dataset Ppd in the power field. Although the accuracy and recall rates of the NER results are lower than the optimal results of the baseline model, the F1 value is 0.2% higher than the optimal result of the baseline model. Similarly, although the accuracy of the RE result is 0.1% lower than the optimal result of the baseline model, both the recall rate and F1 value are higher than in other baseline models. The experimental results show that the information extraction model proposed in this paper is more balanced in performance compared to the other baseline models. Moreover, it is more suitable for handling information extraction in the power field.

3.5 Ablation Experiments

The Role of Rewriting and Prompt. To study the role of original text rewriting and prompts in information extraction, an ablation experiment was conducted. This experiment compared the performance of the information extraction model without rewriting and without the addition of prompts to that of the original information extraction model.

Table 3. Ablation experiment results of rewriting and prompts in information extraction (%)

	NER			RE		
	Acc	Rec	F1	Acc	Rec	F1
w/o rewrite	85.9	85.1	85.5	90.1	88.2	89.1
w/o prompt	83.1	80.6	81.8	86.3	85.7	86.0
Ours	87.9	85.6	86.7	91.1	89.5	90.2

As shown in Table 3, when information extraction is directly performed without rewriting, the performance is reduced in all three evaluation metrics. This indicates that implicit information in the text is helpful for information extraction. When the prompt is not retained, the performance of the model is significantly reduced, indicating that noise interference or other problems may occur in the information extraction stage of the model. This results in the model’s incorrect judgment of the user’s real needs, thereby affecting the final performance.

The Role of Two-Way Recall and LLM. To study the role of two-way recall and LLMs in knowledge question answering, an ablation experiment was conducted to compare the performance of the knowledge question answering model with single-way recall and without using qwen-plus on the Ppd dataset to that of the original knowledge question answering model.

Table 4. Ablation experiment results of two-way recall and LLMs in knowledge question answering (%)

	Acc	F1	Hits@1
w/o keyword recall	78.4	79.5	70.3
w/o vector recall	73.3	77.2	67.9
w/o qwen-plus	80.3	82.3	71.7
Ours	84.2	86.8	72.1

Table 4 clearly shows that whether only vector recall or only keyword recall is retained, the final performance of the model is reduced. By learning the semantic representation of the text, vector recall captures the semantic association between entities, bridging the recalled entities closer to the semantic meaning of the user’s query. Keyword recall performs entity recall through keyword matching and is more suitable for some specific scenarios. Therefore, the synergy between vector recall and keyword recall makes the recall results more comprehensive and accurate.

In addition, the experimental results in Table 4 also prove that the reasonable use of qwen-plus can affect the accuracy of the final answer. When the candidate entity ranked first after comprehensive ranking is directly selected as the final result, the model’s performance is significantly reduced. This indicates that simply selecting the top-ranked entities according to the comprehensive ranking results cannot accurately answer the user’s questions.

4 Conclusion

In this paper, a RAG-based Chinese power knowledge question answering method enhanced by domain term recognition is proposed for grid accident handling plans. The implicit information in the text is mined through rewriting, and the information extraction task is controlled by using prompts. Two methods, namely double-pointer extraction and generative extraction, are proposed. A two-way recall strategy is then adopted for entity linking and relationship linking. Vector recall more accurately captures the semantic association between entities or relationships, while keyword recall is more suitable when the user’s question is more straightforward. The combi-

nation of these two recall methods makes the recalled candidate entity and candidate relationship sets more comprehensive. Finally, the candidate entity and candidate relationship sets are input into qwen-plus to generate an executable logical form, which is then queried in the knowledge graph. The query result and the original question are input into qwen-plus again for final decision-making and answer generation. On the Ppd dataset, the accuracy of the model reaches 84.2%, an F1 value of 86.8%, and Hits@1 of 72.1%. On the WebQSP dataset, the accuracy of the model reaches 75.1%, an F1 value of 77.0%, and Hits@1 of 64.2%. On the CWQ dataset, the accuracy of the model reaches 63.9%, an F1 value of 65.3%, and Hits@1 of 57.3%. The simulation experimental results show that the RAG-based Chinese knowledge question answering method, enhanced by domain term recognition, not only outperforms the baseline model in the power field but also excels in the general field.

Acknowledgements

This work was supported by the Natural Science Foundation of Gansu Province of China (21JR7RE174). 2024 Research Project on Educational and Teaching Reform at the School Level of Tianshui Normal University (JGG-24241435).

References

- [1] Z. Ma, K. Yan, H. Wang, BERT-based Question Answering using Knowledge Graph Embeddings in Nuclear Power Domain, in: Proceedings of the 2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2023.
<https://doi.org/10.1109/CSCWD57460.2023.10152692>
- [2] J. Chen, J. Jia, L. Wang, C. Zhong, B. Wu, Carbon Reduction Countermeasure from a System Perspective for the Electricity Sector of Yangtze River Delta (China) by an Extended Logarithmic Mean Divisia Index (LMDI), *Systems* 11(3)(2023) 117.
<https://doi.org/10.3390/systems11030117>
- [3] Y. Zhou, Z. Lin, L. Tu, Q. Lv, Online Document Transmission and Recognition of Digital Power Grid with Knowledge Graph, *EAI Endorsed Transactions on Scalable Information Systems* 10(3)(2023) e5.
<https://doi.org/10.4108/eetsis.v10i3.2831>
- [4] L. Zeng, X. Yang, Spatial-temporal Attention Model Based on Transformer Architecture for Anomaly Detection in Multivariate Time Series Data, *Journal of Computers (Taiwan)* 35(3)(2024) 193-207.
<https://doi.org/10.53106/199115992024063503014>
- [5] L. Dong, F.R. Wei, M. Zhou, K. Xu, Question answering over Freebase with multi-column convolutional neural networks, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015.
<https://doi.org/10.3115/v1/P15-1026>
- [6] K. Xu, S. Reddy, Y. Feng, S. Huang, D. Zhao, Question answering on Freebase via relation extraction and textual evidence, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, 2016.
<https://doi.org/10.18653/v1/P16-1220>
- [7] Z. Liu, R. Zhou, K. Huang, X. Hu, Z. Jiang, B. Cai, K. Yuan, Intrusion Detection Based on Feature Reduction and Model Pruning in Electricity Trading Network, *Journal of Computers* 34(5)(2023) 213-227.
<https://doi.org/10.53106/199115992023103405017>
- [8] W.T. Yih, M.W. Chang, X. He, J.F. Gao, Semantic parsing via staged query graph generation: question answering with knowledge base, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing, 2015.
<https://doi.org/10.3115/v1/P15-1128>
- [9] D. Ortega, N.T. Vu, Neural-based context representation learning for dialog act classification, in: Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue, 2017.
<https://doi.org/10.18653/v1/W17-5530>
- [10] T. Zhang, S.G. Patil, N. Jain, Sheng Shen, M. Zaharia, I. Stoica, J.E. Gonzalez, RAFT: Adapting language model to domain specific RAG. <<https://arxiv.org/abs/2403.10131v2>>, 2024 (accessed 04.12.2024).
- [11] Y. Gao, Y. Xiong, X. Gao, K. Jia, J. Pan, Y. Bi, Y. Dai, J. Sun, M. Wang, H. Wang, Retrieval-augmented generation for large language models: A survey. <<https://arxiv.org/abs/2312.10997>>, 2023 (accessed 04.12.2024).
- [12] X. Ma, Y. Gong, P. He, H. Zhao, N. Duan, Query rewriting in retrieval-augmented large language models, in: The 2023 Conference on Empirical Methods in Natural Language Processing, 2023.
<https://doi.org/10.18653/v1/2023.emnlp-main.322>

- [13] M. Yu, W. Yin, K.S. Hasan, C. dos Santos, B. Xiang, B. Zhou, Improved neural relation detection for knowledge base question answering, in: Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics (ACL), 2017.
<https://doi.org/10.18653/v1/P17-1053>
- [14] H. Sun, B. Dhingra, M. Zaheer, K. Mazaitis, R. Salakhutdinov, W. Cohen, Open domain question answering using early fusion of knowledge bases and text, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018.
<https://doi.org/10.18653/v1/D18-1455>
- [15] A. Talmor, J. Berant, The web as a knowledge-base for answering complex questions, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), 2018.
<https://doi.org/10.18653/v1/N18-1059>
- [16] Q. Zhang, J. Dong, H. Chen, X. Huang, D. Zha, Z. Yu, KnowGPT: Black-box knowledge injection for large language models. <<https://arxiv.org/abs/2312.06185v1>>, 2023 (accessed 04.12.2024).
- [17] S. Xiao, Z. Liu, P. Zhang, N. Muennighof, C-Pack: Packaged resources to advance general Chinese embedding. <<https://arxiv.org/abs/2309.07597v1>>, 2023 (accessed 04.12.2024).
- [18] J. Berant, A. Chou, R. Frostig, P. Liang, Semantic parsing on Freebase from question-answer pairs, in: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, 2013.
<https://aclanthology.org/D13-1160/>
- [19] H. Bast, E. Haussmann, More accurate question answering on Freebase, in: Proceedings of the 24th ACM International Conference on Information and Knowledge Management, 2015.
<https://doi.org/10.1145/2806416.2806472>
- [20] W. Zheng, L. Zou, X. Lian, J.X. Yu, S. Song, D. Zhao, How to build templates for RDF question/answering: An uncertain graph similarity join approach, in: SIGMOD'15: Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, 2015.
<https://doi.org/10.1145/2723372.2747648>
- [21] A. Abujabal, M. Yahya, M. Riedewald, G. Weikum, Automated template generation for question answering over knowledge graphs, in: Proceedings of the 26th International Conference on World Wide Web, 2017.
<https://doi.org/10.1145/3038912.3052583>
- [22] A. Abujabal, R.S. Roy, M. Yahya, G. Weikum, Never-ending learning for open-domain question answering over knowledge bases, in: Proceedings of the 2018 World Wide Web Conference, 2018.
<https://doi.org/10.1145/3178876.3186004>
- [23] S. Yavuz, I. Gur, Y. Su, M. Srivatsa, X. Yan, Improving semantic parsing via answer type inference, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016.
<https://doi.org/10.18653/v1/D16-1015>
- [24] L. Dong, J. Mallinson, S. Reddy, M. Lapata, Learning to paraphrase for question answering. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017.
<https://doi.org/10.18653/v1/D17-1091>
- [25] Y. Hua, Y.-F. Li, G. Qi, W. Wu, J. Zhang, D. Qi, Less is more: Data-efficient complex question answering over knowledge bases, *Journal of Web Semantics* 65(2020) 100612.
<https://doi.org/10.1016/j.websem.2020.100612>
- [26] Y. Hua, Y.F. Li, G. Haffri, G. Qi, T. Wu, Few-shot complex knowledge base question answering via meta reinforcement learning, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020.
<https://doi.org/10.18653/v1/2020.emnlp-main.469>
- [27] A. Saha, G.A. Ansari, A. Laddha, K. Sankaranarayanan, S. Chakrabarti, Complex program induction for querying knowledge bases in the absence of gold programs, *Transactions of the Association for Computational Linguistics* 7(2019) 185–200.
https://doi.org/10.1162/tacl_a_00262
- [28] G.A. Ansari, A. Saha, V. Kumar, M. Bhambhani, K. Sankaranarayanan, S. Chakrabarti, Neural program induction for KBQA without gold programs or query annotations, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, 2019.
<https://doi.org/10.24963/ijcai.2019/679>
- [29] Y. Gu, S. Kase, M. Vanni, B. Sadler, P. Liang, X. Yan, Y. Su, Beyond I.I.D.: Three levels of generalization for question answering on knowledge bases, in: Proceedings of the Web Conference 2021, 2021.
<https://doi.org/10.1145/3442381.3449992>
- [30] X. Ye, S. Yavuz, K. Hashimoto, Y. Zhou, C. Xiong, RNG-KBQA: Generation augmented iterative ranking for knowledge base question answering, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022.
<https://doi.org/10.18653/v1/2022.acl-long.417>
- [31] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, A. Ahmed, Big Bird: Transformers for longer sequences, in: Proc. Advances in Neural Information Processing Systems, 2020.

- https://papers.neurips.cc/paper_files/paper/2020/file/c8512d142a2d849725f31a9a7a361ab9-Paper.pdf
- [32] X.C. Yao, B. Van Durme, Information extraction over structured data: Question answering with Freebase, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014. <https://doi.org/10.3115/v1/P14-1090>
- [33] Y.C. Hao, Y.Z. Zhang, K. Liu, S. He, Z. Liu, H. Wu, J. Zhao, An end-to-end model for question answering over knowledge bases with cross-attention combining global knowledge, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017. <https://doi.org/10.18653/v1/P17-1021>
- [34] Y.Q. Qu, J. Liu, L.Y. Kang, Q. Shi, D. Ye, Question answering over Freebase via attentive RNN with similarity matrix based CNN. <<https://arxiv.org/abs/1804.03317>>, 2018 (accessed 04.12.2024).
- [35] S. Naseri, J. Foley, J. Allan, B.T. O'Connor, Exploring summary-expanded entity embeddings for entity retrieval, in: Proceedings of the CIKM 2018 Workshops Co-located with 27th ACM International Conference on Information and Knowledge, 2018. <https://ceur-ws.org/Vol-2482/paper7.pdf>
- [36] D. Lukovnikov, A. Fischer, J. Lehmann, S. Auer, Neural network-based question answering over knowledge graphs on word and character level, in: Proceedings of the 26th International Conference on World Wide Web, 2017. <https://doi.org/10.1145/3038912.3052675>
- [37] A. Saxena, A. Tripathi, P. Talukdar, Improving multi-hop question answering over knowledge graphs using knowledge base embeddings, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020. <https://doi.org/10.18653/v1/2020.acl-main.412>
- [38] Y. Wang, B. Yu, Y. Zhang, T. Liu, H. Zhu, L. Sun, TPLinker: Single-stage joint extraction of entities and relations through token pair linking, in: Proceedings of the 28th International Conference on Computational Linguistics, 2020. <https://doi.org/10.18653/v1/2020.coling-main.138>
- [39] D. Sui, Y. Chen, K. Liu, J. Zhao, X. Zeng, S. Liu, Joint entity and relation extraction with set prediction networks. <<https://arxiv.org/abs/2011.01675>>, 2020 (accessed 04.12.2024).
- [40] H. Ye, N. Zhang, S. Deng, M. Chen, C. Tan, F. Huang, H. Chen, Contrastive triple extraction with generative transformer, in: Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, 2021. <https://cdn.aaai.org/ojs/17677/17677-13-21171-1-2-20210518.pdf>
- [41] H. Zheng, R. Wen, X. Chen, Y. Yang, Y. Zhang, Z. Zhang, N. Zhang, B. Qin, X. Ming, Y. Zheng, PRGC: Potential relation and global correspondence based joint relational triple extraction, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, 2021. <https://doi.org/10.18653/v1/2021.acl-long.486>
- [42] W.-T. Yih, M.-W. He, X. Chang, J. Gao, Semantic parsing via staged query graph generation: Question answering with knowledge base, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2015. <https://doi.org/10.3115/v1/P15-1128>
- [43] Y. Chen, H. Li, Y. Hua, G. Qi, Formal query building with query structure prediction for complex question answering over knowledge base, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence Main Track, 2020. <https://doi.org/10.24963/ijcai.2020/519>
- [44] Y. Lan, J. Jiang, Query graph generation for answering multi-hop complex questions from knowledge bases, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020. <https://doi.org/10.18653/v1/2020.acl-main.91>
- [45] S. Chen, Q. Liu, Z. Yu, C.-Y. Lin, J.-G. Lou, F. Jiang, ReTraCk: A flexible and efficient framework for knowledge base question answering, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing: System Demonstrations, 2021. <https://doi.org/10.18653/v1/2021.acl-demo.39>
- [46] X. Ye, S. Yavuz, K. Hashimoto, Y. Zhou, C. Xiong, RNG-KBQA: Generation augmented iterative ranking for knowledge base question answering, in: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2022. <https://doi.org/10.18653/v1/2022.acl-long.417>
- [47] L. Zhang, J. Zhang, Y. Wang, S. Cao, X. Huang, C. Li, H. Chen, J. Li, FC-KBQA: A fine-to-coarse composition framework for knowledge base question answering, in: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2023. <https://doi.org/10.18653/v1/2023.acl-long.57>
- [48] S. Jeong, J. Baek, S. Cho, S.J. Hwang, J. Park, Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity, in: Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 2024. <https://doi.org/10.18653/v1/2024.naacl-long.389>

Appendix

(a) The Prompt of qwen-plus for Original Text Rewriting

Table 5 shows the prompt for inputting the professional nouns and abbreviations related to the grid accident handling plan domain into qwen-plus. Next, for inputting the original text sequence and using qwen-plus to rewrite the original text.

Table 5. Guidance for qwen-plus in original text rewriting

The following are the professional nouns and their abbreviations in the grid accident handling plan domain:

According to the professional nouns and their abbreviations in the grid accident handling plan domain, you must rewrite the following input text sequence, where rewriting rules are as follows:

1. Do not change the meaning of the original text sequence.
2. Complete the abbreviations of the professional nouns in the original text sequence.
3. Complete the abbreviations of the general words in the original text sequence.
4. The entities and relationships in the rewritten text are the same as those in the original text.
5. The implied meaning in the original text must be converted into the literal meaning in the rewritten text.

(b) The Prompt of qwen-plus for Logical Form Generation

Table 6 shows the prompt for inputting the rules of the logical form into qwen-plus and inputting the candidate entities and candidate relationships to generate the corresponding logical form.

Table 6. Guidance for qwen-plus in logical form generation

Rules of the logical form:

The logical form is a structured representation of a natural language question. Taking the S-expression as an example, the logical form usually consists of projections and various operators. The projection operation represents a one-hop query of the triple (s, r, o) on s or o, where (? , r, o) is represented as (JOIN r o) and (s, r, ?) is represented as (JOIN (R r) s). The various operators include: “AND” (AND E1 E2), which obtains the intersection of E1 and E2; “COUNT” (COUNT E1), which counts E1; “ARGMAX” (ARGMAX E1 r), which obtains the maximum literal value after projecting E1 on the r relationship; “ARGMIN” (ARGMIN E1 r), which obtains the minimum literal value after projecting the r relationship on E1; “GT” (GT E1 l), which obtains the part of E1 greater than l; “GE” (GE E1 l), which obtains the part of E1 greater than or equal to l; “LT” (LT E1 l), which obtains the part of E1 less than l; and “LE” (LE E1 l), which obtains the part of E1 less than or equal to l, where E1 or E2 represent a sub-layer logical form.

The following is an example of converting a natural language into a logical form:

The natural language “What is the name of Justin Bieber’s brother?” corresponds to the logical form “(AND (JOIN [people, person, gender] [Male]) (JOIN (R [people, sibling relationship, sibling]) (JOIN (R [people, person, siblings] [Justin Bieber]))).”

According to the above rules and examples, you must generate a logical expression that conforms to the above logical form rules for the following input.
