

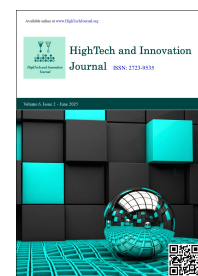


ISSN: 2723-9535

Available online at www.HighTechJournal.org

HighTech and Innovation Journal

Vol. 6, No. 2, June, 2025



Research on RAG-Based Cognitive Large Language Model Training Method for Power Standard Knowledge

Sai Zhang ^{1*}, Xiaoxuan Fan ¹, Bochuan Song ¹, Xiao Liang ¹, Qiang Zhang ¹,
Zhihao Wang ¹, Bo Zhang ²

¹ State Grid Laboratory of Grid Advanced Computing and Applications, China Electric Power Research Institute Co., Ltd., Beijing, China.

² State Grid Wuxi Power Supply Company of Jiangsu Electric Power Co., Ltd., Wuxi, China.

Received 05 December 2024; Revised 21 March 2025; Accepted 09 April 2025; Published 01 June 2025

Abstract

Electrical standards encompass complex technical requirements across multiple disciplines, making their management and application a significant challenge that urgently requires efficient solutions. This paper proposes a knowledge graph retrieval-enhanced training method for large language models (LLMs). By leveraging a pre-trained language model (PLM), highly similar subgraphs are retrieved from the electrical standards knowledge graph. These subgraphs are then parsed into triples using entity linking and semantic reasoning. The triples are converted into natural language text by the LLM, which combines them with the input question to perform reasoning and generate accurate answers. The proposed method addresses the complexity of question answering for electrical standards and offers a novel approach for managing and applying these standards in the field of electrical engineering. Experimental results demonstrate that this approach significantly enhances the model's understanding of electrical standards, enabling it to generate more accurate answers.

Keywords: Electric Standards Knowledge; LLM; RAG; Knowledge Graph; Semantic Reasoning.

1. Introduction

The knowledge system of electric standards is vast and complex, encompassing a wide range in specialized knowledge and technical requirements. Acquiring and updating this knowledge demands significant investments of human and material resources. Furthermore, understanding and applying these standards is challenging, as it involves numerous aspects of the power system, requiring in-depth comprehension and precise application [1]. Additionally, the various forms of standard documents, cross-references, overlaps between standards, and inconsistencies and ambiguities in their interpretation and application pose significant challenges to their effective management and utilization.

The complexity and diversity of knowledge of electric standards underscore the importance of applying a cognitive grand model in this domain. Such a model can intelligently manage and analyze electric standards knowledge, aiding the industry in extracting, organizing, and comprehending relevant information from extensive standard documents. This support is crucial for standard development, implementation, and compliance. By learning and reasoning about the content and requirements of electric power standards, the cognitive grand model can help enterprises and decision-makers better understand and apply these standards [2, 3]. This, in turn, enhances the level of standardization management and promotes the standardized and sustainable development of the electric power industry.

* Corresponding author: zhangsai@geiri.sgcc.com.cn

 <http://dx.doi.org/10.28991/HIJ-2025-06-02-05>

➤ This is an open access article under the CC-BY license (<https://creativecommons.org/licenses/by/4.0/>).

© Authors retain all copyrights.

Although the large language model (LLM) of power has excellent natural language generation capabilities, it often faces some challenges in knowledge-intensive tasks, such as the management of and application of power annotation knowledge. Due to the hallucination problem of the large language model, it often generates factually incorrect content. Therefore, in some specific scenarios, it is necessary to supplement the LLM with external knowledge information, that is, retrieval enhancement, which improves the large language model by integrating information from external reliable knowledge bases, ensuring that the generated output references a large amount of context-rich data and is supplemented by the latest and most relevant available information. For example, Cuconasu et al. [4] proposed RAG extends large language models (LLMs) by adding relevant paragraphs or documents retrieved by an information retrieval (IR) system to the original prompt. For generative AI solutions, RAG is becoming increasingly important, especially in enterprise environments or any field where knowledge is constantly updated and cannot be memorized in an LLM. While retrieval-augmented generation (RAG) has been groundbreaking in enhancing the capabilities of LLMs, certain limitations may affect their effectiveness and applicability, such as difficulties in handling ambiguous queries or those requiring a deeper understanding of the context. Augmenting pre-training with Knowledge Graphs (KGs) can provide LLMs with structured, explicit expressions of factual knowledge about concepts, entities, and their relationships, offering a more nuanced and informed basis for model responses.

The electric standards knowledge graph not only contains extensive factual knowledge about electricity but also helps cognitive macro-models understand the semantic associations between different standards. Compared to vector matching alone, a knowledge graph can perform more complex semantic searches, obtain hidden context information, and provide explicitly structured knowledge representations. This capability enables the generation of more accurate responses and reduces the occurrence of phantom content, significantly improving the model's effectiveness in knowledge-intensive tasks. For example, Prabhon et al. [5] proposed a method called "Knowledge Graph Construction based on Retrieval-Augmented Generation (KGC-RAG)." This approach employs web crawlers to retrieve documents from the Wikipedia pages of target entities and extends the search to related pages. The method leverages LLMs to generate text that supplements and enriches knowledge representation. Their experimental results demonstrate that combining RAG with knowledge graphs significantly enhances performance in knowledge-based question-answering tasks; Xu et al. [6] proposed a novel customer service question-answering method that integrates Retrieval-Augmented Generation (RAG) with knowledge graphs (KG). During the question-answering phase, consumer queries are parsed, and relevant subgraphs are retrieved from the KG to generate answers. Integrating the KG into the method not only improves retrieval accuracy by preserving the structural information of customer service but also enhances answer quality by mitigating the impact of text segmentation. Empirical evaluations based on benchmark datasets show that the experimental results outperform baseline methods by 77.6%.

The paper proposes an algorithm for training LLM based on subgraph retrieval, entity linking, and semantic reasoning within the Electric Standards Knowledge Graph. First, a semantic matching model is trained to extract entity relations within the problem. Second, a subgraph with high similarity is retrieved from the Electric standards Knowledge Graph, filtering the relevant factual triples related to the problem. The next step involves predicting the most probable relational steps and paths for the problem, and sampling reasoning paths consisting of these triples. The results are populated into specific prompts and fed into the LLM along with the problem. Finally, this process converts the data from triples into natural language text and iteratively trains the model, enhancing its capabilities in knowledge-intensive tasks.

The main contributions of this paper are as follows: (1) The paper designs step prediction and relational path prediction methods in subgraph retrieval to retrieve power standard knowledge with high relevance to the problem; (2) The paper develops a method to automatically generate natural language text from extracted subgraphs relevant to the problem; and (3) The paper evaluates our proposed method using various benchmarks. Experimental results demonstrate that the method supplied in the paper outperforms previous knowledge graph enhancement methods across several LLMs.

The organization of this paper is as follows: Section 2 provides an overview of relevant literature, Section 3 presents a detailed description of the design method proposed in this paper, and Section 4 presents the experimental results, while Section 5 concludes the paper.

2. Related Works

Although LLMs are pre-trained based on massive corpora, they are still prone to misleading outputs, factual inaccuracies, and reliance on obsolete information during knowledge-heavy operations. Recent studies have aimed to utilize knowledge graphs to enhance LLMs' capabilities in these tasks [7, 8]. These studies primarily focus on extracting problem-related triples from the knowledge graph and converting them into textual format using various modeling techniques. The textual representations of the triples and questions are then transformed into knowledge-enhancing prompts through predefined templates. These prompts are processed by the question-answering LLM to yield more

dependable responses. Although existing research has demonstrated the success of this method, it has not thoroughly examined how the format of knowledge representation influences the performance of large language models.

Several researchers have proposed more effective graph retrieval methods to provide external information for LLMs, primarily categorized into embedding and semantic parsing approaches. Semantic parsing techniques transform queries into logical representations that can be executed against the knowledge graph [9-11]. These approaches depend on labeled data for supervised learning or are limited to specific domains with a small number of logical predicates [12]. Embedding-based approaches rank entities according to their relevance to the problem, extracting entities from the entire knowledge base or subgraphs [13-16]. While this approach is highly fault-tolerant, the retrieved subgraphs often include many irrelevant entities. Researchers like those behind PullNet [17] and SRN [18] have enhanced retrieval capabilities by training the retriever. However, in these methods, retrieval and reasoning are intertwined, leading to reasoning being performed on only a subset of the retrieved subgraphs. Due to this coupled design, the reasoners in universal hop (UHop), independent recurrent network (IRN), and serial recurrent network (SRN) degrade to simple multilayer perceptrons (MLPs) [19].

Several other research studies have delved into encoding retrieval results, primarily within two key domains. First, researchers have utilized graph neural networks (GNNs) to encode and preserve structural information about subgraphs [20-22]. Their emphasis lies in crafting more sophisticated encoders to derive enriched representations of subgraphs. While graph neural networks excel in encoding graph-structured data, they are constrained by local processing and iterative computation of node representations based on neighboring node features. Second, encoding results using pre-trained language models (PLMs) have gained traction, driven by the emergence of large-scale generative pre-trained language models like BART [23], T5 [24], and GPT [25]. Numerous researchers have begun leveraging these models to generate text from knowledge graphs and treat it as an end-to-end generative task. These endeavors concentrate on modifying model structures and introducing pre-training tasks to enhance structural information extraction. In this study, a pre-trained language model is chosen to encode the retrieval results of the knowledge graph, aligning with the structural characteristics of electric standards knowledge.

3. Methods

Our paper introduces a RAG method for enhancing the cognitive capabilities of LLMs in the context of power standards. Our approach enhances LLM training by incorporating subgraph retrieval and content generation from knowledge graphs. Initially, the user's query content undergoes embedding into a pre-trained language model (PLM), producing a vector representation that encapsulates its semantic essence. Subsequently, the most pertinent nodes in the knowledge graph are identified based on this vector, serving as the local context for the original document or passage. Following this, the retrieved subgraph triples are translated into natural language text, and questions are presented collectively to the large language model for answer generation, as depicted in Figure 1. The main steps encompass subgraph retrieval, text generation, reasoning, and answer synthesis.

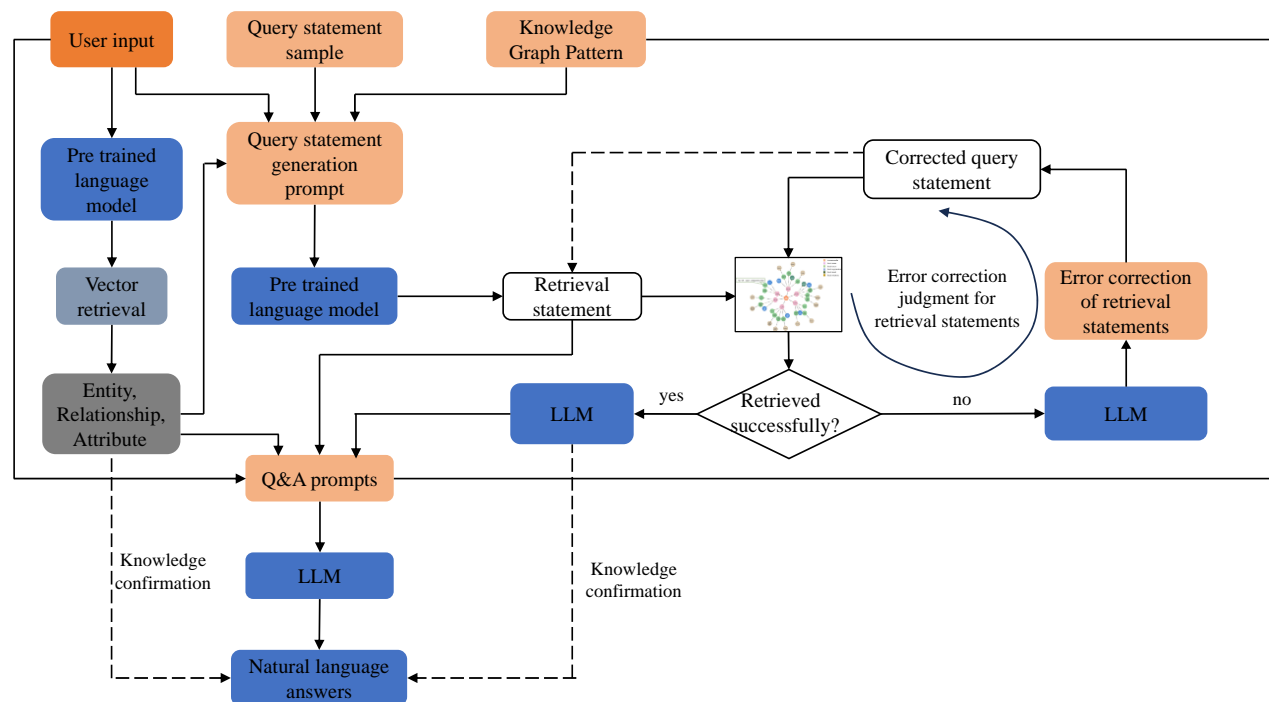


Figure 1. Overall architecture of RAG-based cognitive LLM training for power standards

3.1. Subgraph Retrieval

Subgraph retrieval primarily occurs within the knowledge graph through the computation of semantic similarity between questions and entities, attributes, or relationships. In Figure 2, this process involves identifying relevant entities, relationships, and paths that align with user inputs, followed by transforming the pertinent structured knowledge into natural language inputs using RAG retrieval techniques. These techniques entail predicting the number of retrieval steps, and relational paths, and optimizing multiple tuples. Consequently, the subgraph retrieval process in this paper is structured around step prediction, relation path prediction, and tuple sampling.

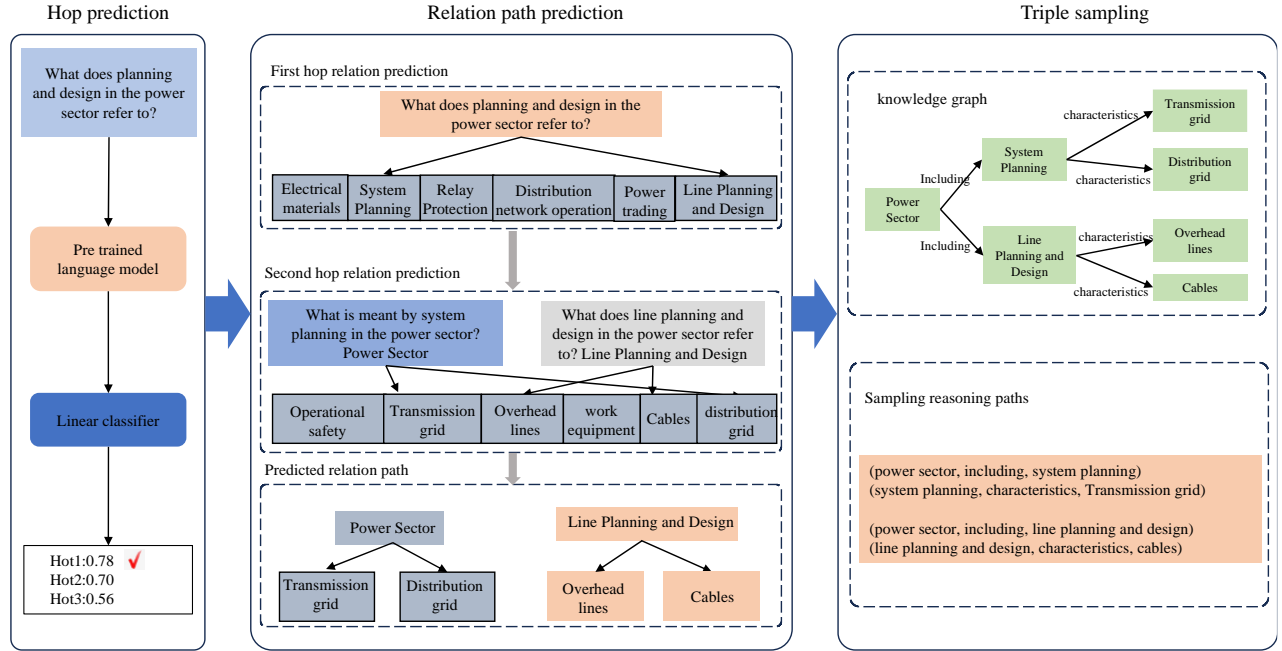


Figure 2. Flowchart of subgraph retrieval for electric standards knowledge graph

(1) Step prediction

Step prediction is used to predict the most likely number of relation-ship jumps in a problem, and to determine how many step relationships to explore in the next step. This paper uses PLM, such as Bert, to transform problem q into a vector q_v , and then uses a linear classifier to predict the probability of each step. Here, the paper first introduces the symbol representations used in some models. d'_{h_c} represents the probability when the number of steps is h_c , and D'_h represents the set of probabilities for all steps. In our paper, h denotes the maximum number of steps predicted by the model, and H denotes the maximum number of steps that can be reached in the graph.

$$q_v = PLM(q) \quad (1)$$

$$d'_{h_c} = P(h_c | q_v), \quad c = 1, 2, \dots, H \quad (2)$$

$$D'_h = [d'_{h_1}, d'_{h_2}, \dots, d'_{h_H}] = Linear(q_v) \quad (3)$$

$$h = \arg \max_{h_c} d'_{h_c}, \quad c = 1, 2, \dots, H \quad (4)$$

In the training data, this paper uses the one-hot form as the Groundtruth D_h for the number of steps, which means the probability of non real answers is 0 and the probability of real answers is 1. Cross entropy is used as the loss function to refine the L_{ce} parameter.

$$D_h = [d_{h_1}, d_{h_2}, \dots, d_{h_H}] \quad (5)$$

$$d_{h_c} = \begin{cases} 0, & h_c \neq h_{groundtruth} \\ 1, & h_c = h_{groundtruth} \end{cases} \quad c = 1, 2, \dots, H \quad (6)$$

$$L_{ce} = -D_h \log D'_h = -\sum_{c=1}^H d_{h_c} \log d'_{h_c} \quad (7)$$

(2) Relationship path prediction

The paper presents a relationship path prediction model similar to the step prediction, with a one-hot relationship probability distribution. Finally, the paper uses the cross-entropy loss function for optimization. In the algorithm for subgraph retrieval, the specific prediction of the number of steps is as follows: if some steps are given, the prediction of the t step relationship path is based on PLM and the relationship path from the previous $t - 1$ steps and the initial problem classification. The first K paths with the highest probability are selected from all the retrieved relationship paths for the next prediction.

Before predicting the relationship path, the problem q is converted into an embedding vector by PLM, and then the first step of prediction is performed. A linear classifier *Linear* is used to calculate the probability distribution $D'_{r,1}$ of all relationships R in the power standard knowledge graph, and the top K largest relationship paths p_1 are sorted by probability size.

$$D'_{r,1} = [d'_{r_1}, d'_{r_2}, \dots, d'_{r_R}] = \text{Linear}(q_v) \quad (8)$$

$$d'_{r_c} = P(r_c | q_v), \quad c = 1, 2, \dots, R \quad (9)$$

At step t , calculate the probability of the i relationship path $p_{t-1,i}$ based on the relationship path from step $t - 1$, as follows,

$$p_{t-1,i} = r_{i,1} | r_{i,2} | \dots | r_{i,t-1}, \quad i = 1, 2, \dots, K^{t-1} \quad (10)$$

Here, $r_{i,t-1}$ represents the relationship between the i relationship path and the step $t - 1$, then the probability distribution of the relationship path $D'_{r,t}$ is calculated using linear classification and PLM based on problem q and the relationship path $p_{t-1,i}$. Where, Q_t represents the input sequence of the model.

$$Q_t = q | r_{i,1} | r_{i,2} | \dots | r_{i,t-1} \quad (11)$$

$$Q_{t,v} = \text{PLM}(Q_t) \quad (12)$$

$$D'_{r,t} = [d'_{r_1}, d'_{r_2}, \dots, d'_{r_R}] = \text{Linear}(Q_{t,v}) \quad (13)$$

$$d'_{r_c} = P(r_c | Q_{t,v}), \quad c = 1, 2, \dots, R \quad (14)$$

Based on the calculation results, the algorithm selects the first K relationship paths as the current relationship path for step t . After h steps of prediction, the algorithm will obtain K^h relationship paths, where the evaluation score of relationship path $p_{t,i}$ is obtained through multiplying the probabilities of all the path's relationships.

$$\text{Score}(p_{t,i}) = \text{Score}(r_{i,1} | r_{i,2} | \dots | r_{i,t}) = \prod_{l=1}^t d'_{r_{i,l}}, \quad i = 1, 2, \dots, K^h \quad (15)$$

(3) Tuple sampling

The sampling of tuples in the paper first uses the relational path algorithm to calculate the evaluation score for each related path, and then sorted in descending order of score size to select the desired path, which is the triplet. In this paper, the selected optimal sub-tuples of tuples are used as relevant knowledge retrieved from the Electric standards knowledge graph (ESKG) as supplementary information for LLM enhancement.

3.2. Text Conversion

(1) Generate fine-tuning data for LLM

Before inputting the relevant tuples retrieved from the subgraph into LLM for answer generation, LLM needs to convert the tuples into a natural language form that LLM can understand as shown in Figure 3. This section generates training data for LLM fine-tuning. Due to the strong professionalism of power standard knowledge, the tuples retrieved from the subgraph are used as input data x for LLM fine-tuning, and then annotated based on electricity specialists. The result is natural language text, which is labeled y in the model calculation. Then, x is filled into the template of prompt p_1 of the model, "Please convert knowledge graph tuples into one or more senses. The KG is {triple form text x }, and the transformed sense is:", where y is used as the output of the model for supervised training.

The training data comes from the annotated knowledge graph of power standards. By setting different questions, subsets of the annotated knowledge graph are extracted, and then the questions and subgraph tuples are filled in $p1$. Finally, the natural language text answers to the questions are generated through the model.

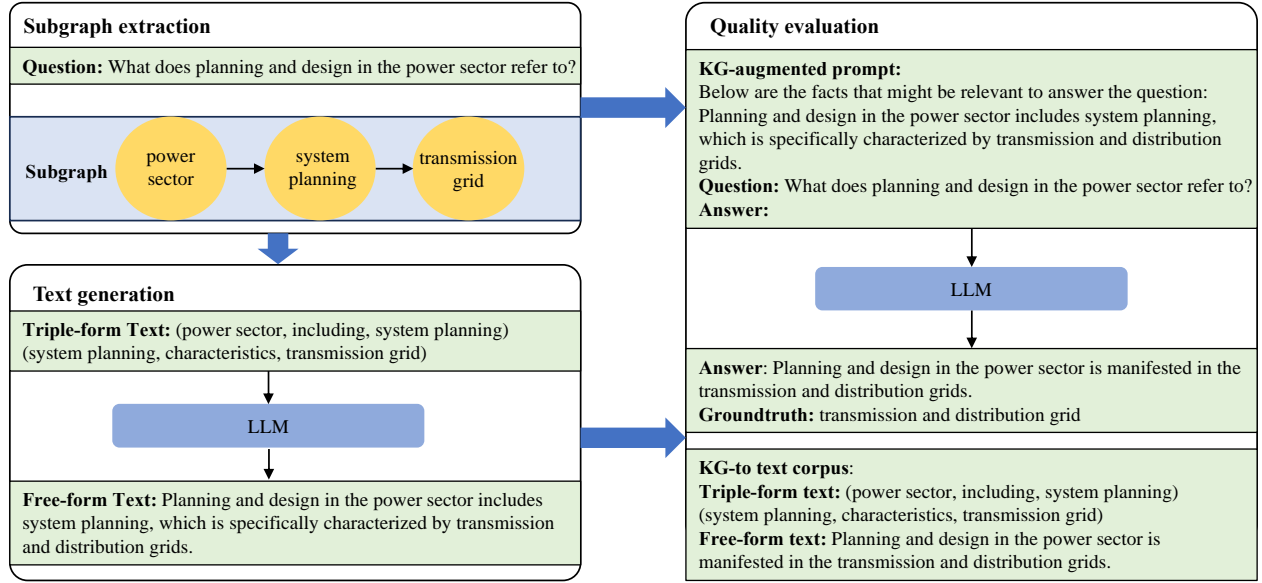


Figure 3. Conversion process diagram from sub entity group to natural language text

(2) Problem answer generation based on LLM

This section uses the finely tuned LLM to generate answers to questions. In the t step of training, given prompt $p1$, the output of Groundtruth is $y = [y_1, y_2, \dots, y_T]$, as well as the vocabulary $[v_1, v_2, \dots, v_V]$. The model will predict the probability distribution $D'_{v,t}$ of all tokens in the current $t - 1$ step based on the correct token sequence in step $[y_1, y_2, \dots, y_{t-1}]$, denoted by d'_{v_c} , which represents the probability of token v_c .

$$D'_{v,t} = [d'_{v_1}, d'_{v_2}, \dots, d'_{v_V}] \quad (16)$$

$$d'_{v_c} = P(v_c | p_1, y_1, y_2, \dots, y_{t-1}), \quad c = 1, 2, \dots, V \quad (17)$$

Similarly, the paper uses the one-hot form to set the true probability distribution $D_{v,t}$ and calculate the loss function using cross entropy.

$$D_{v,t} = [d_{v_1}, d_{v_2}, \dots, d_{v_V}] \quad (18)$$

$$d_{v_c} = \begin{cases} 0, & v_c \neq y_t \\ 1, & v_c = y_t \end{cases}, \quad c = 1, 2, \dots, V \quad (19)$$

$$J_t = -D_{v,t} \log D'_{v,t} = - \sum_{c=1}^V d_{v,c} \log d'_{v,c} \quad (20)$$

$$L_{ce} = \frac{1}{T} \sum_{t=1}^T J_t \quad (21)$$

In the process of converting tuples to natural language text using fine-tuning LLM, firstly, each path is linearized into triplet form text, and then transformed into a prompt through a template. The above prompt is the input of the fine-tuned LLM to obtain the corresponding natural language text. The texts are merged as supplementary knowledge to enhance LLM's ability to answer questions.

3.3. Reasoning and Answer Generation

In the reasoning and answer generation stage of LLM, this paper designs prompt $p2$, which takes the form of "The following is the information associated with the question: {free form text}, where the question is Question: {question} Answer is:". LLM can generate corresponding answers based on the prompt and fill them in the template for output.

4. Experiments

The paper conducted experiments on two datasets and three open-source LLMs. It has been proven that converting triplet text into free-form text can enable LLM to better understand the external knowledge provided and enhance its ability in KGQA. Experimental results demonstrate that, compared to other knowledge representation formats, the Text Conversion method proposed in this paper significantly enhances the accuracy of LLMs in question answering.

4.1. Datasets

The experiments conduct comparative experiments based on the open-source dataset Movie Text Audio QA (MetaQA) and the National Grid Laboratory Dataset (NGLD) dataset developed by our workplace.

MetaQA is a large-scale multi-step knowledge graph question-answering dataset in the film industry. It provides a KG consisting of 9 relationships, 135000 triples, and 43000 entities, and contains over 400000 questions, divided into 1-hop, 2-hop, and 3-hop based on the number of steps taken. Among them, 1-hop originates from the "Wiki_entities" branch of the Facebook MovieQA dataset. Compared to MovieQA, 1-hop removes ambiguous entities in the problem, so the dataset is relatively small. 2-hop and 3-hop also originate from this knowledge base. Among them, 2-hop has 21 types of questions, 3-hop has 15 types of questions, and each type has 10 text templates. Each question is labeled with the answer, head entity, and entity category involved in the inference path. In the experiment, the paper selected MetaQA 3-hop as the experimental dataset and selected 142744 questions (114196 training questions, 14274 development questions, and 14274 testing questions).

The NGLD is Knowledge about China's electric standards provided by a laboratory of State Grid Corporation of China, containing 24326 questions. This paper divides these problems into training set, development sets, and testing sets (18977 training sets, 2560 development sets, and 2789 testing sets). These problems are mainly one-hop or two-hop problems in the field of power labeling knowledge. Each problem has a head entity, an answer, and an optimal relational path. In addition, the dataset also provides a set of over 13000 triples, 10012 entities, and 78 relationships.

4.2. Ablation Experiment

This paper compares and analyzes different knowledge representation formats, and conducts ablation experiments on knowledge representation formats using multiple LLM models on MetaQA and NGLD datasets. The ablation study aims to investigate the impact of different tuple conversion formats and LLM on Knowledge about China's electric standards.

This paper uses Baichuan2-7B-Chat, Baichuan2-13B-Chat chat as the Text Conversion model, and Baichuan2-7B-Chat, Qwen1.5-7B Chat as the problem-solving model. Among them, the comparison of tuple conversion methods is: no tuple conversion, tuple conversion, and existing tuple conversion models.

No Knowledge Questions: Refers to questions being directly inputted into LLM without additional knowledge. This experimental approach is used to explore the effectiveness of knowledge augmentation methods in solving Knowledge about China's electric standards problems and enhancing LLM.

Triple Knowledge: It is a commonly used method of generating triples into text in traditional methods. It first performs disambiguation and deduplication on the retrieved triplets to reduce semantic redundancy and then connects the subject, relationship, and object to provide a simple semantic representation of each triplet.

MVP Knowledge is a text-generation LLM model developed by other researchers. It first conducts supervised text to text format pre-training on 11 different natural language generation tasks on 77 datasets, and then on specific tasks pre trains soft prompts for specific tasks to enhance the model's ability. This paper uses a variant of MVP, MVP data to text, which is pre-trained on annotated text datasets and can perform KG-to-text conversion. Due to MVP not supporting Chinese, this paper did not use this knowledge representation format in the NGLD dataset.

4.3. Comparison of the Experimental Results

To validate the effectiveness of the knowledge graph retrieval algorithm proposed in this paper, comparative experiments were conducted using the following models: Word Frequency - Inverse Document Frequency (TF-IDF): This baseline model retrieves entities or relationships matching query keywords by calculating keyword weights, commonly used in information retrieval; Best Matching 25 (BM25): An improved frequency-based retrieval algorithm that calculates the relevance between documents and queries to retrieve relevant entities or relationships; Cosine Similarity: This method retrieves relevant entities by calculating the cosine similarity between the query terms and the vector representations of entities or relationships in the knowledge graph. The experimental process utilized a 5-fold cross-validation method, and the results are as follows.

From Table 1 and Figure 4, it can be observed that the knowledge graph retrieval algorithm proposed in this paper improves retrieval accuracy by 25% compared to the baseline model and exceeds the Cosine-similarity algorithm by 6% in accuracy. Additionally, in terms of the F1 score, the proposed method outperforms all other models, demonstrating superior performance. Although the proposed method shows slightly lower recall compared to the Cosine-similarity model, the difference is not significant. Finally, in the AUC comparison, the proposed model achieves an AUC value of 0.86, still exhibiting excellent performance. Therefore, it can be concluded that the knowledge graph retrieval method proposed in this paper is highly effective.

Table 1. Comparative experimental results for different knowledge conversion methods on MetaQA and NGLD

Model	ACC	F1	Recall	AUC
Baseline (TF-IDF)	0.76	0.71	0.67	0.69
BM25	0.84	0.75	0.72	0.72
Cosine-similarity	0.89	0.89	0.90	0.80
Our method	0.95	0.92	0.89	0.86

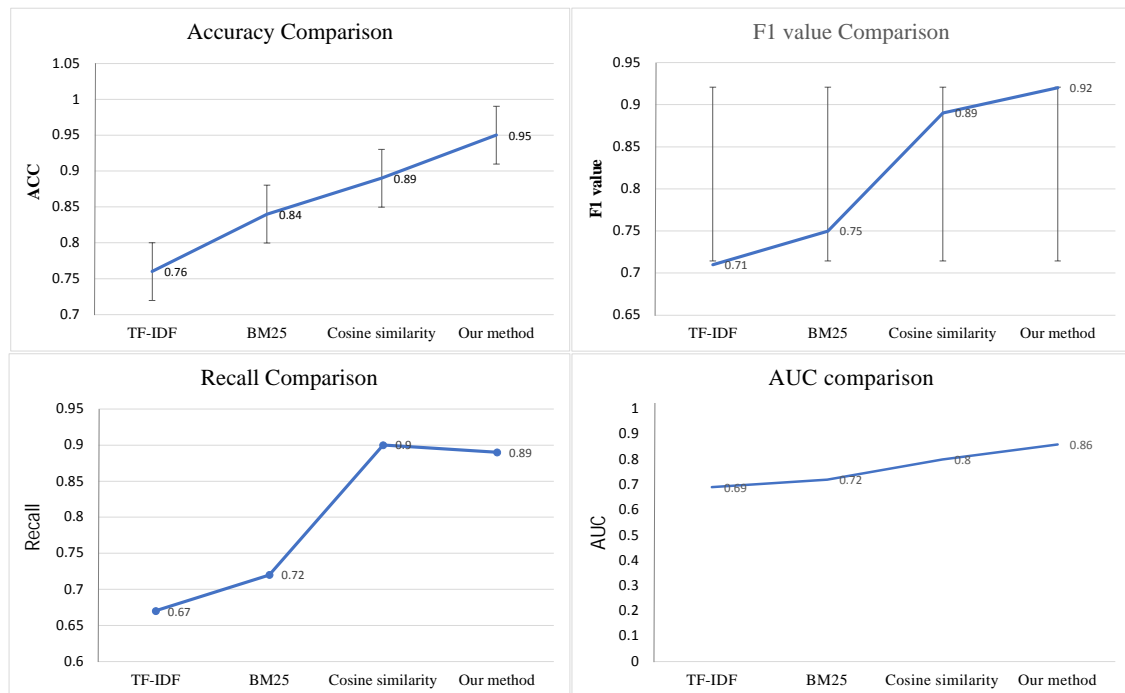


Figure 4. Results of comparative experiments for search algorithms

Since the retrieved subgraphs have differences in their organization, although they have the same score, and hence in their conversion to natural language text, this paper directly compares the performance of LLMs using subgraphs in the experimental part. The section conducted comparative experiments on different tuple conversion methods and different LLMs, and the specific experimental results are shown in Table 2. Among them, the retrieval-enhanced Seq2seq KGC (ReSKGC) [26] is a retrieval-based Seq2seq KGC model that selects semantically relevant triples from the knowledge graph (KG) and uses them as explicit reasoning evidence to guide output generation. The table lists the scores for each model for answering the questions based on the retrieval results.

Table 2. Comparative experimental results for different knowledge conversion methods on MetaQA and NGLD

Knowledge Format	MetaQA			NGLD		
	Baichuan2-7B-Chat	Qwen1.5-7B-Chat	ReSKGC	Baichuan2-7B-Chat	Qwen1.5-7B-Chat	ReSKGC
No Knowledge	30.29	32.78	31.65	19.21	21.57	19.10
Triple Knowledge	94.89	93.43	92.98	87.33	86.67	85.23
MVP Knowledge	88.12	87.64	87.59	-	-	-
Baichuan2-7B-Chat	96.79	95.88	94.67	91.89	92.19	90.19
Baichuan2-13B-Chat-chat	97.12	96.46	92.76	91.23	93.99	89.43

Table 2 presents the overall experimental results of the proposed triple transformation method compared to existing methods on the MetaQA and NGLD datasets. The triple transformation model in this paper is implemented based on Baichuan2-13B-Chat and Baichuan2-7B-Chat, fully utilizing their powerful text generation and contextual understanding capabilities. Additionally, Llama-2-7B-Chat and Qwen1.5-7B-Chat, which have excellent performance, were selected as problem-solving models to verify the generalizability and applicability of the proposed triple transformation method. In the experiments, ReSKGC is used as the comparison model.

The experimental results show that the triple transformation method proposed in this paper significantly outperforms existing triple transformation strategies on both question-answering models, particularly in terms of answer accuracy and output quality. Notably, Qwen1.5-7B-Chat performed exceptionally well on the NGLD dataset, likely due to its large-scale pretraining on Chinese corpora, which makes it better suited for the Chinese-dominated knowledge graph content in the NGLD dataset. Additionally, the outstanding performance of the Baichuan models on the MetaQA dataset also confirms the effectiveness of the proposed method across different corpora and scenarios. Furthermore, the proposed method outperforms the comparison algorithm ReSKGC on both the MetaQA and NGLD datasets, with the proposed method showing an improvement of up to 4.7% in five different scenarios compared to ReSKGC.

Overall, the proposed method of transforming triples into natural language text not only helps large language models (LLMs) better understand supplementary knowledge provided by external sources, but also effectively enhances their reasoning and question-answering capabilities for complex knowledge graphs, significantly improving their performance in knowledge graph question answering (KGQA) tasks.

To further validate the effectiveness of the proposed method, this paper provides a detailed analysis of the experimental results on the NGLD dataset. By establishing two baselines—the No Knowledge Baseline and the Triple Knowledge Baseline—other knowledge formats are compared against these baselines to study the positive and negative impacts of different triple generation methods on the problem-solving model. The No Knowledge Baseline represents scenarios where the model performs reasoning without any external knowledge input, while the Triple Knowledge Baseline represents reasoning based solely on standardized triple knowledge. The comparative analysis shows that the proposed method is more stable and efficient in understanding and utilizing supplementary knowledge, effectively avoiding the model's over-reliance on low-quality input knowledge.

The specific performance of the models is shown in Table 3, which compares the number of questions answered correctly and incorrectly by the models. This further reveals the performance differences of the triple transformation method under different knowledge formats. The results indicate that the optimized triple generation strategy can significantly improve knowledge retrieval and question-answering accuracy, providing an efficient and feasible solution for the knowledge graph question answering domain.

Table 3. Comparison of Text Generation Methods for Different Baselines

Knowledge Format	No Knowledge Baseline		Triple Knowledge Baseline	
	Helpful	No-help	Helpful	No-help
Triple Knowledge	8567	215	-	-
MTL Knowledge	7634	345	387	1294
Baichuan2-7B-Chat	8612	78	476	254
Baichuan2-13B-Chat-chat	8690	43	499	213

The experiments calculated the number of questions that were initially answered incorrectly by the baseline but correctly by other knowledge formats (i.e., helpful), as well as those that were answered correctly by the baseline but incorrectly by other knowledge formats (i.e., no-help). Here, Baichuan2-13B-Chat chat is chosen as the answer model. From Table 2, it can be seen that this paper uses Baichuan2-7B-Chat and Baichuan2-13B-Chat chat as tuple conversion methods. The problem-solving model can understand the contextual semantics well, and even provide useful answer information for the problem in the presence of incorrect contextual information.

5. Conclusion

The paper presents a RAG framework for LLM subgraph retrieval based on knowledge graphs. Initially, optimization methods are designed for step prediction and relationship path prediction in subgraph retrieval to extract subgraph tuples relevant to the problem. These tuples are then transformed into natural language through fine-tuning an LLM. A prompt template is used to represent both the problem and the tuple as natural language text, which is input into the LLM to generate the answer. The fine-tuning data for the LLM comes from manually annotated power standard data. Experimental results show that the proposed framework outperforms previous RAG methods by a significant margin. Furthermore, compared to traditional knowledge graph-based RAG methods, this framework generates natural language texts that are more comprehensible for the LLM. Finally, experiments confirm that the LLM derived from this framework delivers relatively accurate responses when answering questions related to power standard knowledge.

This paper acknowledges certain limitations in the knowledge graph retrieval enhancement method based on large language models. Existing approaches primarily rely on predefined domain ontologies and rules, which restrict their flexibility in handling cross-domain and multi-dimensional knowledge. As a result, these models struggle to address implicit relationships in complex power systems. Additionally, current retrieval systems face challenges when handling semantically ambiguous queries, especially when confronted with incomplete or unclear power standard data, leading to inaccurate responses. To address these issues, future research should focus on integrating deep semantic understanding with reasoning capabilities, particularly enhancing the model's ability to understand the knowledge structure within the power domain. Furthermore, the use of multimodal data and dynamic knowledge updates will help improve the timeliness and accuracy of both knowledge graphs and retrieval results.

6. Declarations

6.1. Author Contributions

Conceptualization, S.Z., X.F., and B.S.; methodology, S.Z. and X.F.; validation, X.L., Q.Z., Z.W., and B.Z.; formal analysis, S.Z., X.F., B.S., and X.L.; writing—original draft preparation, S.Z., X.F., B.S., X.L., Q.Z., Z.W., and B.Z.; writing—review and editing, S.Z., X.F., B.S., X.L., Q.Z., Z.W., and B.Z. All authors have read and agreed to the published version of the manuscript.

6.2. Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6.3. Funding and Acknowledgments

This research is supported by China Electric Power Research Institute Co., Ltd. “Research on the technology of standard knowledge aided generation based on large language model.” (Grand No. 5700-202358832A-4-2-KJ).

6.4. Institutional Review Board Statement

Not applicable.

6.5. Informed Consent Statement

Not applicable.

6.6. Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

7. References

- [1] Almughrabi, A., & Hiary, H. (2024). Hand-drawn Electric Circuit Diagrams Recognition using Deep Learning. 2024 28th International Conference on Information Technology, IT 2024, 1–4. doi:10.1109/IT61232.2024.10475731.
- [2] Fang, J., Meng, Z., & Macdonald, C. (2024). Reano: Optimising retrieval-augmented reader models through knowledge graph generation. Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, Volume 1: Long Papers, 2094–2112.
- [3] Sen, P., Mavadia, S., & Saffari, A. (2023). Knowledge Graph-augmented Language Models for Complex Question Answering. Proceedings of the Annual Meeting of the Association for Computational Linguistics, 1–8. doi:10.18653/v1/2023.nlrse-1.1.
- [4] Cuconasu, F., Trappolini, G., Siciliano, F., Filice, S., Campagnano, C., Maarek, Y., Tonello, N., & Silvestri, F. (2024). The Power of Noise: Redefining Retrieval for RAG Systems. SIGIR 2024 - Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 719–729. doi:10.1145/3626772.3657834.
- [5] Prabhong, T., Kertkeidkachorn, N., & Trongratsameethong, A. (2024). KGC-RAG: Knowledge Graph Construction from Large Language Model Using Retrieval-Augmented Generation. CEUR Workshop Proceedings, 3853.
- [6] Xu, Z., Cruz, M. J., Guevara, M., Wang, T., Deshpande, M., Wang, X., & Li, Z. (2024). Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering. SIGIR 2024 - Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval, 2905–2909. doi:10.1145/3626772.3661370.
- [7] Wang, J., Wen, Z., Li, X., Guo, Z., Yang, J., & Liu, Z. (2024). Pair Then Relation: Pair-Net for Panoptic Scene Graph Generation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 46(12), 10452–10465. doi:10.1109/TPAMI.2024.3442301.

- [8] Yu, F., Lu, C., Zhou, J., & Yin, L. (2024). Mathematical model and knowledge-based iterated greedy algorithm for distributed assembly hybrid flow shop scheduling problem with dual-resource constraints. *Expert Systems with Applications*, 239, 122434. doi:10.1016/j.eswa.2023.122434.
- [9] Pramanik, S., Alabi, J., Roy, R. S., & Weikum, G. (2024). UNIQORN: Unified question answering over RDF knowledge graphs and natural language text. *Journal of Web Semantics*, 83, 100833. doi:10.1016/j.websem.2024.100833.
- [10] He, G., Lan, Y., Jiang, J., Zhao, W. X., & Wen, J. R. (2021). Improving Multi-hop Knowledge Base Question Answering by Learning Intermediate Supervision Signals. *WSDM 2021 - Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 553–561. doi:10.1145/3437963.3441753.
- [11] Roh, J., Kim, M., & Bae, K. (2024). Towards a small language model powered chain-of-reasoning for open-domain question answering. *ETRI Journal*, 46(1), 11–21. doi:10.4218/etrij.2023-0355.
- [12] Joshi, P., Gupta, A., Kumar, P., & Sisodia, M. (2024). Robust Multi Model RAG Pipeline for Documents Containing Text, Table & Images. *Proceedings of the 3rd International Conference on Applied Artificial Intelligence and Computing, ICAAIC 2024*, 993–999. doi:10.1109/ICAAIC60222.2024.10574972.
- [13] Diaz-Pace, J. A., Tommasel, A., & Vazquez, H. C. (2024). The JavaScript Package Selection Task: A Comparative Experiment Using an LLM-based Approach. *CLEI Electronic Journal (CLEIeJ)*, 27(2), 19. doi:10.19153/cleiej.27.2.4.
- [14] Shao, Y., Li, H., Gu, X., Yin, H., Li, Y., Miao, X., Zhang, W., Cui, B., & Chen, L. (2024). Distributed Graph Neural Network Training: A Survey. *ACM Computing Surveys*, 56(8), 1–39. doi:10.1145/3648358.
- [15] Kundu, S., & Aakur, S. N. (2023). IS-GGT: Iterative Scene Graph Generation with Generative Transformers. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2023-June*, 6292–6301. doi:10.1109/CVPR52729.2023.00609.
- [16] Meng, Y., Xiong, C., Bajaj, P., Tiwary, S., Bennett, P., Han, J., & Song, X. (2021). COCO-LM: Correcting and Contrasting Text Sequences for Language Model Pretraining. *Advances in Neural Information Processing Systems*, 28, 23102–23114.
- [17] Alapati, P. R., Lawrance, J. C., Sambath, P., Murugan, R., Rengarajan, M., Raj, I. I., & Bala, B. K. (2024, April). Cross-Lingual Transfer Learning in NLP: Enhancing English Language Learning for Non-Native Speakers. *10th International Conference on Communication and Signal Processing (ICCSP)*, 1042–1047. doi:10.1109/ICCSP60870.2024.10544031.
- [18] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). Language models are unsupervised multitask learners. *OpenAI Blog*, 1–24.
- [19] Ribeiro, E., Ribeiro, R., & De Matos, D. M. (2019). Deep dialog act recognition using multiple token, segment, and context information representations. *Journal of Artificial Intelligence Research*, 66, 861–899. doi:10.1613/jair.1.11594.
- [20] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., ... & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in neural information processing systems*, 33, 9459–9474.
- [21] Meng, G., Tariq, M., Jain, S., Elmetwaly, S., & Schlick, T. (2020). RAG-Web: RNA structure prediction/design using RNA-As-Graphs. *Bioinformatics*, 36(2), 647–648. doi:10.1093/bioinformatics/btz611.
- [22] Datta, V. D., Ganesh, S., Haas, R. E., & Talukder, A. K. (2023). GREAT AI in Medical Appropriateness and Value-Based-Care. *International Conference on Big Data Analytics*, Springer Nature, Cham, Switzerland. doi:10.1007/978-3-031-49601-1_2.
- [23] Jain, S., Tao, Y., & Schlick, T. (2020). Inverse folding with RNA-As-Graphs produces a large pool of candidate sequences with target topologies. *Journal of Structural Biology*, 209(3), 107438. doi:10.1016/j.jsb.2019.107438.
- [24] Wadhwa, S., DeYoung, J., Nye, B., Amir, S., & Wallace, B. C. (2023). Jointly Extracting Interventions, Outcomes, and Findings from RCT Reports with LLMs. *Proceedings of Machine Learning Research*, 219, 754–771.
- [25] Ahmed, T., & Devanbu, P. (2022). Few-shot training LLMs for project-specific code-summarization. *ACM International Conference Proceeding Series*, 1–5. doi:10.1145/3551349.3559555.
- [26] Yu, D., & Yang, Y. (2023). Retrieval-Enhanced Generative Model for Large-Scale Knowledge Graph Completion. *SIGIR 2023 - Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2334–2338. doi:10.1145/3539618.3592052.