Article | Received 18 July 2024; Accepted 19 September 2024; Published 25 September 2024
https://doi.org/10.55092/sc20240012

Framework and construction methodology of underground engineering domain knowledge large language model: UndergrGPT

Zhenhao Xu[1,2,*], Zhaoyang Wang[1,2], Pengjie Ren[3], Xiao Zhang[3] and Tianhao Li[1,2]

[1]State Key Laboratory for Tunnel Engineering, Jinan, China
[2]School of Qilu Transportation, Shandong University, Jinan, China
[3]School of Computer Science and Technology, Shandong University, Qingdao, China

* Correspondence author; E-mail: zhenhao_xu@sdu.edu.cn.

Abstract:Large Language Models (LLMs) have achieved tremendous success in natural language processing tasks. However, in vertical domains such as underground engineering, the lack of large-scale structured corpora and the requirement for high-accuracy responses significantly increase the difficulty of constructing domain knowledge LLMs. Simultaneously, these challenges amplify the inherent flaws of models, such as machine hallucination and outdated content. To address the aforementioned challenges, firstly, we proposed a framework for constructing underground engineering domain knowledge LLMs, which includes human-machine collaborative knowledge extraction, parameter efficient fine-tuning, and retrieval-augmented generation. The framework aims to provide a universal construction paradigm for underground engineering domain knowledge LLMs. Secondly, within the proposed framework, we established an underground engineering domain knowledge database (UndergrData), introduced a domain knowledge extraction approach based on a few-shot prompt learning data augmentation strategy, resulting in the formation of the training dataset; we applied an underground engineering domain knowledge LLMs fine-tuning method under low-resource and constructed the first underground engineering domain knowledge LLM base (UndergrGPT); we adopted the approach of underground engineering domain knowledge retrieval-augmented generation, enabling high-quality response generation. Finally, compared with general LLMs, UndergrGPT effectively reduced hallucinations, demonstrating a significant advantage in response accuracy. It's worth noting that the proposed framework can be extended to other vertical domains.

Keywords: domain knowledge large language model; domain knowledge database; construction framework; knowledge extraction; parameter efficient fine-tuning; retrieval-augmented generation

1. Introduction

Underground engineering construction faces severe risks from adverse geologies, leading to potential hazards like water or mud inflows, rockburst, collapses, and deformations. These events can cause construction delay and economic loss [1–3]. Providing timely, scientific and effective decision-making solutions is crucial for tunnel disaster prevention and control when encountering unfavorable geologies during construction.

Traditional underground engineering disaster prevention and control has historically relied heavily on manual intervention, requiring skilled personnel to be actively involved

in monitoring, assessing, and responding to potential hazards in real-time situations. However, with advancements in multi-source information perception and data collection capabilities, the modalities and quantity of underground engineering data are gradually increasing. The multi-source and multimodal data pose a significant challenge for manual processing, hindering the efficient derivation of scientifically rigorous decision-making solutions. Recent advances in computing and information technology [4,5], particularly in natural language processing, have led to the emergence of intelligent question-answering system [6–8]. Intelligent question-answering systems offer new pathways for decision-making in underground engineering disaster prevention and control.

Knowledge graph-based question answering (KB-QA), as a vital component of intelligent question-answering systems, has wide application scenarios in various stages of underground engineering [9]. Knowledge graph is a type of semantic network that serves as a knowledge repository, where nodes represent entities and edges represent the relationships between entities. During question processing, entity recognition techniques are used to identify the main entities mentioned in the user's query. These entities are then matched with other relevant entities in the knowledge base, facilitating the retrieval of answers [10,11]. However, KB-QA faces various challenges such as the need for large amounts of data, the difficulty in constructing knowledge graph network structures, poor readability and lack of specialization in response content.

The emergence of LLMs provides new solutions for intelligent question-answering systems in underground engineering. LLMs excel in tasks such as instruction comprehension and generation, exhibiting strong generalization capabilities in natural language processing (NLP) tasks. The advent of open-source pretrained models propels the development of LLMs in vertical domains such as transportation, medicine, energy, finance, and agriculture [12–16]. However, in field of underground engineering, the contradiction between demand of high-accuracy responses and insufficient availability of structured training data exacerbates the inherent flaws of LLMs. The construction of underground engineering LLMs faces three main challenges: extraction of high-quality domain knowledge, efficient utilization of limited computational resources and generation of highly accurate responses.

Extraction of high-quality domain knowledge. Knowledge graph was a crucial method and tool for knowledge extraction, storage, and querying in recent times [17–20]. Knowledge graphs extract triples from sentences to describe specific knowledge, and store it in the form of directed graphs. There has been significant research into knowledge graphs within the field of underground engineering [21]. However, current triple extraction models for knowledge graphs mostly rely on supervised learning approaches, which require abundant high-quality annotated data to enhance the accuracy and generalization of the models. The general LLMs can utilize in-context learning, including zero/one/few-shot prompt learning, to tackle new tasks. In-context learning offers high flexibility and requires low data set quantities. General LLMs show excellent performance in in-context learning, achieving comparable or even superior performance to fine-tuned pretrained models on certain tasks [22]. Text generation methods based on zero/one/few-shot prompt learning offer new insights into extracting domain knowledge.

Efficient utilization of limited computational resources. Domain knowledge LLMs are primarily achieved through parameter fine-tuning on domain-specific datasets. Full-parameter fine-tuning involves updating all parameters of the pre-trained model, which poses challenges to LLMs that have enormous parameter sizes due to the rapid data updates [23]. Recently proposed parameter-efficient fine-tuning methods such as Low-Rank Adaptation (LoRA) offer new solutions to fine-tuning large models with limited resources [24]. LoRA allows to train some dense layers in a neural network indirectly by optimizing rank decomposition matrices of the dense layers' change during adaptation instead, while keeping the pre-trained weights frozen. LoRA does not update the original parameter matrix, thus preserving the

model's generalization ability. Meanwhile, the number of parameters that need to be updated is reduced to 1% or even less of the original, which offers the potential for frequent updates during training.

High-accuracy response generation. It's been proved that LLMs extract knowledge from corpus data to form parameterized memories, resulting in accomplishing general domain natural language processing tasks without relying on external databases [25]. In vertical domains like underground engineering, the strict accuracy requirements for tasks lead to two challenges for domain knowledge LLMs. First, the amount of data generated is enormous. Once the training of large language models is completed, the content quickly becomes outdated. Additionally, LLMs have inherent flaws like machine hallucination, making them prone to generating inaccurate responses. To address the aforementioned challenges, a response generation approach named retrieval-augmented generation (RAG) has been proposed. RAG combines parameterized memory (LLMs) and non-parameterized memory (database retrieval). By establishing an external database and retrieving relevant content related to the query, RAG guides the large model to generate answers in a one-shot or few-shot manner. RAG effectively expands real-time data and suppresses the machine hallucination of LLM, thus improving the accuracy of responses [26,27].

To address the aforementioned three issues, a framework for building underground engineering domain knowledge LLMs was proposed, which offers a LLM universal paradigm for underground engineering. Within the general framework, we first established an underground engineering knowledge database (UndergrData) through corpus data collection. Then, we introduced a domain knowledge extraction method based on few-shot prompt learning data augmentation strategy, resulting in the formation of a model training dataset. Besides, we constructed the first LLM for underground engineering (UndergrGPT), achieving fine-tuning training of LLM with low resources. Further, we proposed a retrieval-augmented text generation method for underground engineering domain knowledge LLM. The proposed method mitigates the machine hallucination and enhances the accuracy of model responses effectively.

2. Framework and construction methodology of UndergrGPT

As shown in Figure 1, the construction framework of underground engineering domain knowledge LLM (UndergrGPT) consists of three parts: human-machine collaborative knowledge extraction, parameter efficient fine-tuning and retrieval-augmented generation.

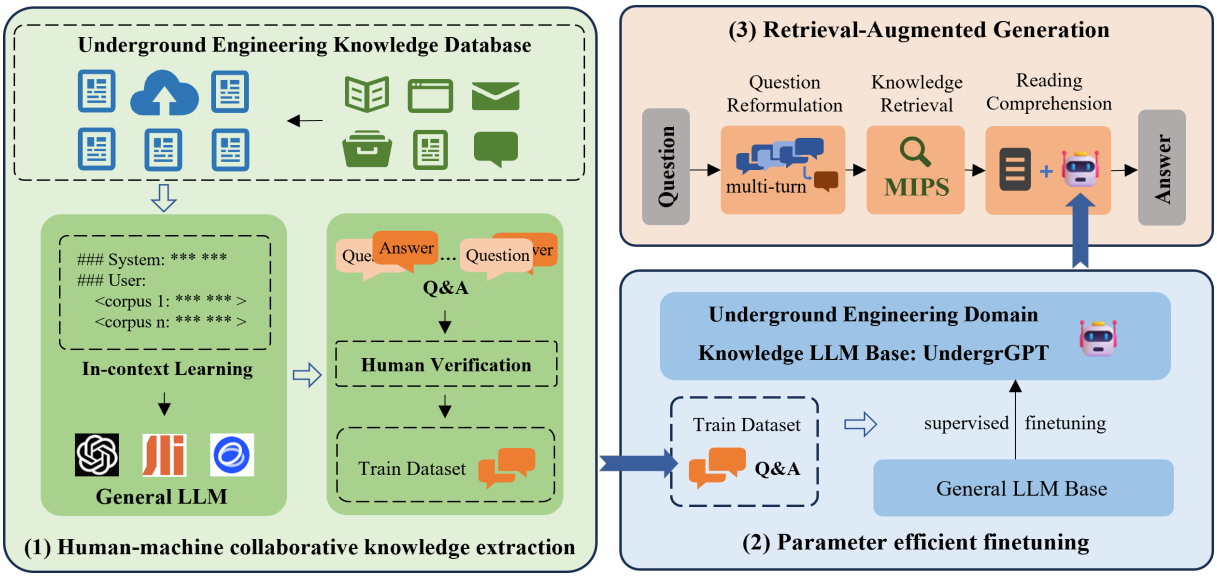

Figure 1. UndergrGPT construction framework.

2.1. *Human-machine collaborative knowledge extraction*

High-quality text annotation data is a crucial factor for the performance of LLMs. In field of underground engineering, there is a vast corpus of literature, standard specifications, and geological reports, all containing rich domain knowledge. However, a systematic knowledge base and large model training data have not yet been established. The lack of efficient and accurate knowledge extraction methods is the primary reason. Following the widespread application of general LLMs, data augmentation strategy based on in-context learning offers new method for knowledge extraction and dataset construction.

When performing single task, LLM can be expressed in a probabilistic framework as estimating a conditional distribution p (output | input). Since LLM should be able to perform many different tasks, even for the same input, it should condition not only on the input but also on the task to be performed p (output | input, task). Task conditioning is often implemented at an architectural level, such as the task specific encoders and decoders [28] or at an algorithmic level, such as the inner and outer loop optimization framework [29]. At the current stage, a more flexible approach (in-context learning) to implementing multitasking has been proposed, which includes zero/one/few-shot prompt learning. In-context learning refers to accomplishing multitasking by incorporating examples into the input, without updating the parameters of the large model. Such as, the execution of a translation task can represent the input as a sequence (translate to French, English text, French text), while the reading comprehension task can represent the input example as a sequence (answer the question, document, question, answer).

In-context learning has demonstrated powerful capabilities in multitasking with LLMs, achieving or even surpassing the performance of fine-tuned pretrained models in certain tasks. Following the widespread application of general LLMs, standardized in-context learning formats have successively emerged across various models. We proposed a method for underground engineering knowledge extraction through one/few-shot prompt learning, achieving automated knowledge extraction with minimal human intervention. Figure 2 illustrates the extraction process for question-answer pairs.

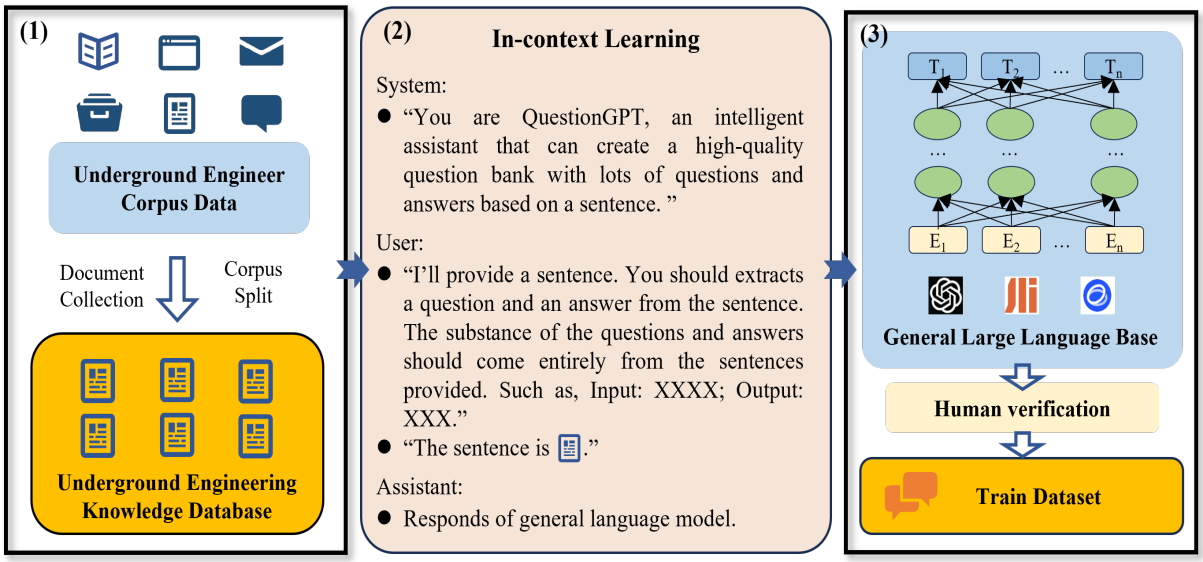

Figure 2. Underground engineering domain knowledge extraction.

Knowledge Database Construction. Gather and decode training corpus data, then cleanse it to create a knowledge base. Segments of the corpus are divided based on the granularity of the knowledge extraction content, preparing it for the extraction process.

In-context Learning. Assign the role of data annotator to GPT, with the task of extracting question-answer pairs from sentences. Provide examples of knowledge extraction and input

them into the model.

Train Dataset Construction. Provide the sentences that need to be extracted, input them into the model and wait for the model's response. After manual correction, form the training dataset.

It is important to note that there are significant differences in language understanding capabilities among different models. To ensure data quality, in addition to the corpus data that needs to be extracted, it is advisable to use language expressions that the model excels in. Additionally, putting too many examples into the prompt is not a wise approach. It is sufficient to select typical question-answer pairs for the current task to include in few-shot learning, with the specific number and type adjusted according to the task. Moreover, manually tuning the results of knowledge extraction is crucial for several reasons. One primary reason is that general LLMs do not possess exceptional capabilities in handling underground engineering domain knowledge. Relying solely on general LLMs for knowledge extraction would confine the capabilities of our trained model to those of the general LLMs. Another reason is that using data generated by the model for training, especially after multiple iterations, can potentially lead to model collapse.

2.2. *Parameter efficient fine-tuning*

LLM fine-tuning includes full fine-tuning and parameter-efficient fine-tuning. With hundreds of millions of parameters of large models, full fine-tuning requires updating all parameters, which consumes substantial computational power and presents a high barrier for the application of underground engineering LLM. In recent years, parameter-efficient fine-tuning methods represented by LoRA have been proposed. These methods, by freezing the original parameter matrices and introducing low-rank matrices, reduce the parameters that need updating to 1% or even less of the original, significantly lowering computational resource consumption while ensuring training effectiveness.

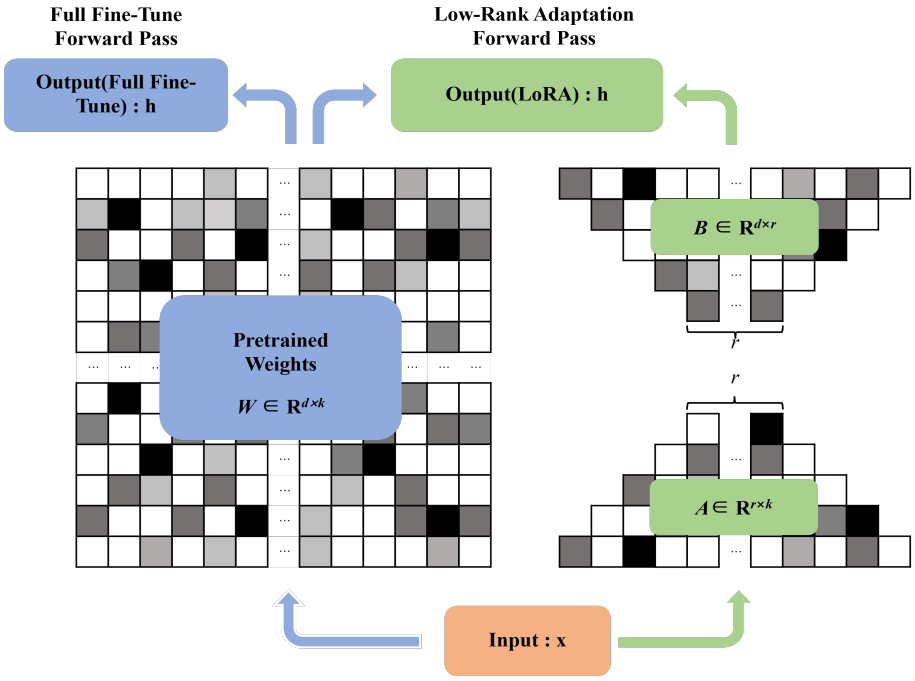

Figure 3. Low-Rank adaptation.

The principle of LoRA is shown in Figure 3. A neural network contains many dense layers which perform matrix multiplication. For a pre-trained language model weight matrix $W \in R^{d\times k}$ its update is constrained by representing the latter with a low-rank decomposition $W + \Delta W = W + BA$, where $B \in R^{d\times r}$, $A \in R^{r\times k}$, and the rank r « min(d, k). During training, W

is frozen and does not receive gradient updates, while A and B contain trainable parameters. Note both W and $\Delta W = BA$ are multiplied with the same input, and their respective output vectors are summed coordinate-wise. For $h = Wx$, our modified forward pass yields [24]:

$$h = Wx + \Delta Wx = Wx + BAx \tag{1}$$

2.3. *Retrieval-Augmented generation*

LLMs can extract knowledge from training data to form parameter memories, allowing them to generate independent responses without relying on external corpus databases. However, parameter memories cannot be easily modified. Aside from retraining, it's challenging to incorporate new content, which means the content can become outdated soon after training is completed. Additionally, this approach has the flaw of producing machine hallucinations. Retrieval-Augmented Generation (RAG) approach facilitates the formation of non-parametric memories by retrieving information from external databases. Non-parametric memories are integrated with parametric memories to establish a hybrid model, which effectively mitigates the limitations related to content obsolescence and hallucinations inherent in LLMs [26].

RAG can be defined that pre-trained, parametric-memory generation models (LLM) is endowed with a non-parametric memory through a general-purpose fine-tuning approach. As shown in Figure 4, RAG model is built where the non-parametric memory is a dense vector index of domain knowledge corpus, accessed with a pre-trained neural retriever and the parametric memory is a pre-trained seq2aeq generator. The retriever provides latent documents conditioned on the input, and the seq2aeq generator then conditions on these latent documents together with the input to generate the output. RAG model uses the input sequence x to retrieve text documents z and use them as additional context when generating the target sequence y. It leverages two components: (a) a retriever $p_\eta(z|x)$ with parameters that returns (top-K truncated) distributions over text passages given a query x and (b) a generator $p(y_i|x,z,y_{1:i-1})$ parametrized by that generates a token based on the context of the previous $i-1$ tokens $y_{1:i-1}$, the original input x and a retrieved passage z [26].

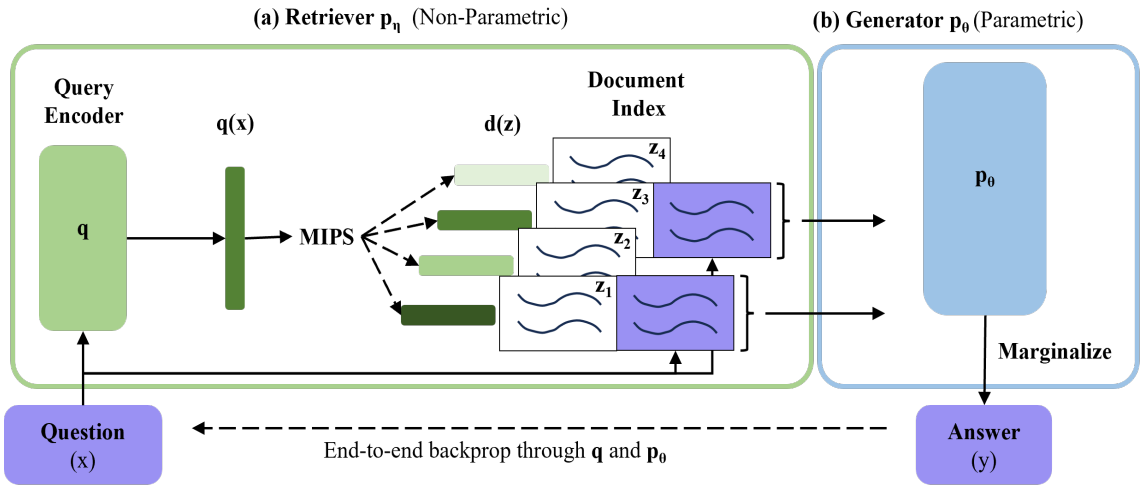

Figure 4. Overview of Retrieval-Augmented Generation (RAG).

We proposed a method for underground engineering domain knowledge LLM text generation based on RAG (Figure 5). The main process is divided into three parts: question reformulation, knowledge retrieval, and reading comprehension.

Question reformulation. In multi-turn dialogues, there are common references, pronouns, ellipses, and shifts in user focus. Using a single turn of dialogue for retrieval may result in inaccurate content retrieval. It is necessary to input the content of multiple rounds dialogue into the general LLM simultaneously to perform operations such as pronoun

resolution and semantic disambiguation, and then carry out retrieval operations with the reconstructed question.

Knowledge retrieval. Embed the reconstructed question to generate a question vector, and simultaneously issue a query request for retrieval in an external database. The Retriever performs a Maximum Inner Product Search (MIPS) between the question vector and the vectors of corpus segments, selecting one or more corpus segments that have the highest degree of match with the question vector. The number of segments selected can either be fixed or chosen based on their matching score.

Reading comprehension. From the indexed corpus segments, generate prompts for the LLM that are combined with the question. An example of such a prompt might be, 'Please answer the following question based on the prompt. Question: XX. Prompt: XX.' The format guides the LLM to generate a response using a one/few-shot approach.

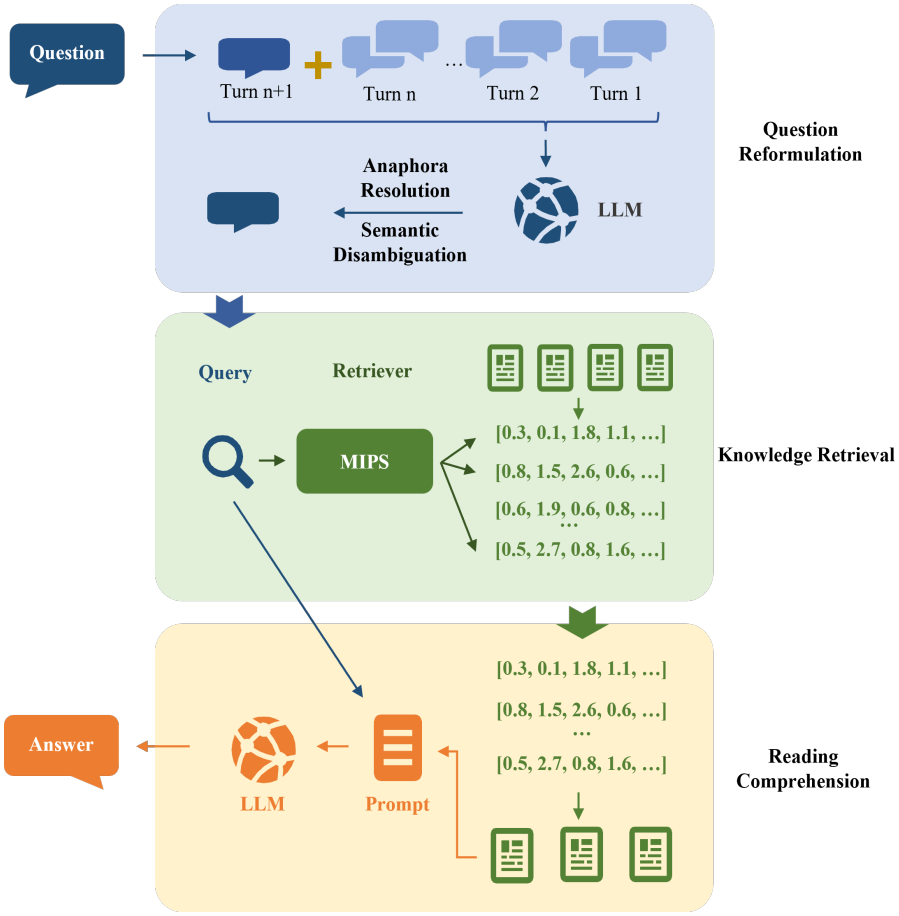

Figure 5. High-quality response generation

3. Case study and verification

To verify the validity of UndergrGPT construction framework, we started with further pre-training ChatGLM, a Chinese foundation model. LLM fine-tune corpus datasets are derived from underground engineering literature.

3.1. Domain knowledge extraction

Currently, the publicly available knowledge databases and training datasets in underground engineering are not comprehensive. We attempted to construct a knowledge database (UndergrData) and training dataset from scratch. The sources of the corpus data include literature, books, standards, and geological reports, with approximately 500 million characters

organized to date.

Domain knowledge extraction and training dataset construction initially started with literature. The search keywords for the literature include underground engineering, tunnel engineering, tunnel boring machines, advance geological prediction, unfavorable geological conditions in tunnels, and tunnel disasters. Extracts are taken from the abstracts of the literature. Knowledge extraction is conducted using the method proposed above with ChatGPT. ChatGPT has a better understanding of English, so except for the content to be extracted, all other expressions are in English. The process of domain knowledge extraction is illustrated in Figure 6.

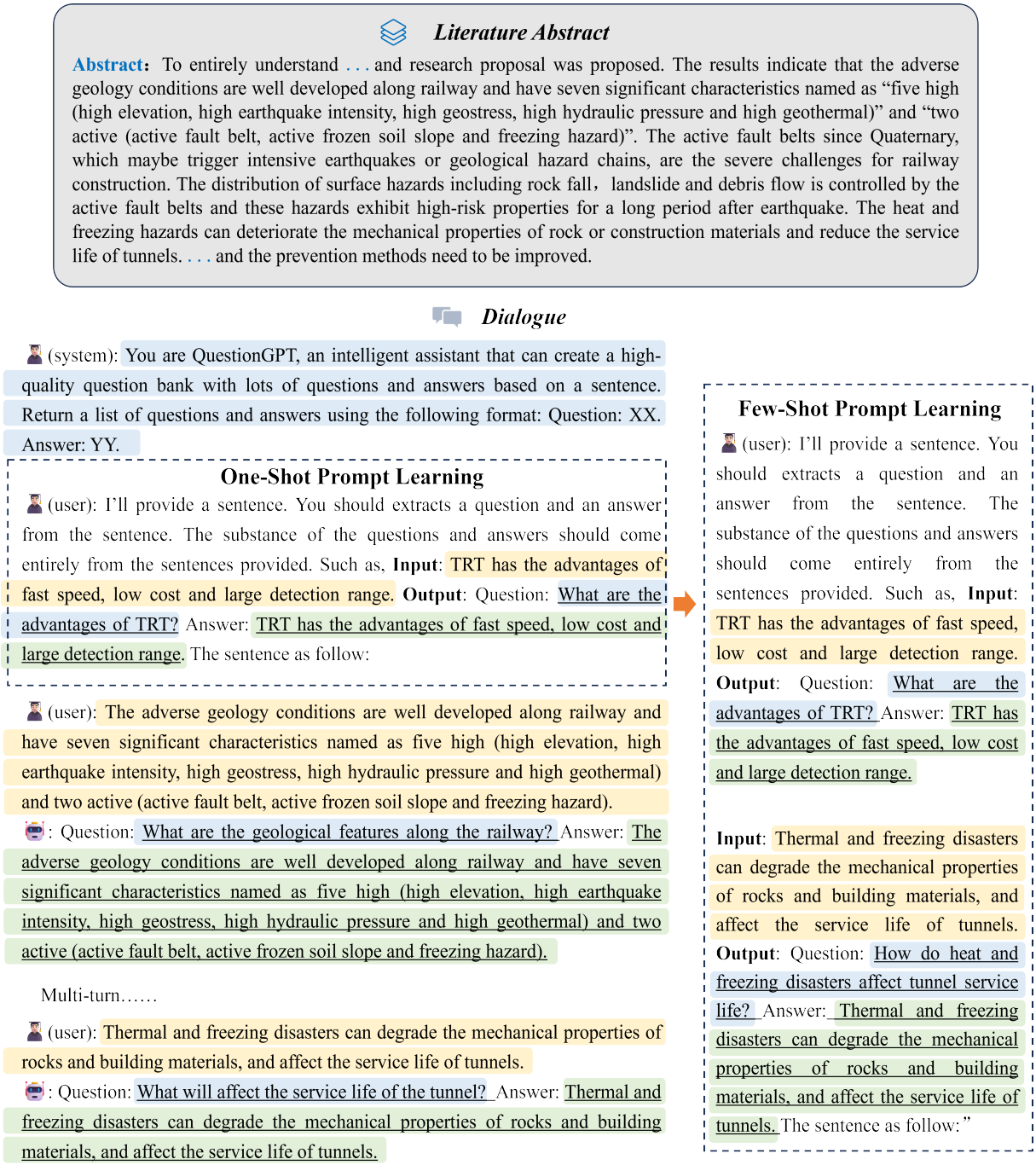

Figure 6. Domain knowledge extraction and dataset construction.

Prepare the corpus to be extracted. "To entirely understand……and the prevention methods need to be improved.".

Assign GPT the role of a data annotator, and specify that the annotator's job is to extract question-answer pairs from sentences, in the required output format. Such as "You are QuestionGPT, an intelligent assistant that can create a high-quality question bank with lots of questions and answers based on a sentence. Return a list of questions and answers using the following format: Question: XX. Answer: YY".

Provide a detailed description of the requirements for extracting question-answer pairs, including an example of domain knowledge extraction in a one-shot format. Such as "I'll provide a sentence. You should extract a question and an answer from the sentence. The substance of the questions and answers should come entirely from the sentences provided. Such as, Input: TRT has the advantages of fast speed, low cost and large detection range. Output: Question: What are the advantages of TRT? Answer: TRT has the advantages of fast speed, low cost and large detection range. The sentence is as follow:".

List the sentences for extraction one by one. Such as "The adverse geologies are well developed along railway and there are seven significant characteristics: five high (high elevation, high earthquake intensity, high geostress, high hydraulic pressure and high geothermal) and two active (active fault belt, active frozen soil slope and freezing hazard)". Wait for the model's response.

General LLMs possess strong text understanding and generation capabilities, and the extraction results generally meet the requirements for training domain-specific knowledge models. However, some extraction results contain errors in the focus of the questions. For example, "Thermal and freezing disasters can degrade the mechanical properties of rocks and building materials, and affect the service life of tunnels". The focus of the question should be "How do thermal and freezing disasters affect the lifespan of tunnels", which would better meet professional requirements. However, the large model's question focused on the types of factors affecting the lifespan of tunnels. Therefore, it is necessary to include this sentence into the one-shot prompt learning, transforming the one-shot into a few-shot learning. It is worth noting that adding too many examples may affect the understanding of the general LLMs. Only sentences typical for the current task should be selected and included in the few-shot learning. The specific number and types of these sentences should be adjusted according to the task. We found that 5-10 is the optimal number during the experiment.

It is important to note that manually tuning the results of knowledge extraction is crucial. Relying solely on general LLMs for knowledge extraction will limit capabilities of our trained model to the general LLMs. During the manual correction process, there will inevitably be variations in the quality of extraction due to differences in human expertise, especially when there are many individuals involved in the extraction. Therefore, it is essential to establish a set of unified standards, such as specifying uniform sentence structure requirements and content that should be omitted. When this framework is transferred to other fields, it is crucial to have personnel with knowledge of the relevant domain involved.

Currently, the first phase of domain knowledge organization has been completed. Training dataset for UndergrGPT was preliminarily formed (2W). The work on expanding the dataset is ongoing, and we will continue to release updates in the future. It is worth noting that some underground engineering information may be sensitive, and desensitization operations should be performed in the data.

3.2. Construction of UndergrGPT

Generative Pre-Trained Transformer (GPT) is a deep learning-based natural language processing model that uses a multi-head, multi-layer transformer architecture[30,31]. The core of GPT is to complete various custom tasks through a general generative language model combined with fine-tuning training. The primary method for constructing UndergrGPT is to fine-tune a general LLM on a domain-specific knowledge dataset.

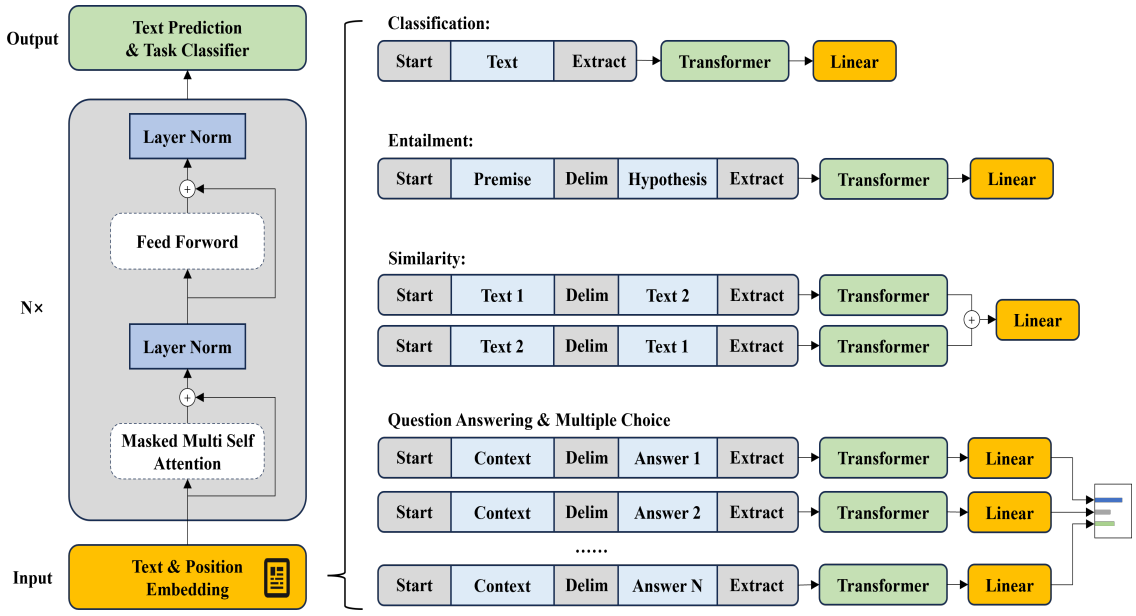

Figure 7. Supervised fine-tune data conversion method.

Generative LLMs are trained on continuous text, while some natural language processing tasks use non-continuous text data. For example, question-answering tasks include three part: text, questions, and answers, requiring adjustments to fit the input format of generative language models. Radford[30] carried out a traversal-style approach, where structured inputs were converted into an ordered sequence that pre-trained model can process. As shown in Figure 7, for Question Answering and Multiple-Choice tasks, each answer is concatenated with the context using a delimiter (delim) to form the input for the generative language model.

We used ChatGLM3-6b as base model, which is an open-source, large-scale pre-trained language model employing the transformer architecture and consisting of 6 billion parameters, with a context token length of up to 8K. As for the part of LoRA, the lora rank is set to 8, lora alpha to 32, and lora dropout to 0.1. The number of parameters that need to be updated is approximately 1.95M, which is only 0.03% of the ChatGLM large model's 6000M parameters. The model training was conducted using 8 A40(40G) GPUs, with a batch size of 23 per GPU. The maximum input and output length was set to 256 tokens. To accelerate training, the DeepSpeed ZeRO-2 strategy was employed. The model was trained for 700 epochs, with a training duration of approximately 4 hours. After 600 iterations, the loss stabilized at around 0.033.

It is noteworthy that training solely on underground engineering domain knowledge can reduce the model's general question-answering capabilities. Therefore, it is necessary to include general question-answering corpora during the training process. Moreover, compared to the parameter settings of LoRA, the batch size has a greater impact on the quality of the model's responses.

As for retrieval-augmented generation, to avoid disrupting the original semantics of corpus in the knowledge database, we divided the retrieval granularity by sentences rather than by length. The retrieval results most relevant to the question are selected and combined with the question as the prompt.

3.3. Evaluation of UndergrGPT

The foundation for testing LLMs in the field of underground engineering is a publicly available benchmark test dataset, which ensures a unified standard for large language model testing. After extensive research, we discovered that there was no fundamental, publicly available test dataset in the domain of underground engineering. Consequently, we constructed a benchmark

dataset (UndergrTest) and carried out provided standard answers and scoring rules, as shown in Table 1.

Table 1. The benchmark test dataset for underground engineering (UndergrTest).

Type	Evaluation Criteria	Num
Multiple choice	Single choice: Full marks for correct answers, no points for incorrect answers; Multiple choice: Full marks for selecting all correct options, 70% of the points for missing options, no points for incorrect selections.	600
Open-ended question	Assign points based on the proportion of correct keywords provided in the answer.	400
Note: The final score is the average score of all questions.		

To demonstrate the superior performance of UndergrGPT, we conducted a comparative analysis with the baseline models Baichuan, ChatGLM, and ChatGPT based on the UndergrTest. The evaluation results are shown in Figure 8. In both multiple choice and open-ended questions, UndergrGPT achieved significantly higher average scores than general large language models.

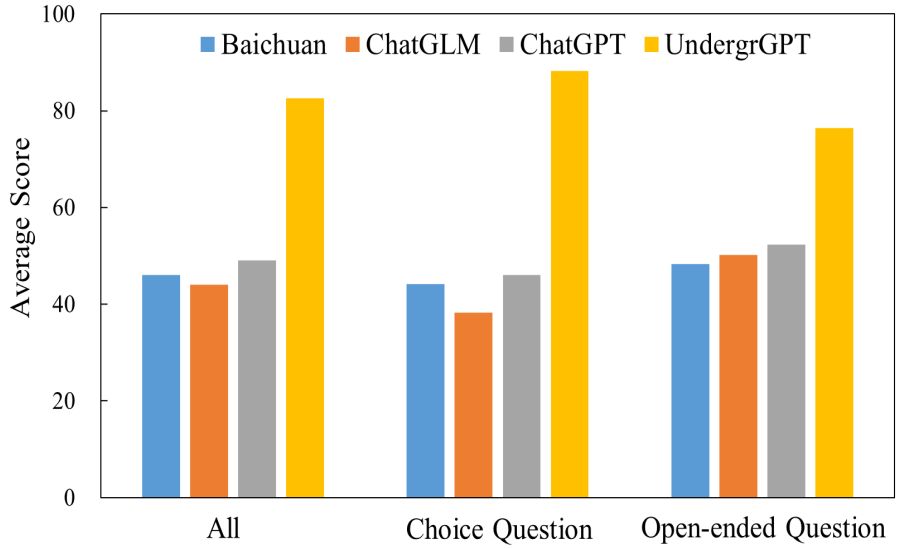

Figure 8. Model comparison result.

The results of the model responses are shown in Table 2. Based on the framework proposed, UnderGPT effectively suppresses hallucinations. For example, in questions (2) and (3), the BEAM system was first developed by a German company, and the Makou Pai Tunnel is part of the Longhua Expressway project in Guangdong Province. ChatGPT incorrectly answered that the BEAM system was developed by the Institute of Geology and Geophysics, Chinese Academy of Sciences, and that the Makou Pai Tunnel was part of the Nanjing Metro Line S7 project. UndergrGPT, with the support of an external database, can provide accurate answers.

Table 2. Evaluation result between Baichuan, ChatGLM, ChatGPT and UndergrGPT.

LLM	Question and Answer
Baichuan	[Single-choice question] When classifying highway tunnels by length, the criterion for defining an extra-long tunnel is ( ). A. L > 3000m B. 3000m L > 1000m C. 1000m L > 500m D. L 500m B

Table 2. *Cont.*

LLM	Question and Answer
ChatGLM	The question is about tunnel engineering. An extra-long tunnel refers to a tunnel longer than 3000 meters. Extra-long tunnels can be classified according to different criteria, such as length: Extra-long tunnel (L > 3000m), long tunnel (3000m L > 1000m), medium tunnel (1000m L > 500m), and short tunnel (L 500m). Option A is correct. Options B, C, and D: These are all distractors. They do not match the question stem and can be eliminated. Therefore, the correct answer is A.
ChatGPT	A.L>3000m
UndergrGPT	Correct answer: A. Explanation: Highway tunnels can be classified into four categories based on their length: Extra-long tunnel (L > 3000m), long tunnel (3000m L > 1000m), medium tunnel (1000m L > 500m), and short tunnel (L 500m).
	Which project does Makoupai tunnel belong to?
Baichuan	The Makoupai Tunnel is located in Fuzhou, Fujian Province, China. And it is part of the Fuzhou-Xiamen Expressway (Fu-Xiamen Expressway).
ChatGLM	The Macau Border Tunnel is a part of the Guangdong Province's Zhuhai to Hong Kong Expressway (G4) in Xiangzhou District, Zhuhai City, China.
ChatGPT	Majoupai Tunnel belongs to Nanjing Metro Line 7 project.
UndergrGPT	Majoupai Tunnel belongs to Longhuai high-speed project in Guangdong Province.
	What is BEAM in advanced geological prediction?
Baichuan	BEAM refers to the formation lithology prediction in advance geological prediction, which classifies and describes the underground rock through the comprehensive analysis of the underground rock type, structure, structure and other characteristics of the underground rock.
ChatGLM	BEAM is a technique in geological forecasting, which is primarily used for seismic exploration data processing and geological interpretation.
ChatGPT	BEAM system is developed by the Institute of Geology and Geophysics, Chinese Academy of Sciences, which is used for real-time monitoring, analysis and prediction of geological hazards.
UndergrGPT	BEAM is a new type of electrical advance prediction technology developed by a German company, which is based on electrical testing. Technicians install electrodes on the TBM equipment and the tunnel wall to emit current. And they judge the geological conditions aheadthrough the monitor in real time.

4. Discussion

The development direction for underground engineering domain knowledge LLM aims to establish a general-purpose question-answering system, which includes downstream tasks such as basic knowledge queries, construction plan recommendations, and disaster response decisions, etc. Achieving this requires domain knowledge databases and massive datasets. Currently, we focus on basic knowledge question-answering. To develop a comprehensive domain question-answering system, the research directions and dataset expansion capabilities of a single institution are limited. Therefore, we plan to expand our dataset while actively open-sourcing our work. We hope to collaborate with peers to build and maintain the knowledge base and datasets, thereby contributing to the advancement of large language models for underground engineering.

5. Conclusion

To address the challenges of limited domain knowledge training datasets, high computational resource consumption, and model hallucinations in the construction of domain LLMs, we proposed a framework for building underground engineering domain knowledge LLM. The framework comprises three components: human-machine collaborative knowledge extraction, parameter efficient fine-tuning, and retrieval augmented generation. Human-machine collaborative knowledge extraction method significantly reduces human labor in constructing the knowledge database and training dataset with the assistance of general LLMs, which provides a novel solution for building large-scale datasets in this field. Parameter efficient fine-tuning method reduces the number of parameters that need to be updated to less than 1% of the original, enabling model fine-tuning with fewer and lower version GPUs. Retrieval augmented generation method incorporates non-parametric knowledge from an external

domain knowledge base, effectively enhancing the accuracy of the model's responses and mitigating model hallucinations. Under this framework, we constructed the domain knowledge database (UndergrData), training dataset, benchmark test dataset (UndergrTest), and trained the first underground engineering domain knowledge LLM (UndergrGPT).

Acknowledgments

This work was funded by the National Natural Science Foundation of China(Grant Nos. 52279103 and 52379103).

Conflicts of Interests

The authors declare that they have no conflicts of interest in this paper.

Authors contribution

Zhenhao Xu: Methodology, Conceptualization, Investigation, Funding acquisition, Writing - review & editing. Zhaoyang Wang: Methodology, Data curation, Validation, Visualization, Writing - review & editing. Pengjie Ren: Methodology, Software. Xiao Zhang: Investigation. Tianhao Li: Data curation.

References

[1] Xu Z, Liu F, Lin P, Shao R, Shi X. Non-destructive, in-situ, fast identification of adverse geology in tunnels based on anomalies analysis of element content. *Tunnelling Underground Space Technol.* 2021, 118:104146.

[2] Xu Z, Yu T, Lin P, Wang W, Shao R. Integrated geochemical, mineralogical, and microstructural identification of faults in tunnels and its application to TBM jamming analysis. *Tunnelling Underground Space Technol.* 2022, 128:104650.

[3] Xu Z, Yu T, Lin P, Li S. Anomalous patterns of clay minerals in fault zones. *Eng. Geol.* 2023, 325:107279.

[4] Du X, Hao H, Spencer BF, Zhao Y. Smart Construction: digitization and intelligence transformation. *Smart Constr.* 2024, 1(1):0006.

[5] Xiao Y, Yang T. An automated crane operation and construction material supply strategy. *Smart Constr.* 2024, 1(1):0002.

[6] Meng F, Wang W, Wang J. Visual analysis of the research status of intelligent question answering system. In *2021 13th International Conference on Computational Intelligence and Communication Networks (CICN)*. Piscataway, NJ: IEEE, 2021, pp. 145–149.

[7] Wang H, Fu T, Du Y, Gao W, Huang K, *et al.* Scientific discovery in the age of artificial intelligence. *Nature* 2023, 620(7972):47–60.

[8] Zhu S, Yu T, Xu T, Chen H, Dustdar S, *et al.* Intelligent computing: the latest advances, challenges, and future. *Intell. Comput.* 2023, 2:0006.

[9] Zhu Q, Wang S, Ding Y, Zeng H, Zhao L, *et al.* A method of safety-quality-schedule knowledge graph for intelligent management of drilling and blasting construction of railway tunnels. *Geomatics Inf. Sci. Wuhan Univ.* 2022, 47(8):1155–1164.

[10] Yang Q. Research on the intelligent question answering based on knowledge graph. In *2021 International Conference on Big Data and Intelligent Decision Making (BDIDM)*. Piscataway, NJ: IEEE, 2021, pp. 226–229.

[11] Nair L, Shivani M. Knowledge graph based question answering system for remote school education. In *2022 International Conference on Connected Systems & Intelligence (CSI)*. Piscataway, NJ: IEEE, 2022, pp.1–5.

[12] Bonadia R, Trindade F, Freitas W, Venkatesh B. On the potential of ChatGPT to generate distribution systems for load flow studies using openDSS. *IEEE Trans. Power Syst.* 2023,

38(6):5965–5968.
[13] Li Y, Wang S, Ding H, Chen H. Large language models in finance: A survey. In *Proceedings of the Fourth ACM International Conference on AI in Finance (ICAIF)*. New York: ACM, 2023, pp. 374–382.
[14] Liu B, Zhang H, Liu J, Wang Q. ACIGS: An automated large-scale crops image generation system based on large visual language multi-modal models. In *International Conference on Sensing, Communication, and Networking (SECON)*. Piscataway, NJ: IEEE, 2023, pp.7–13.
[15] Wan W, Liao B, Cao J, Zhang B, Zhang L, *et al*. Transforming automated customer service responses using ChatGLM model: a case study of expressway toll station information system. In *2023 IEEE 7th Information Technology and Mechatronics Engineering Conference (ITOEC)*. Piscataway, NJ: IEEE, 2023, pp. 1968–1973.
[16] Wang Z, Zhou Q, Zhao J, Wang Y, Ding H, *et al*. A knowledge-enhanced medical named entity recognition method that integrates pre-trained language models. In *2023 IEEE International Conference on Medical Artificial Intelligence (MedAI)*. Piscataway, NJ: IEEE, 2023, pp. 296–301.
[17] Cheng B, Zhang J, Liu H, Cai M, Wang Y. Research on medical knowledge graph for stroke. *J. Healthc. Eng.* 2021:5531327.
[18] Peng C, Xia F, Naseriparsa M, Osborne F. Knowledge graphs: opportunities and challenges. *Artif. Intell. Rev.* 2023, 56(11):13071–13102.
[19] Qiu Q, Wang B, Ma K, Lv H, Tao L, *et al*. A practical approach to constructing a geological knowledge graph: a case study of mineral exploration data.*J. Earth Sci.* 2023, 34(5):1374–1389.
[20] Zhang Y, Liu J, Hou K. Building a knowledge base of bridge maintenance using knowledge graph. *Adv. Civ. Eng.* 2023:6047489.
[21] Peng F, Qiao Y, Yang C. Building a knowledge graph for operational hazard management of utility tunnels. *Expert Syst. Appl.* 2023, 223:119901.
[22] Brown T, Mann B, Ryder N, Subbiah M, Kaplan J, *et al*. Language models are few-shot learners. *arXiv* 2020, ArXiv:2005.14165.
[23] Lv K, Yang Y, Liu T, Gao Q, Guo Q, *et al*. Full parameter fine-tuning for large Language models with limited resources. *arXiv* 2023, ArXiv:2306.09782.
[24] Hu E, Shen Y, Wallis P, Allen-Zhu Z, Li Y, *et al*. Lora: low-rank adaptation of large language models. *arXiv* 2021, ArXiv: 2106.09685.
[25] Roberts A, Raffel C, Shazeer N. How much knowledge can you pack into the parameters of a language model? *arXiv* 2020, ArXiv: 2002.08910.
[26] Lewis P, Perez E, Piktus A, Piktus A, Petroni F, *et al*. Retrieval-augmented generation for knowledge-intensive NLP tasks. *Adv. Neural Inf. Process. Syst.* 2020, 33:9459–9474.
[27] Karpukhin V, Oguz B, Min S, Lewis P, Wu L. Dense passage retrieval for open-domain question answering. *arXiv* 2020, ArXiv: 2004.04906.
[28] Kaiser L, Gomez A, Shazeer N, Vaswani A, Parmar N. One model to learn them all. *arXiv* 2017, ArXiv: 1706.05137.
[29] Finn C, Abbeel P, Levine S. Model-agnostic Meta-Learning for fast adaptation of deep networks. In Proceedings of the 34th International Conference on Machine Learning (ICML). New York: PMLR, 2017, PP.1126–1135.
[30] Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, *et al*. Attention is all you need. *Adv. Neural Inf. Process. Syst.* 2017.
[31] Radford A, Narasimhan K, Salimans T, Sutskever I. Improving language understanding by generative pre-training. 2018.