Article | Received 11 June, 2026; Accepted 18 September, 2026; Published 10 October, 2026
https://doi.org/10.55092/pe20260013

Knowledge generation and evaluation for construction quality assessment using retrieval-augmented large language models

Weiting Liu[1], Meng Wang[2,*], Ziheng Tian[2], Yufei Hu[2] and Chun Li[2]

[1] Shanghai Urban Construction Vocational College, Shanghai, China
[2] China Construction Digital Technology Co., Ltd., Beijing, China

* Correspondence author; E-mail: wangmeng.work@cscec.com.

Abstract: To address the problems of delayed knowledge updating, inefficient information acquisition, and difficulty in cross-project knowledge reuse in civil engineering construction quality assessment, this paper proposes a retrieval-augmented generation (RAG)-based methodology for injecting domain knowledge into large language models (LLMs). Multi-source knowledge, including national standards, journal articles, textbooks, and engineering reports, was systematically collected, cleaned, semantically segmented, and encoded using the bge-large-zh-v1.5 model to build a Milvus vector database. A general-purpose LLM was selected via semantic perplexity, and similarity-based retrieval was employed to construct prompt templates integrating role settings, prior knowledge, user queries, and response requirements, thereby enabling domain adaptation without additional training. An evaluation dataset containing 390 objective questions from registered architect and supervising engineer qualification examinations was constructed, and six representative LLMs along with their RAG-enhanced counterparts were evaluated. The results show that the average accuracy of general LLMs is 61.3%, with DeepSeek achieving the highest at 73.9%; after RAG enhancement, the accuracy rises to 86.3%, with significant improvements across all subjects. The proposed method effectively mitigates the hallucination phenomenon and provides traceable answers with source citations, enhancing both professionalism and interpretability, and offering a viable technical pathway for the digital transformation of the civil engineering industry.

Keywords: retrieval-augmented generation; large language model; civil engineering; construction quality assessment; knowledge management

1. Introduction

Construction quality assessment in civil engineering represents a highly complex, knowledge-intensive domain. Traditionally, knowledge management in this field has relied heavily on expert experience and document retrieval, leading to issues such as delayed knowledge updating, inefficient information acquisition, and difficulties in cross-project knowledge reuse. As construction projects continue to grow in scale and technical complexity, effectively leveraging advanced artificial intelligence technologies to enhance the efficiency of knowledge management for construction quality assessment has emerged as a critical problem demanding urgent solutions.

Copyright©2026 by the authors. Published by ELSP. This work is licensed under Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium provided the original work is properly cited.

Recent breakthroughs in artificial intelligence, particularly the rapid development of large language models (LLMs), are profoundly transforming operational paradigms across industries. Since the release of ChatGPT in November 2022, LLMs have attracted global attention owing to their powerful performance on a wide range of natural language tasks. By training on massive text corpora with billions of parameters, these models have demonstrated general-purpose language understanding and generation capabilities [1]. Concurrently, retrieval-augmented generation (RAG) [2], as an innovative knowledge fusion approach, provides an effective means of addressing the challenges of knowledge updating and factual accuracy in LLMs by combining parametric and non-parametric memories. Meta's LLaMA series models [3] have exerted a significant influence on the open-source LLM landscape. The LLaMA series offers open-source models with parameter scales ranging from 7B to 65B, trained on trillions of tokens, demonstrating that state-of-the-art models can be trained using only publicly available datasets without relying on proprietary data. Google's PaLM series models [4] have further evidenced the continuous benefits of scale by achieving state-of-the-art few-shot learning results across hundreds of language understanding and generation benchmarks.

In terms of model architecture optimization, the Mixture-of-Experts (MoE) architecture allows model parameters to be scaled to trillions with ease, activating only a few experts during computation to improve computational efficiency. The DeepSeekMoE architecture [5] finely segments experts into multiple units and selectively activates them, permitting more flexible expert combinations; meanwhile, it isolates some experts as shared experts to capture common knowledge and reduce redundancy among routed experts. GLM-130B [6], the first open-source 100-billion-level bilingual (Chinese-English) pre-trained language model, has exhibited significant advantages across a broad range of multilingual benchmarks. The release of OpenAI's GPT-4 model [7] marked the advent of multi-modal LLMs, capable of accepting both image and text inputs and producing text outputs, and demonstrating human-level performance on various professional and academic benchmarks.

Although pre-trained LLMs store vast amounts of factual knowledge in their parameters and achieve state-of-the-art results when fine-tuned on downstream NLP tasks, their ability to access and precisely manipulate knowledge remains limited; consequently, their performance on knowledge-intensive tasks lags behind that of task-specific architectures. The core idea of RAG is to integrate parametric and non-parametric memories for language generation. The basic architecture of a RAG model comprises two principal components: a retriever and a generator. The retriever fetches relevant documents from an external knowledge base based on the input query, and the generator produces the final output conditioning on both the retrieved documents and the original input [2]. The development of RAG technology has primarily focused on two aspects: retrieval algorithm optimization and generation quality enhancement. Regarding retrieval algorithms, researchers have proposed a variety of innovative methods to improve retrieval efficiency and accuracy. The s3 framework [8], a lightweight and model-agnostic framework, decouples the searcher from the generator and trains the searcher using a Gain Beyond RAG reward—the improvement in generation accuracy relative to naive RAG. Sparse RAG [9] reduces computational cost through sparsity by encoding retrieved documents in parallel, thereby eliminating the latency introduced by long-range attention over retrieved documents; the LLM then selectively decodes the output via prompts with special control tokens, attending autoregressively only to highly relevant caches. xRAG [10], a novel context compression method designed specifically for RAG, redefines document embeddings as features of the retrieval modality and employs modality fusion to effectively incorporate these embeddings into the language model's

representation space, eliminating the need for textual counterparts and achieving extremely high compression rates. TurboRAG [11] proposes a hybrid offline–online paradigm that precomputes chunk-level key–value (KV) caches and stitches them together at inference time using independent attention and reordering RoPE techniques, preserving answer quality without modifying the model architecture. Retrieval-Augmented CM3 (RA-CM3) [12], the first multi-modal model capable of retrieving and generating both text and images, significantly outperforms baseline multi-modal models such as DALL-E and CM3 on image and caption generation tasks.

In terms of generation quality improvement, researchers have introduced various methods to strengthen the generative capability of RAG systems. The MINT framework [13] enhances RAG by jointly training RAG and generation-augmented retrieval (GAR) within a unified encoder–decoder architecture, bridging the gap between the retriever and the generator and employing an iterative co-training strategy between RAG and GAR to achieve mutual enhancement through pseudo-sample generation. CRAG (Corrective Retrieval Augmented Generation) [14] designs a lightweight retrieval evaluator to assess the overall quality of retrieved documents and return a confidence score, based on which different knowledge retrieval actions can be triggered. RAT (Retrieval-Augmented Thoughts) [15] explores how to iteratively revise chains of thought with the aid of information retrieval, substantially improving the reasoning and generation capabilities of LLMs on long-text generation tasks while dramatically reducing hallucinations. Know³-RAG [16], a knowledge-aware RAG framework, utilizes the structured knowledge of knowledge graphs (KGs) to guide the three core stages of the RAG process—retrieval, generation, and filtering. The framework introduces a knowledge-aware adaptive retrieval module that employs KG embeddings to evaluate the confidence of generated answers and determine the necessity of retrieval.

To systematically assess RAG performance, researchers have established dedicated benchmarking frameworks. RGB (Retrieval-Augmented Generation Benchmark) [17] is the first benchmark specifically designed for evaluating the retrieval-augmented generation capabilities of LLMs, created in both English and Chinese. FedRAG [18] serves as a framework for fine-tuning RAG systems, supporting state-of-the-art fine-tuning methods across both centralized and federated architectures and providing simple, intuitive interfaces with seamless transitions from centralized to federated training tasks.

In the field of civil engineering construction, LLMs have shown certain application potential. GPT-4 scored approximately 67% on the NCEES (National Council of Examiners for Engineering and Surveying) civil engineering and geotechnical engineering practice examination questions. While this result demonstrates a degree of specialized knowledge understanding, it also exposes limitations in handling complex technical problems [19]. In vertical domains, the lack of large-scale structured corpora and the requirement for high-precision responses significantly increase the difficulty of building domain-specific knowledge LLMs. UndergrGPT [20] proposed a framework and construction methodology for a knowledge LLM in the field of underground engineering. RAG technology has also been employed to enhance engineering risk analysis. The growing complexity and volume of unstructured risk-related data in engineering projects pose major challenges for timely and accurate risk analysis. In the context of construction safety risk assessment, Speiser *et al.* [21] developed a safety risk assessment platform that leverages an agent-based AI system, LLMs, hybrid semantic search, and RAG to generate expert-level safety risk assessments. The system integrates a wide range of safety knowledge sources into a single framework, enabling users to create risk assessment reports for specific work tasks based purely on natural language input.

Although LLMs and RAG technology have achieved notable success in general domains, their application in the specialized field of civil engineering construction remains in its infancy. Current research predominantly focuses on general text processing and question-answering systems, with relatively limited studies addressing civil engineering construction expertise. This research gap not only constrains the depth of AI application in this domain but also hinders the digital transformation of the civil engineering industry.

2. Methods

LLMs enables the efficient retrieval of external databases to supply the model with domain-specific knowledge prompts, thereby achieving knowledge embedding in the field of civil engineering through a zero-training approach. Accordingly, this chapter adopts the LLM-based RAG methodology: it collects knowledge from the civil engineering domain, performs structured processing, and constructs a vector database. Domain adaptation is realized through the evaluation of general-purpose LLMs on civil engineering knowledge. By combining user queries with retrieved relevant knowledge as prompts, the methodology enhances the LLM's capability to generate professional answers and improves the efficiency of knowledge updating.

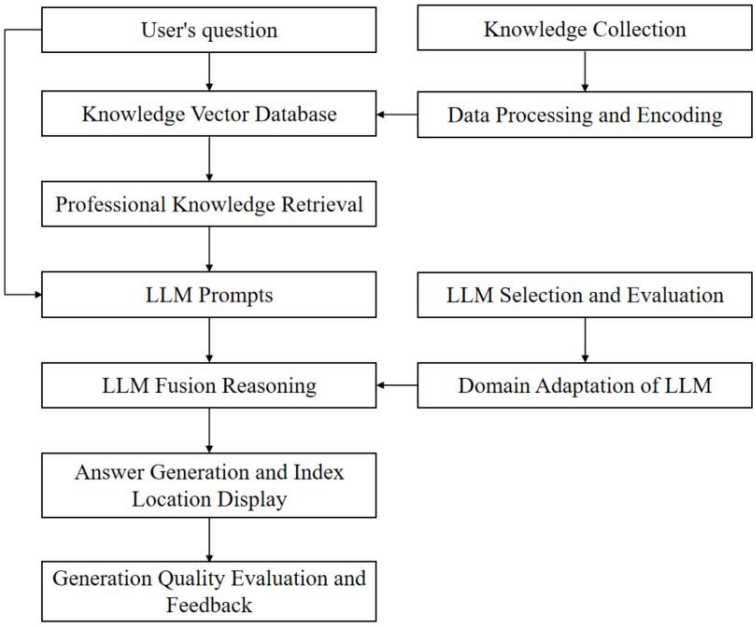


Figure 1. RAG methodology for LLMs.

As illustrated in Figure 1, this chapter collects knowledge from the domain of civil engineering construction quality assessment, encodes it, and establishes a vector database. A general-purpose LLM suitable for RAG in the civil engineering domain is selected using a semantic perplexity metric. Target knowledge is identified by computing the similarity between the user's question and the indexed entries in the database, and then formulated into prompts, thereby augmenting the LLM's ability to generate specialized knowledge in civil engineering.

2.1. Data acquisition and processing

In the data acquisition phase, this section systematically covers the principal knowledge carriers in the field of civil engineering construction quality assessment, specifically including five categories of sources. First, current and effective national and industry standards and specifications, such as the Unified Standard for Construction Quality Acceptance of Building Engineering and the Code for Acceptance of Construction Quality of Concrete Structures, are collected, with priority given to editable-format documents officially released. Second, journal articles, dissertations, and conference papers published over the past decade in databases such as CNKI, Wanfang Data, and Web of Science are gathered, with an emphasis on highly cited literature and core journal publications. Third, textbooks widely used in civil engineering programs at domestic universities are sourced, and chapters directly related to quality assessment are extracted. Fourth, construction quality assessment reports, completion acceptance reports, monthly supervision reports, and quality issue handling records from typical engineering projects are collected, with any sensitive information therein subjected to desensitization. Fifth, auxiliary materials, including design drawing specifications, construction organization plans, industry technical reports, expert lecture notes, and internal enterprise quality standards, are also incorporated.

During the data cleaning and preprocessing stage, all collected documents are first uniformly converted into plain text format with UTF-8 encoding. For PDF files, a combination of optical character recognition and layout analysis tools is employed to distinguish different structural elements such as body text, tables, headings, and annotations. Subsequently, a noise removal operation is performed to eliminate headers and footers, watermarks, irrelevant advertisements, garbled characters, and duplicate paragraphs, as well as content not directly related to construction quality assessment. On this basis, structured metadata tags—including source type, publication date, professional sub-discipline, and a unique document identifier—are attached to each document to support subsequent retrieval filtering and traceability. After cleaning, a document segmentation strategy based on semantic boundaries is applied to partition knowledge units appropriately. Using chapters or natural paragraphs as the basic unit, the maximum segment length is set to 512 characters; for units exceeding this length, a sliding window approach with a 256-character overlap is adopted to preserve contextual continuity. For specification-type documents, segmentation is performed independently by clause number, ensuring that each specification clause can serve as an independent knowledge unit for retrieval.

2.2. Data encoding and vector database construction

The core task of data encoding is to transform preprocessed textual knowledge units into computable vector representations and to build a vector storage engine that supports efficient similarity retrieval. Tokenization is first performed on the text, segmenting it into words or subwords $w_1, w_2, \cdots, w_T$. The encoding model aims to predict the context words given a center word; its encoding function essentially constructs word vector representations such that semantically similar words are located close to each other in the vector space. The calculation is expressed in Equation (1):

$$P(w_j \mid w_i) = \frac{e^{\mathbf{v}_{w_i} \cdot \mathbf{u}_{w_j}}}{\sum_{k \in V} e^{\mathbf{v}_{w_i} \cdot \mathbf{u}_k}} \tag{1}$$

where w_i denotes the center word, w_j denotes a context word within a window of size c, $\mathbf{v}_w$ represents the input word vector and $\mathbf{u}_w$ represents the output word vector. The objective of the model is to adjust the

values of the word vectors $\mathbf{v}_w$ and $\mathbf{u}_w$ so as to maximize the probability of predicting all context words given the center words. Encoded vectors with small spatial distances thereby possess similar semantics.

With respect to the selection and configuration of the encoding model, this section adopts a general-purpose embedding model that is suitable for Chinese semantic understanding and supports long texts. Specifically, the bge-large-zh-v1.5 model, pre-trained on large-scale corpora, is employed. The vector dimensionality is set to 768, striking a balance between semantic expressiveness and retrieval efficiency. The encoding process adopts a batch processing mechanism, with 64 text units per batch, and utilizes GPU acceleration. Each text unit generates a floating-point vector, which is persistently stored in NumPy array format, while the mapping relationships among the vectors, document identifiers, and text passages are retained.

Regarding the selection and configuration of the vector database, Milvus is chosen as the storage engine due to its support for high-concurrency and low-latency vector retrieval. The HNSW (Hierarchical Navigable Small World) index type is adopted. The maximum number of connections per node is set to 16; a larger value improves the recall rate but increases memory consumption and construction time. The number of candidate neighbors is set to 200; a higher value yields better index quality but prolongs construction time, thereby achieving an optimal trade-off between recall rate and retrieval speed. Cosine similarity is selected as the distance metric; the vectors are L2-normalized before computing the dot product, resulting in a value range of [−1,1], where a larger value indicates greater semantic similarity. Memory is allocated according to the data scale, with a 20% redundancy reserved to support dynamic insertions. When the data volume exceeds the capacity of a single node, horizontal sharding is performed based on document source types, creating sub-collections such as a specification repository, a literature repository, and a case repository, thereby facilitating sub-repository retrieval and permission management. Data persistence is simultaneously enabled, with a full backup configured once per day and incremental backups every hour to safeguard data security. During the data ingestion and index construction stage, each record comprises four fields: vector, unique document identifier, text content, and metadata. After the index construction is completed, index warm-up is performed to load frequently used index blocks into memory, thereby reducing first-time retrieval latency, and the index validity is verified through sampling queries.

2.3. Retrieval-augmented generation

This section aims to realize the collaborative operation of the retrieval module and the LLM, so as to generate professional answers that integrate prior knowledge, and to establish a feedback loop for continuous system performance optimization.

A general-purpose LLM with strong capabilities in understanding and reasoning over civil engineering domain knowledge is selected for RAG. Civil engineering domain texts are chosen, and based on the model's own parameters, the probability of each word in the text given the preceding words is calculated; the semantic perplexity of the model is then computed as the evaluation metric, as shown in Equation (2):

$$PPL(W) = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}} = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{p(w_i | \ w_1 w_2 \ldots w_{i-1})}} \tag{2}$$

where w_i denotes the i-th word in the text, and the text length is N. A model with lower semantic perplexity can more accurately capture the semantics and linguistic regularities of civil engineering professional texts, thereby making it easier to generate knowledge based on the prompts.

A retrieval module based on similarity computation is designed. The objective of the retrieval module is to find the k documents most similar to the user's query, *i.e.*, to compute the similarity scores between the query vector and all stored vectors, and return the document records with the highest scores. Regarding the design and implementation of the retrieval module, after a user inputs a query, the same encoding model used in the second stage is first employed to convert the query into a query vector, and then vector similarity search is executed. The similarity calculation method is shown in Equation (3):

$$s(x,d) = \frac{x \cdot d}{\| x\| \cdot \| d\|} \tag{3}$$

where the question input is q, and d represents a document encoding in the knowledge base, with s(q, d) denoting the similarity between the question and the knowledge, representing the vector dot product, and $\|\cdot\|$ representing the Euclidean norm of the vector.

The number of recalled items is set according to the application scenario: the top 5 items are retrieved for routine question answering, and the top 10 for complex engineering problems. During the retrieval process, metadata-based filtering can be applied, for instance, restricting the retrieval scope solely to the specification and standard repository or to a specific professional sub-discipline. Meanwhile, a lower similarity threshold of 0.65 is set, and retrieval results below this threshold are discarded to avoid introducing low-relevance knowledge that would interfere with generation. The retrieved results are sorted in descending order of similarity; when multiple documents originate from the same source and possess highly similar content, only the one with the highest similarity is retained. The recalled text passages are then concatenated, with original passage boundaries preserved, and source annotations are added to facilitate traceability.

At the prompt construction and generation enhancement stage, this method adopts a four-part prompt template structure consisting of "role setting—prior knowledge—user question—response requirements." The retrieved knowledge texts are sorted by relevance and inserted into the template, and the total length of the retrieved knowledge portion is controlled to not exceed 60% of the model's context length, thereby reserving space for answer generation. The generation parameters are configured with a temperature coefficient of 0.2 to reduce the randomness of the generated content, and the maximum generation length is set to 1024 tokens to meet the requirements of most question-answering scenarios. The user query and the retrieved similar knowledge texts are merged to form a civil engineering prior knowledge prompt. The concatenation method is shown in Equation (4):

$$c = \text{Concat}(p, d_1, d_2, \ldots, d_k, x, r) \tag{4}$$

where Concat() denotes the vector concatenation operation, p is the role setting, d_i denotes the i-th retrieved similar text out of a total of k items, x is the user question input, r represents the response requirements, and c is the fully concatenated prompt.

Furthermore, the prompt is fed into the LLM to augment the generation of professional civil engineering knowledge answers. The objective function is shown in Equation (5):

$$p(y \mid x, D) = \frac{\exp(s(c, y))}{\sum_{y' \in Y} \exp(s(c, y'))} \tag{5}$$

where y is the target generated text, D denotes the knowledge base, and y' represents the set of all possible output text types.

To ensure the credibility of the generated content, a provenance tracing mechanism is integrated into the model output. Through instruction constraints, the model is required to annotate the positions where knowledge is cited with the corresponding knowledge source indices in square brackets, enabling users to independently verify the professionalism and reliability of the answers by cross-referencing the indices in the output with the document sources in the original knowledge base.

3. Evaluation dataset and models

3.1. Evaluation dataset

To evaluate the capability improvement brought by the RAG method for LLMs in the domain of civil engineering construction quality knowledge, this paper selects examination questions from professional qualification examinations, including those for Class 1 Registered Architect, Class 2 Registered Architect, and Supervising Engineer. In order to avoid the subjective bias and uncertainty associated with manual scoring, the evaluation dataset exclusively adopts an objective multiple-choice format. Since this evaluation focuses solely on the textual understanding and reasoning abilities of LLMs with respect to civil engineering domain knowledge, questions involving multimodal information such as tables and figures are manually excluded during the dataset curation process. After data processing, a Chinese evaluation dataset for LLMs in the civil engineering domain is ultimately constructed, covering 6 subjects with a total of 340 single-choice questions and 50 multiple-choice questions. The distribution of the number of questions is presented in Table 1, and parenthesized abbreviations denote the corresponding subjects.

Table 1. Distribution of question types in the benchmark dataset

Certification	Subject	Question Type	Number
Class 1 Registered Architect	Building Economics, Construction and Design Business Management (Management)	Single-choice	68
	Building Materials and Construction (Materials)	Single-choice	63
	Pre-design and Site Design (Design)	Single-choice	41
Class 2 Registered Architect	Building Structures, Building Physics and Equipment (Equipment)	Single-choice	51
Supervising Engineer	Basic Theories and Related Regulations of Construction Project Supervision (Regulations)	Single-choice	50
		Multiple-choice	30
	Construction Project Objective Control (Control)	Single-choice	67
		Multiple-choice	20

3.2. Evaluation models

This paper selects six representative LLMs that are highly recognized and frequently used in terms of model scale, technical innovation, or performance, to evaluate their professional knowledge question-answering capabilities in the civil engineering construction domain. The degree of recognition can be determined based on the models' rankings on widely acknowledged benchmark leaderboards.

Table 2. Basic Information of Some LLMs

Model Name	Model Version	Developer	Features
Spark	spark3.5max	iFLYTEK	Hundreds of billions of parameters; strong capabilities in mathematics, Chinese, coding, and multimodal tasks. Suitable for scenarios demanding higher performance, such as mathematical computation and logical reasoning.
ChatGLM	GLM4	Zhipu AI	Outstanding performance in Chinese language processing; capable of autonomously invoking multiple tools to complete complex tasks.
DeepSeek	DeepSeek-R1	DeepSeek	Post-trained using reinforcement learning, significantly enhancing reasoning capabilities; particularly excels at complex tasks such as mathematics, coding, and natural language reasoning.
GPT	o1-preview	OpenAI	A reasoning large model trained using reinforcement learning that solves complex problems through deep thinking and the generation of internal reasoning chains.
Gemini	Gemini-1.5-pro	Google	Capable of processing up to 1 million tokens of information; possesses significant advantages in tasks involving long documents, long videos, and long audio.
Claude	Claude-3-Opus	Anthropic	Higher accuracy and performance when handling complex tasks; outstanding real-time responsiveness.

4. Evaluation results and analysis

4.1. Evaluation of general-purpose large language models

This paper treats the accuracy of the option labels generated by the LLMs as the scoring criterion. For multiple-choice questions, a score is awarded only when the model's selected options exactly match all correct answers. The criterion for single-choice questions is to select one correct answer from four options, a type of problem that is relatively straightforward. In contrast, multiple-choice questions require selecting all correct answers from several options, necessitating not only the identification of correct answers but also the exclusion of incorrect ones, thus making them more complex. To better assess the LLMs' comprehensive understanding and application of knowledge points, scoring weights are assigned based on educational measurement theory, considering the amount of information provided by each question type and its contribution to the overall evaluation objectives. A higher score for multiple-choice questions reasonably reflects their greater difficulty and the deeper level of knowledge required from the models compared with single-choice questions. Therefore, in accordance with actual examination score settings, single-choice questions are assigned 1 point, multiple-choice questions are assigned 2 points, and the total subject score is calculated by weighting these points.

Answers to the evaluation questions are generated using a zero-shot prompting approach without incorporating chain-of-thought reasoning. Sample evaluation questions are presented in Table 3.

Table 3. The LLMs test accuracy (in %)

Subject	DeepSeek	Spark	ChatGPT	Gemini	ChatGLM	Claude	Average
Management	75.0	**79.4**	72.1	63.2	66.2	42.6	66.4
Materials	**82.5**	73.0	77.8	65.1	65.1	61.9	70.9
Design	**61.0**	58.5	51.2	58.5	36.6	39.0	50.8
Equipment	**74.5**	72.5	**74.5**	70.6	54.9	43.1	65.0
Regulations	**72.8**	72.7	56.4	50.9	49.1	47.3	58.2
Control	**77.6**	76.2	59.1	44.9	60.5	21.1	56.6
Average	**73.9**	72.1	65.2	58.9	55.4	42.5	61.3

Since LLMs generally generate responses through stochastic sampling, only the model's first response is used as the test result during evaluation. Furthermore, even though the prompts for single-choice and multiple-choice questions explicitly state "this question has a single correct option" and "this question has multiple correct options" respectively, the models still exhibit abnormal response behaviors, such as outputting multiple options for a single-choice question, outputting too few options for a multiple-choice question, or failing to provide valid options. All such cases are treated as incorrect responses.

Overall, DeepSeek achieved an average accuracy of 73.9% across the six subjects and obtained the best result in five subjects. The highest average accuracy of any single model on a single subject was 82.5%, and the average accuracy across three subjects was below 60%. This indicates that a gap still exists between LLMs and human experts, and that general-purpose LLMs are still far from achieving the level of capability required for practical application in high-reliability fields such as building construction engineering.

4.2. Evaluation of the retrieval-augmented method

This section evaluates the LLMs enhanced by the RAG method described in Chapter 2 and compares them with standard LLMs. As shown in Table 4, the RAG-enhanced models significantly outperform the average of ordinary LLMs across all subjects. The core advantage lies in the ability to dynamically query an external knowledge base to assist in answering, thereby substantially reducing the "hallucination" phenomenon. That is, instead of relying solely on the potentially outdated or inaccurate "memory" within the model parameters, it retrieves relevant, reliable information in real time as a basis for generation. This makes RAG more accurate and timely when handling questions that require up-to-date data, domain-specific knowledge, and long-tail details, while also enhancing answer interpretability and credibility by citing the retrieved sources.

Table 5 presents examples of responses from the RAG-enhanced LLM and an ordinary LLM to specific questions.

An ordinary LLM relies entirely on the parametric knowledge acquired during training; when encountering ambiguous, contradictory, or uncovered information in the training data, it is prone to generating seemingly plausible but wholly incorrect content, known as model hallucination. RAG first retrieves relevant snippets from an external knowledge base before generation, allowing the model to answer based on actual retrieval results. This effectively provides reference material for the model, greatly reducing the likelihood of fabrications. Moreover, when an ordinary LLM gives an answer, users have no way of knowing its basis. In contrast, RAG can simultaneously return the retrieved source snippets, for

example, specifying a particular clause from a given specification—enabling users to verify the authenticity of the information independently. This not only increases trust in the answer, but also allows for rapid diagnosis of whether an error originates in the retrieval stage or the generation stage when an answer is incorrect.

Table 4. Accuracy of the RAG-enhanced LLM evaluation (in %)

Subject	Retrieval-Augmented LLM	DeepSeek	Average
Management	**86.2**	75.0	66.4
Materials	**90.5**	82.5	70.9
Design	**81.3**	61.0	50.8
Equipment	**84.6**	74.5	65.0
Regulations	**85.3**	72.8	58.2
Control	**89.7**	77.6	56.6
Average	**86.3**	73.9	61.3

Table 5. Response examples

Question	**Among the following reinforced concrete members, the one to which the bonded steel plate strengthening method is NOT applicable is ( ) A. Large eccentric tension member B. Large eccentric compression member C. Flexural memberD. Plain concrete member**
Correct Answer	D
ChatGLM	The bonded steel plate strengthening method is generally not applicable to large eccentric tension members, because in such cases the member may develop a complex stress state, making it impossible for the bonded steel plate to effectively carry tensile forces. Therefore, the correct answer is: A. Large eccentric tension member.
retrieval-augmented LLM	According to Article 9.1.1 of the Code for Design of Strengthening Concrete Structures (GB 50367-2013), this method is applicable to the strengthening of reinforced concrete flexural members, large eccentric compression members, and tension members. This method is not applicable to plain concrete members, including those where the reinforcement ratio on one side of the longitudinal tension reinforcement is less than 0.2%. Therefore, the correct answer is: D. Plain concrete member.

5. Conclusion

By constructing an efficiently retrievable external knowledge database for civil engineering construction quality, this study leverages LLMs to efficiently aggregate user queries and professional knowledge into prompts, thereby answering specialized questions in the field of civil engineering construction quality. The model responds rapidly with high accuracy. Through similarity computation, the model can provide the indexed locations of relevant knowledge, enabling users to verify against the corresponding information and thus enhancing the interpretability and professionalism of the generated responses. This study employs a zero-training approach to infuse general-purpose LLMs with knowledge in civil engineering construction quality, addressing the challenges of learning and updating domain-specific knowledge as well as the risk of catastrophic forgetting in LLMs.

Declaration of generative AI and AI-assisted technologies

During the preparation of this manuscript, the authors used generative AI tools only to improve language and readability. The authors take full responsibility for the content of the manuscript.

Acknowledgments

The work was supported by Technology R&D Project of China Construction Digital Technology Co., Ltd, Development Planning and Scenario Validation Project of Large Language Model for the Construction Industry.

Authors' contribution

Conceptualization, Meng Wang and Weiting Liu; methodology, Meng Wang and Ziheng Tian; software, Ziheng Tian and Yufei Hu; validation, Weiting Liu, Ziheng Tian and Chun Li; formal analysis, Meng Wang and Yufei Hu; investigation, Weiting Liu and Chun Li; resources, Meng Wang and Chun Li; data curation, Ziheng Tian and Yufei Hu; writing—original draft preparation, Weiting Liu and Ziheng Tian; writing—review and editing, Meng Wang, Yufei Hu and Chun Li; visualization, Ziheng Tian and Yufei Hu; supervision, Meng Wang; project administration, Meng Wang. All authors have read and agreed to the published version of the manuscript.

Conflicts of interests

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.

References

[1] Minaee S, Mikolov T, Nikzad N, Chenaghlu M, Socher R, *et al.* Large language models: a survey. *arXiv* 2024, arXiv:2402.06196.

[2] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, *et al.* Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020)*, Vancouver, Canada, December 6–12, 2020, pp. 9459–9474.

[3] Touvron H, Lavril T, Izacard G, Martinet X, Lachaux M-A, *et al.* LLaMA: open and efficient foundation language models. *arXiv* 2023, arXiv:2302.13971.

[4] Chowdhery A, Narang S, Devlin J, Bosma M, Mishra G, *et al.* PaLM: scaling language modeling with pathways. *J. Mach. Learn. Res.* 2023, 24(240): 1–113.

[5] Dai D, Deng C, Zhao C, Xu R X, Gao H, *et al.* DeepSeekMoE: towards ultimate expert specialization in mixture-of-experts language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand, August 11–16, 2024, pp. 1280–1297.

[6] Zeng A, Liu X, Du Z, Wang Z, Lai H, *et al.* GLM-130B: an open bilingual pre-trained model. *arXiv* 2022, arXiv:2210.02414.

[7] Achiam J, Adler S, Agarwal S, Ahmad L, Akkaya I, *et al.* GPT-4 technical report. *arXiv* 2023, arXiv:2303.08774.

[8] Jiang P, Xu X, Lin J, Xiao J, Wang Z, *et al.* s3: you don't need that much data to train a search agent via RL. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China, November 2025, pp. 21599–21617.

[9] Zhu Y, Gu J-C, Sikora C, Ko H, Liu Y, *et al.* Accelerating inference of retrieval-augmented generation via sparse context selection. *arXiv* 2024, arXiv:2405.16178.

[10] Cheng X, Wang X, Zhang X, Ge T, Chen S-Q, *et al.* xRAG: extreme context compression for retrieval-augmented generation with one token. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024)*, Vancouver, BC, Canada, December 10–15, 2024, pp. 109487–109516.

[11] Lu S, Wang H, Rong Y, Chen Z, Tang Y. TurboRAG: accelerating retrieval-augmented generation with precomputed KV caches for chunked text. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, Suzhou, China, November 2025, pp. 6588–6601.

[12] Yasunaga M, Aghajanyan A, Shi W, James R, Leskovec J, *et al.* Retrieval-augmented multimodal language modeling. In *Proceedings of the 40th International Conference on Machine Learning*, Honolulu, Hawaii, USA, July 23–29, 2023, pp. 39755–39769.

[13] Tang Y, Zhang R, Guo J, de Rijke M, Fan Y, *et al.* Boosting retrieval-augmented generation with generation-augmented retrieval: a co-training approach. In *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Padua, Italy, July 13–18, 2025, pp. 2441–2451.

[14] Yan S-Q, Gu J-C, Zhu Y, Ling Z-H. Corrective retrieval augmented generation. *arXiv* 2024, arXiv:2401.15884.

[15] Wang Z, Liu A, Lin H, Li J, Ma X, *et al.* RAT: retrieval augmented thoughts elicit context-aware reasoning and verification in long-horizon generation. In *Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS 2024)*, Vancouver, BC, Canada, December 10–15, 2024.

[16] Liu X, Liu Y, Wu S, Zhang Y, Zhang K, *et al.* Know3-RAG: a knowledge-aware RAG framework with adaptive retrieval, generation, and filtering. *arXiv* 2025, arXiv:2505.12662.

[17] Chen J, Lin H, Han X, Sun L. Benchmarking large language models in retrieval-augmented generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vancouver, BC, Canada, February 20–27, 2024, Vol. 38, No. 16, pp. 17754–17762.

[18] Fajardo V A, Emerson D B, Singh A, Lotif M, Chatrath V, *et al.* FedRAG: a framework for fine-tuning retrieval-augmented generation systems. In Proceedings of the ICML 2025 Workshop on Championing Open-source Development in Machine Learning (CODEML 2025), Vancouver, Canada, July 18, 2025.

[19] Shoemaker T A, Beaino C, Centella R D M, Zhao W, Tanissa C, *et al.* Generative AI: the new geotechnical assistant? *J. Geotech. Geoenviron. Eng.* 2023, 149(10): 02823004.

[20] Xu Z, Wang Z, Ren P, Zhang X, Li T. Framework and construction methodology of underground engineering domain knowledge large language model: UndergrGPT. *Smart Constr.* 2024, 1(2): 0012.

[21] Speiser K, Maciocci G, Teizer J. Risk assessment for construction safety with retrieval-augmented generation and agentic artificial intelligence. In Proceedings of the 24th International Conference on Construction Applications of Virtual Reality, Sydney, Australia, November 4–6, 2024, pp. 397–411.