Applied Human Factors and Ergonomics (AHFE2025), Vol. 199, 2025, 1085–1094
https://doi.org/10.54941/ahfe1006921

Knowledge Graph-Enhanced Large Language Model Framework for Privacy-Preserving Document Processing in the AEC Domain

Fan Yang[1], Hazar Nicholas Dib[2], and Jiansong Zhang[1]

[1]Automation and Intelligent Construction (AutoIC) Lab, School of Construction Management Technology, Purdue University, 363 N. Grant Street, West Lafayette, IN 47907, USA
[2]Information Visualization and Management (IVM) Lab, School of Construction Management Technology, Purdue University, 47907, West Lafayette, IN, USA

ABSTRACT

Data privacy and safety are critical concerns for companies in the Architecture, Engineering, and Construction (AEC) domain, which routinely handle sensitive textual data such as design criteria, project specifications, and compliance records. Protecting this information is vital for maintaining competitive advantage, meeting legal requirements, and ensuring safety and accountability. However, processing such domain-specific data is challenging. Rule-based systems require extensive manual rule sets, while supervised machine learning models need large, annotated datasets - both of which limit scalability and applicability in AEC contexts. Recent advances in large language models (LLMs) offer a promising alternative due to their ability to perform natural language tasks with minimal supervision. Yet, general-purpose LLMs pose two major concerns: they may generate inaccurate or irrelevant outputs on technical content, and their reliance on online services introduces significant privacy risks. To address these issues, this paper proposes a knowledge graph-enhanced LLM framework designed for local, privacy-preserving processing of sensitive AEC documents. Using the 2015 International Building Code (IBC) as an example, the framework operates in two stages. First, an LLM converts selected IBC chapters into a structured knowledge graph with 234 entities, 131 relationships, and 8 communities. Second, another LLM retrieves relevant context from the graph to generate accurate query responses. The system employs open-source models - nomic-embed-text for text embeddings and deepseek-r1 for context retrieval and generation. Evaluation using 661 query-answer-context records showed an average semantic similarity score of 0.83 and an average answer relevancy score of 0.71, indicating high accuracy and contextual alignment. The system runs entirely on a standalone machine, preserving full data privacy and incurring no cost. This work demonstrates a secure and effective approach for using LLMs in privacy-sensitive, domain-specific applications and lays the foundation for broader adoption in similar fields.

Keywords: Knowledge graph, Building code interpretation, Large language models, Retrieval augmented generation, Data privacy

INTRODUCTION

Protecting documents in Architecture, Engineering, and Construction (AEC) is essential to maintain a competitive edge, comply with legal requirements, and safeguard the safety, integrity, and accountability of the built environment. This necessity stems from two primary concerns: data security and data privacy. First, data security is critical because projects rely

on up-to-date textual information (e.g., plans, specifications, contracts, and site reports). If these files are compromised or encrypted in a cyberattack, operations may halt, costs can rise, and on-site risks may increase. For example, a U.K. industry report recorded 77,000 online crime incidents against construction premises in a single year, with most cases linked to malware and only a small share reported to the police, showing both the scale of the threat and the extent of under-reporting (The Construction Index, 2016). Equally important is the issue of data privacy, as many AEC documents are intended for internal use only. Bid packages, design criteria, and security-related plans are often subject to contractual and policy-based restrictions. Guidance such as ISO 19650–5 outlines security-minded information management practices that limit who may access and process sensitive project information (British Standards Institution, 2018). Breaching these controls can have serious consequences. For instance, sending restricted files to online platforms outside from organizational control increases the risk of unauthorized access. If internal bidding documents or detailed cost breakdowns are leaked, competitors can exploit that information to adjust their offers, potentially gaining an unfair advantage. This can lead to financial losses, disputes over award decisions, or even forced re-tendering of projects. Recent construction-sector studies have identified unauthorized access to bidding documents as a high-impact risk (Yao and García de Soto, 2024). Given these risks, adopting a local, privacy-preserving approach to AEC document processing is essential to safeguard both daily operations and sensitive business information.

Implementing a privacy-preserving approach to document processing in the AEC domain requires addressing the inherent limitations of current computational methods. Existing research is shaped by two dominant paradigms, i.e., rule-based and machine learning-based approaches, each of which presents trade-offs in data privacy, scalability, and performance. Rule-based systems rely on manually crafted linguistic and semantic rules to extract and interpret information from domain-specific texts such as building codes (Fuchs, 2021). These systems can achieve high precision in narrow tasks, such as extracting compliance requirements (Zhang and El-Gohary, 2016), yet they demand significant expert input and customization. Moreover, adapting these systems to new projects or jurisdictions often necessitates access to proprietary or sensitive datasets, raising concerns about confidentiality and compliance.

In contrast, machine learning-based methods offer greater flexibility by learning patterns from annotated examples using models such as LSTMs and Transformers (Zhang and El-Gohary, 2019; Zhong et al., 2020). While these models improve adaptability across diverse document types, they typically require large volumes of labeled data. In the AEC context, this presents a challenge, as such datasets often contain contract-bound or security-sensitive content, making them difficult to share, even internally within organizations. Additionally, the opaque nature of deep learning decision-making complicates efforts to ensure transparency, accountability, and alignment with privacy frameworks, particularly in applications that process regulatory or legally binding texts.

Recent advances in large language models (LLMs) offer a new direction for addressing these limitations. LLMs introduce a promising avenue for interpreting and processing AEC text documents due to their ability to perform complex natural language tasks with minimal supervision. In the regulatory domain, Yang and Zhang (2024) used prompt-based LLMs to translate building code provisions into logic programming language, achieving over 97% precision and enabling more efficient compliance checking. Similarly, Fuchs et al. (2024) explored few-shot prompting with GPT-3.5 to convert building regulations into machine-readable logic, demonstrating syntactic and semantic coherence in the generated outputs. Despite this potential, however, general-purpose LLMs raise two critical concerns when applied to private AEC data. First, such models may produce hallucinated, unrelated, or inaccurate outputs, especially when dealing with technical or domain-specific content. Second, the use of online, closed-source LLMs raises serious privacy concerns, as organizations may be unwilling or unable to upload confidential documents to third-party servers for processing.

To mitigate these challenges, in this paper, the authors propose a knowledge graph-enhanced retrieval-augmented generation (RAG) LLMs framework designed for secure, local processing of sensitive AEC documents. The framework consists of two integrated components based on open-source tools: one for generating dense text embeddings and another for context-aware response generation. In the first stage, a large language model extracts semantic entities and relationships from textual documents and organizes them into a structured knowledge graph. In the second stage, this graph is used to retrieve relevant contextual information in response to user queries, which is then used to guide the generation of accurate and contextually grounded answers. To validate the framework, it was applied to 661 query-answer-context records drawn from two chapters of the 2015 International Building Code (IBC). The system achieved an average semantic similarity score of 0.83 and an average answer relevancy score of 0.71, indicating strong alignment between generated responses and the regulatory source. Crucially, the entire framework operates locally on a standalone machine, ensuring that no sensitive data is transmitted to external servers. These results demonstrate the feasibility of integrating knowledge graph and LLMs into privacy-sensitive document processing tasks and highlight the framework's potential for broader adoption in secure, AI-assisted applications across the AEC sector.

METHODOLOGY

Figure 1 shows the workflow of the proposed knowledge graph-based retrieval-augmented generation (RAG) LLMs framework for interpreting and querying textual documents. The system is fully deployed in a local computing environment, ensuring that all large language models operations and data handling remain on-premise to preserve information privacy and organizational security. The process begins with an input document, such

as building codes, specifications, standards, contracts, or technical reports, typically provided in PDF or plain text format. The document is parsed and segmented into manageable text chunks, which are processed by a locally hosted embedding model to extract semantic features.

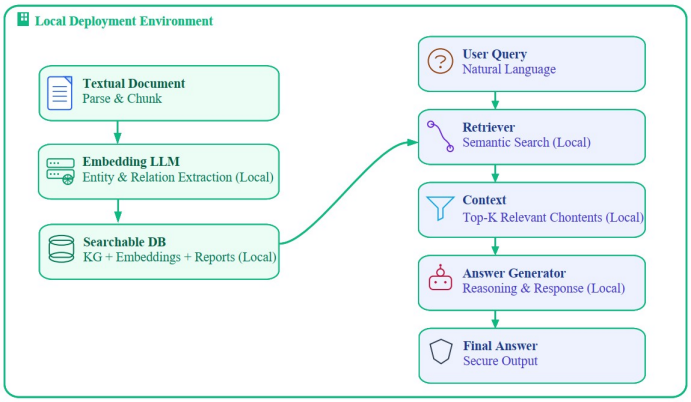

Figure 1: Workflow of the knowledge graph–enhanced retrieval-augmented generation (RAG) LLMs framework for privacy-preserving AEC document interpretation.

A set of carefully designed prompts is then used to convert these chunks into structured knowledge. These prompts follow a consistent strategy: each one clearly defines the model's role, states the extraction goal, and provides step-by-step instructions. Strict output formats ensure that entities, relationships, and claims are produced in predictable schemas, while grounding rules restrict the model to information explicitly supported by the text. Together, these strategies guide the model through a logical pipeline: first identifying entities, then extracting relationships, then capturing claims, and finally grouping related entities into thematic communities. This process produces structured, reliable information rather than free-form text. The extracted entities, relationships, community summaries, and source-aligned metadata are then assembled into a unified knowledge graph. This graph serves as a structured semantic layer over the source material, enabling efficient retrieval, graph-based reasoning, and contextually grounded querying within the framework.

When a user submits a natural-language query, a local embedding model encodes the query into a vector representation. A retrieval module then performs semantic similarity matching between the query and elements of the knowledge graph (KG), whose nodes and edges represent structured entities, relationships, and claims derived from the source text. Node embeddings incorporate both semantic content and the KG's structural properties, such as relationship types and community groupings, allowing the retrieval process to leverage not only textual similarity but also graph-based contextual relevance. For example, entities closely connected within a community or linked by specific relationships are more likely to be retrieved together when relevant to the query. The candidate segments are filtered and ranked based on their semantic alignment and contextual

completeness. The selected information, along with the user query, is passed to a locally deployed LLM, which synthesizes a concise, domain-consistent response grounded exclusively in the input document. By integrating text parsing, knowledge graph construction, semantic retrieval, and response generation within a self-contained, on-premise environment, the framework provides an accurate, explainable, and privacy-preserving solution for automated document interpretation across a wide range of AEC texts.

EXPERIMENTS

To validate the proposed framework, building code documents were selected as the test material. A domain-specific question-answering dataset was constructed, consisting of 661 entries derived from Chapters 5 and 10 of the 2015 International Building Code (IBC) (International Code Council, 2014). Building on earlier work by Xue et al. (2024), each record comprises a triplet of context, question, and answer, manually created to reflect detailed interpretations of regulatory provisions. Contexts were selected based on their specificity, with individual subsections treated as distinct segments where applicable. For each context, one or more question-answer pairs were generated, provided the content was sufficiently informative. Provisions that lacked enough detail to support meaningful questions were excluded to maintain the dataset's relevance and clarity.

The dataset is formatted according to the widely adopted Stanford Question Answering Dataset (SQuAD) structure (Rajpurkar et al., 2016), ensuring compatibility with standard evaluation tools and models. As shown in Table 1, the average word count is 18.38 for questions, 108.65 for contexts, and 4.28 for answers. This balance of granularity and conciseness supports precise system evaluation and meaningful performance assessment. Overall, the dataset offers a robust, structured resource for benchmarking LLM-based question-answering systems tailored to AEC regulatory documents.

Table 1: Average word count analysis of question answering dataset.

Attributes	Question	Context	Answer
Average word numbers	18.38	108.65	4.28

Chapters 5 and 10 of the IBC 2015 were first compiled into a single PDF document and converted into plain text using Marker, a lightweight and structure-preserving PDF parser (Paruchuri, 2025). The resulting text was segmented into chunks of 100 tokens, with a 20-token overlap to preserve contextual continuity across boundaries. This chunk size was

selected to approximate the typical length of individual clauses found in building code documents, ensuring that each segment captured a coherent and self-contained unit of meaning. Each chunk was embedded using the nomic-embed-text model (nomic-embed-text, 2025), chosen for its efficiency and accuracy in capturing both semantic and syntactic relationships within AEC regulatory texts. User queries were embedded using the same model to ensure consistent vector representations for retrieval.

To support context selection, the system implemented a graph-based retrieval-augmented generation mechanism. All embedded content was organized into a knowledge graph comprising 234 entities, 131 relationships, and 8 communities, where nodes represent regulatory entities or text segments, and edges denote semantic or logical relationships among them. To further structure the graph, related nodes and edges were clustered into communities based on their connectivity and thematic relevance, facilitating more efficient traversal and interpretation. This graph structure allows the retrieval component to surface not only direct matches but also semantically connected provisions, thereby enriching the contextual basis for answer generation. The embedding and graph construction processes were performed once during system initialization.

For response generation, the retrieved context and user query were passed to DeepSeek-R1:70B (deepseek-r1:70b, 2025), a high-capacity open-source LLM selected for its strong performance in natural language generation tasks. All components, including retrieval and generation, were deployed entirely on-premise within a secure computing environment, which consisted of an AMD Ryzen Threadripper PRO 7975WX (32 cores at 4.00 GHz), 128 GB RAM, and an NVIDIA RTX 6000 Ada Generation GPU. This self-hosted setup ensures full control over data security, avoiding reliance on third-party APIs or cloud services.

After initialization, all 661 queries in the evaluation dataset were independently processed through this pipeline to retrieve relevant context and generate corresponding answers.

An illustrative example of the system's behavior is shown in Table 2. For the question asking which section governs the height design of unlimited area buildings, the generated answer ("Section 507") matches the gold standard answer exactly. However, the generated context refers to Section 506.1.1, while the gold standard context points to Section 504.1.1. Upon inspection, both provisions explicitly stated that the design must comply with Section 507. This outcome demonstrates the system's ability to recognize semantically equivalent sources across structurally different sections of the document, made possible through graph-based retrieval. Rather than relying solely on exact section matches, the system identifies and integrates conceptually aligned content, enhancing its robustness in interpreting complex regulatory language.

Table 2: Example of generated answers and contexts and its comparison with gold standard.

Attributes	Content
Question	What is the section that the height of unlimited area buildings shall be designed in accordance with?
Context	504.1.1 Unlimited area buildings. The height of unlimited area buildings shall be designed in accordance with Section 507.
Answer	Section 507
Generated answer	The height of unlimited area buildings shall be designed in accordance with Section 507.
Generated context	This information is found in Section 506.1.1 of the building codes, which explicitly states that unlimited area buildings must comply with the design requirements outlined in Section 507.

To evaluate the quality of generated answers, two primary metrics were used: answer semantic similarity and answer relevance, both scored on a scale from 0 to 1. The answer semantic similarity metric quantifies how closely the generated answer matches the gold standard in meaning and phrasing. It is computed by embedding both the generated and reference answers using a specified embedding model and calculating the cosine similarity between their vector representations. In the example from Table 2, the generated answer (i.e., "The height of unlimited area buildings shall be designed in accordance with Section 507.") aligned well with the gold standard answer "Section 507", resulting in a semantic similarity score of 0.88.

The second metric, answer relevance, assesses how well the generated answer addresses the user's original query. This involves generating multiple artificial questions based on the generated answer, computing the cosine similarity between each of these questions and the original query, and averaging the scores. For the same example, the answer relevance score was 0.93, indicating that the generated response was highly aligned with the user's intent. These evaluation procedures were applied to all 661 records in the dataset, and the aggregated results are presented in the following section.

RESULTS

The generated answers and contexts for all 661 queries were recorded and analyzed. Table 3 presents the average word count statistics for these outputs. On average, the generated answers contain 26.93 words, which is notably longer than the concise gold standard answers that typically provide only the essential information needed to address each query. This increase in length reflects the framework's design, which favors completeness and explanatory depth in generated responses. The generated contexts, averaging 47.51 words, are also more concise than the original regulatory excerpts. This is because the reference contexts are drawn directly from the building code and often include broader sections that may address multiple topics. In contrast,

the generated contexts are synthesized summaries that integrate information from semantically related provisions across the document, offering focused, query-specific support for the generated answers.

Table 3: Average word count for generated answers and contexts.

Attributes	Generated Answer	Generated Context
Average word numbers	26.93	47.51

To evaluate the framework's performance across the full set of queries, two key metrics were used: semantic similarity and answer relevance, as summarized in Table 4. The average semantic similarity score was 0.83, indicating that the generated answers closely matched the gold standard answers in terms of meaning and contextual alignment. The answer relevance score averaged 0.71, reflecting a strong correspondence between the generated answers and the user queries. The difference between the two metrics is reasonable given how they are computed: semantic similarity directly compares the embeddings of the generated and reference answers, while answer relevance is obtained by generating multiple artificial questions from the model's answer and comparing each of them to the original query. As a result, if the generated answer shifts slightly away from the user's intent, or if it contains inaccuracies, the artificial questions derived from it will diverge more from the original query, leading to a lower relevance score. Even with this stricter evaluation method, the relevance score remains high, demonstrating that the framework produces responses that are both semantically accurate and well aligned with user intent, while maintaining privacy through a fully local deployment.

Table 4: Performance metrics for the proposed framework.

Metrics	Semantic Similarity	Answer Relevance
Score	0.83	0.71

CONCLUSION

This study presents a knowledge graph–enhanced retrieval-augmented generation (RAG) LLMs framework designed for local, privacy-preserving processing of textual documents in the AEC domain. By integrating open-source tools for embedding and response generation, the system enables on-premise interpretation of complex regulatory texts, such as building codes, without transmitting data to external servers. Experimental evaluation using 661 query-answer-context records derived from the 2015 International Building Code demonstrated the framework's effectiveness in producing accurate and contextually relevant responses. With an average semantic similarity score of 0.83 and an answer relevance score of 0.71, the system

reliably captured both the intended meaning of the reference answers and their alignment with user queries. These findings highlight the viability of combining knowledge graphs and large language models for secure, scalable document understanding in data-sensitive domains, providing a strong foundation for practical applications in regulatory compliance, design validation, and knowledge management within the AEC sector.

Despite the promising results, this study has several limitations. First, the framework was tested on a subset of the building code, and while the selected chapters provide a representative sample, broader coverage across diverse AEC document types, such as contracts, specifications, and safety guidelines, is needed to assess generalizability. Second, the system was deployed on a high-performance machine to enable on-premise execution of large language models. While this setup ensured full data privacy and responsiveness, it also represents a potential limitation for organizations without access to comparable computational resources. Future work will explore more resource-efficient deployment strategies, including model compression and modular inference, to support broader adoption. Future work will also include benchmarking key performance metrics, such as inference latency and memory usage, to better characterize the system's computational requirements. In addition, ongoing efforts will address other limitations by advancing automated knowledge graph construction, incorporating human-in-the-loop evaluation, and fine-tuning LLMs on AEC-specific corpora to improve domain alignment, interpretability, and responsiveness across a wider range of document types.

ACKNOWLEDGMENT

The authors would like to thank the National Science Foundation (NSF). This material is based on work supported by the NSF under Grant No. 1827733. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the NSF. During the preparation of this work, the authors used OpenAI's GPT-4o to assist with improving readability and language. All content was subsequently reviewed and edited by the authors, who take full responsibility for the final version of the publication.

REFERENCES

British Standards Institution. (2020) . *ISO 19650–5:2020 ISO*. Available at: https://www.iso.org/standard/74206.html (Accessed: October 24, 2025).

deepseek-r1:70b. Available at: https://ollama.com/deepseek-r1:70b (Accessed: April 24, 2025).

nomic-embed-text. Available at: https://ollama.com/nomic-embed-text (Accessed: April 24, 2025).

Fuchs, S. (2021) "Natural Language Processing for Building Code Interpretation: Systematic Literature Review Report.".

Fuchs, S. *et al.* (2024) "Using Large Language Models for the Interpretation of Building Regulations." arXiv. Available at: https://doi.org/10.48550/arXiv.2407.21060.

International Code Council (ed.) (2014) *International Building Code 2015 IBC*. Country Club Hills, Ill: International Code Council.

Paruchuri, V. (2025) "VikParuchuri/marker." Available at: https://github.com/VikParuchuri/marker (Accessed: February 24, 2025).

Rajpurkar, P. *et al*. (2016) "SQuAD: 100, 000+ Questions for Machine Comprehension of Text." arXiv. Available at: https://doi.org/10.48550/arXiv.1606.05250.

The Construction Index. (2016). *Cyber criminals target construction*. Available at: https://www.theconstructionindex.co.uk/news/view/cyber-criminals-target-construction (Accessed: October 24, 2025).

Xue, X., Zhang, J. and Chen, Y. (2024) "Question-answering framework for building codes using fine-tuned and distilled pre-trained transformer models," *Automation in Construction*, 168, p. 105730. Available at: https://doi.org/10.1016/j.autcon.2024.105730.

Yang, F. and Zhang, J. (2024) "Prompt-based automation of building code information transformation for compliance checking," *Automation in Construction*, 168, p. 105817. Available at: https://doi.org/10.1016/j.autcon.2024.105817.

Yao, D. and García de Soto, B. (2024) "Enhancing cyber risk identification in the construction industry using language models," *Automation in Construction*, 165, p. 105565. Available at: https://doi.org/10.1016/j.autcon.2024.105565.

Zhang, J. and El-Gohary, N. M. (2016) "Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking," *Journal of Computing in Civil Engineering*, 30(2), p. 04015014. Available at: https://doi.org/10.1061/(ASCE) CP.1943–5487.0000346.

Zhang, R. and El-Gohary, N. (2019) "A machine learning-based method for building code requirement hierarchy extraction," in. *Proceedings, Annual Conference-Canadian Society for Civil Engineering*, pp. 1–10.

Zhong, B. *et al*. (2020) "Deep learning-based extraction of construction procedural constraints from construction regulations," *Advanced Engineering Informatics*, 43, p. 101003. Available at: https://doi.org/10.1016/j.aei.2019.101003.