An LLM-Based Framework with Retrieval-Augmented Generation for Building Code Interpretation

Fan Yang[1], Yiru Hou[2] and Jiansong Zhang[3]*

[1] School of Construction Management Technology, Purdue University, 363 N. Grant Street
West Lafayette, IN 47907, USA. ORCID: https://orcid.org/0000-0001-9842-719X
[2] School of Construction Management Technology, Purdue University, 363 N. Grant Street
West Lafayette, IN 47907, USA
[3] School of Construction Management Technology, Purdue University, 363 N. Grant Street
West Lafayette, IN 47907, USA. ORCID: https://orcid.org/0000-0001-5225-5943

ABSTRACT: Building codes are inherently complex and challenging to understand due to their detailed and interrelated safety requirements, technical language, and regional variations. Traditional manual approaches, such as consulting building code experts, are effective but constrained by high labor costs and the frequent updates to building codes. The advent of large language models (LLMs) offers a promising solution for automating and intelligently interpreting building codes. However, LLMs can sometimes produce unrelated or incorrect information due to hallucinations. In this paper, the authors propose a method that combines LLMs with retrieval-augmented generation (RAG) techniques to develop a "building code expert" capable of accurately answering user queries about building codes. To validate the approach, a dataset of 150 data records - each consisting of a query, an answer, and the relevant context - was extracted from Chapters 5 and 10 of the International Building Code 2015. Experimental results demonstrate that the proposed method outperforms the state-of-the-art question-answering framework. This research provides a notable step forward in leveraging artificial intelligence (AI) to improve accessibility, accuracy, and efficiency in understanding building codes, with potential applications such as compliance checking, automated design validation, and risk assessment.

1. INTRODUCTION

Building codes are inherently complex and challenging to understand, as they encompass detailed and interrelated requirements designed to ensure the usability of buildings and benefits of stakeholders. This complexity is further heightened by the use of technical language and regional variations, which can create additional barriers to interpretation and compliance. Although traditional human building code experts are effective in interpreting and applying regulations, their expertise comes with high labor costs. As building codes vary across regions and are frequently updated, maintaining up-to-date knowledge requires continuous training, leading to additional expenses. These ongoing costs can pose significant challenges for organizations striving to ensure compliance while managing limited resources.

Large Language Models (LLMs) stand out in the field of Artificial Intelligence (AI) due to their advanced natural language processing capabilities, which offers a promising avenue for developing intelligent and user-friendly building code expert system. These AI-driven systems have the potential to address the limitations of traditional human experts by reducing labor costs and providing consistent, easily accessible answers. However, universal LLMs are typically trained on diverse datasets, such as Wikipedia, books, and movie reviews, which can limit their accuracy when handling domain-specific queries. As a result, they may produce irrelevant or incorrect responses - a phenomenon known as hallucination - posing challenges for their direct applications in specialized fields like building code interpretation.

Retrieval-Augmented Generation (RAG) technology has been proposed to address the limitations of LLMs, particularly their tendency to produce inaccurate or irrelevant responses when handling domain-specific queries (Lewis et al., 2020). RAG enhances the capabilities of LLMs by integrating them with an external knowledge retrieval system, allowing the model to access and incorporate up-to-date, domain-relevant information during the response generation process. Instead of relying solely on pre-trained knowledge, RAG retrieves specific documents or data from curated sources, grounding its outputs in accurate and context-specific information. This approach significantly reduces the risk of hallucination and improves the reliability of LLMs, making them better suited for specialized applications, such as interpreting complex and evolving building codes.

In this paper, the authors present an LLM-based framework enhanced with RAG to develop an intelligent "building code expert" capable of accurately responding to user queries about building codes. By leveraging RAG, the framework integrates the natural language understanding of LLMs with real-time access to domain-specific information, ensuring both accuracy and relevance in its responses. To evaluate the feasibility and performance of the proposed framework, a dataset of 150 question-and-answer pairs was compiled from Chapters 5 and 10 of the International Building Code 2015 (IBC2015) (International Code Council, 2014), covering key structural and safety regulations. The framework's performance was quantitatively assessed using three metrics: answer accuracy, semantic similarity between generated and reference answers, and completion time. Experimental results demonstrate that the proposed framework achieves higher answer accuracy compared to state-of-the-art machine learning-based approaches, highlighting its effectiveness in addressing the complexities of building code interpretation.

The proposed framework marks a promising step forward in leveraging AI to improve the accessibility, accuracy, and efficiency of interpreting complex building codes. With its ability to provide precise and user-friendly guidance, the framework has broad potential applications, including compliance checking, automated design validation, and risk assessment. By reducing reliance on human experts for routine code interpretation and minimizing the risk of errors, it lowers operational costs while promoting safer and more efficient building practices. This study underscores the value of AI-driven tools in bridging the gap between complex regulations and practical industry needs.

2. LITERATURE REVIEW

2.1 Building Code Interpretation

Building code interpretation involves understanding and applying regulations to ensure construction projects comply with safety, health, and environmental standards. The primary outputs of this process, particularly in automated systems, include compliance reports that assess a design's adherence to relevant codes and highlight areas of non-compliance, along with actionable recommendations to guide necessary design modifications (Eastman et al., 2009).

Various methodologies have been developed to tackle the complexities of code interpretation, each with distinct strengths and limitations. Rule-based approaches rely on explicitly defined rules derived from building codes, enabling systematic assessments of design compliance. While these methods produce clear and interpretable results, their rigidity often limits their ability to handle the nuanced and evolving nature of modern codes. For instance, Zhang and El-Gohary (2016) introduced a semantic, rule-based natural language processing approach that uses pattern-matching information extraction rules, conflict resolution mechanisms, and ontology-supported semantic feature recognition to achieve high precision and recall in extracting quantitative requirements from the 2009 International Building Code.

Machine learning-based approaches offer greater flexibility by inferring compliance rules from datasets containing examples of building designs and their compliance statuses. This data-driven method can adapt to complex code applications, though its success depends heavily on the availability of extensive, high-quality training data. For example, R. Zhang and El-Gohary (2019) developed a machine learning-based method to automatically decompose complex building code requirements into manageable units. Using techniques such as data preparation, deep neural network-based dependency parsing, and requirement segmentation, the approach demonstrated strong precision and recall when tested on the 2009 International Building Code and the Champaign 2015 IBC Amendments.

Building on these foundations, LLM-based approaches leverage Large Language Models to directly interpret building code texts through advanced natural language processing. These models provide dynamic, context-aware interpretations, enabling detailed analyses, explanations, and tailored recommendations based on a deeper understanding of the language and intent within the codes. While traditional rule-based and machine learning methods have laid a solid foundation for automated building code interpretation, LLM-based approaches offer a more flexible and context-sensitive solution, better equipped to handle the complexity and variability inherent in modern building codes.

2.2 Large Language Models

Large Language Models (LLMs) are advanced machine learning models designed to understand and generate human-like text based on the training they receive from vast amounts of textual data. The development of Large Language Models began with rule-based systems in the mid-20th century, transitioning to statistical models like n-grams and Hidden Markov Models by the 1990s (Eddy, 1996). The resurgence of neural networks in the 2000s led to the creation of word embeddings and recurrent neural networks (Lipton et al., 2015), setting the stage for the transformative introduction of transformer architectures in 2017 (Vaswani et al., 2017). This breakthrough enabled the development of models like OpenAI's GPT series (*Models - OpenAI*, 2025) and Google's BERT (*BERT*, 2025), which have significantly advanced natural language processing capabilities and are widely implemented in various practical applications, such as chatbots, automated content generation, sentiment analysis, and enhanced search engine functionalities (Naveed et al., 2024).

LLMs have significantly enhanced project management and efficiency in the Architecture, Engineering, and Construction (AEC) industry by streamlining various construction tasks. One example is the development of the Natural-language-based Architectural Detailing through Interaction with AI (NADIA) system, which advances civil structure design by enabling natural language-driven architectural detailing. This innovative LLM-BIM chaining framework achieved an 83.33% accuracy rate in producing exterior wall details that meet specific design requirements(Jang et al., 2024). Similarly, the BIMS-GPT framework - a dynamic, GPT-based virtual assistant - facilitates natural language-driven searches within building information models (BIM), achieving an impressive 99.5% accuracy in query classification while using minimal data, thereby greatly enhancing BIM accessibility in the construction industry (Zheng & Fischer, 2023). In another study, LLMs were employed to convert building code information into a logic programming language using a prompt-based framework, significantly improving both accuracy and efficiency. This method achieved 97.37% precision and 95.88% recall when applied to International Building Code requirements, while also substantially reducing code generation time and minimizing manual effort (Yang & Zhang, 2024). These examples highlight the transformative potential of LLMs in enhancing efficiency, accuracy, and accessibility across various construction processes within the AEC industry.

While LLMs offer significant advantages in the AEC industry, they still face key limitations, particularly the issue of hallucination - the generation of convincing but inaccurate answers due to limited domain-specific data and knowledge, which can lead to erroneous decisions (Huang et al., 2025). Additionally, LLMs trained on historical data often struggle to adapt to modern construction practices, making them less effective for handling state-of-the-art changes (Pulkkinen, 2024).

2.3 Retrieval-Augmented Generation

Retrieval-Augmented Generation (RAG) is a technology that enhances language models by combining the generative capabilities of models like the Transformer with retrieval-based techniques (Lewis et al., 2020). This hybrid approach allows the model to generate responses based not only on fixed parameters learned during training but also on dynamically retrieved information relevant to the task at hand. Essentially, RAG extends the Transformer architecture by integrating a retriever component that fetches relevant document snippets from a large corpus, which are then used by the generator to produce informed and contextually enriched outputs.

The concept of RAG was primarily developed to address limitations in pure generative models, such as GPT, which can generate plausible but sometimes factually incorrect or uninformative responses. The development of RAG was inspired by earlier works that incorporated external knowledge bases into neural networks to enhance their information recall capabilities. By integrating retrieval mechanisms directly into the generation process, RAG allows models to leverage both parametric and non-parametric knowledge, leading to improvements in the quality and reliability of generated text. The model was further popularized by its implementation in various machine learning frameworks, notably by Hugging Face's Transformers library, which made it accessible for a wide range of applications and research experiments, such as Question Answering system (Xu et al., 2024), Medical Diagnostics (Zelin et al., 2024), and Legal and Financial Services (Pipitone & Alami, 2024).

Within the AEC domain, RAG technology is starting to be recognized for its potential to transform several critical operations, such as contract risks analysis (Shuai & Caldas, 2024), construction safety(Jeong et al., 2024), construction management (Wu et al., 2025). RAG's ability to dynamically integrate retrieved data with generative modeling opens up new avenues for improving accuracy and efficiency in the AEC sector and beyond. As this technology continues to evolve, its integration into industry-standard tools is anticipated to grow, further enhancing its utility and transforming traditional workflows.

3. METHODOLOGY

The LLM-based framework with RAG for building code interpretation is illustrated in Figure 1. This framework uses building codes as an external information source for LLMs. A PDF parser is first used to convert building codes from PDF format into plain text, facilitating the embedding process. An LLM-based embedder then transforms the textual data into embeddings, which are vector space representations designed to capture semantic and syntactic relationships. Simultaneously, the embedder converts user queries from natural language into corresponding embeddings. Since both the building codes and user queries are represented in the same vector space, a searching algorithm can be applied to identify relevant content within the building codes. The retrieved content is then re-ranked based on relevance and combined with the original user query before being fed into another LLM to generate the final response. Because the LLM generates responses grounded in the retrieved content, the resulting answers are contextually relevant and accurate.

To evaluate the performance of the proposed framework, three metrics were used: answer accuracy, semantic similarity, and completion time. Answer accuracy is a binary metric that assesses the extent to which a generated answer aligns with a predefined gold standard. A score of 1 is assigned if the generated answer fully covers the gold standard, while a score of 0 indicates incomplete coverage. This metric focuses on evaluating the completeness of responses. Semantic similarity measures the degree of semantic alignment between the generated answer and the gold standard, using a scale from 0 to 1. A higher score indicates a stronger alignment in meaning and contextual relevance. This metric is essential for determining how effectively the generated response captures the intent and nuances of the expected answer. Completion time records the duration, in seconds, from when a query is submitted to when the complete answer is produced. This metric reflects the system's efficiency and responsiveness, with shorter times indicating higher performance. This is particularly valuable in real-time applications where timely decision-making is critical.

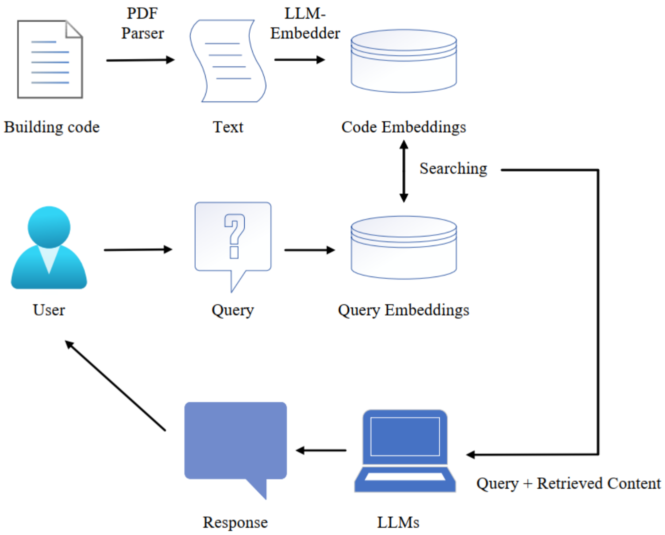

Figure 1. The LLM-based framework with RAG for building code interpretation.

4. EXPERIMENT

4.1 Dataset Preparation

To evaluate the performance of the framework, a question-answering dataset (Xue et al., 2024) was constructed using content from Chapters 5 and 10 of the IBC2015, which focus on building heights, areas, and means of egress. The dataset comprises 150 records, each consisting of a question, its corresponding answer, and the relevant context from the building code. When selecting regulations for data generation, the most specific provisions were identified as context. In cases where sections contained multiple subsections, each subsection was treated as a distinct context to create individual or multiple question-answer pairs. This approach ensures comprehensive coverage of the building code while maintaining concise and focused questions and answers.

The dataset was designed to follow the Stanford Question Answering Dataset (SQuAD) format (Rajpurkar et al., 2016), promoting standardization and enhancing its applicability for question-answering systems. The average word counts for questions, contexts, and answers are summarized in Table 1. On average, questions contain 18.25 words and typically begin with interrogative phrases such as "What…?" The context represents the original text extracted from the building code regulations, with an average length of 110.67 words. Answers are presented in the most concise form possible while still fully addressing the corresponding question, averaging 4.25 words. This structured dataset supports effective evaluation of the framework’s ability to interpret building codes accurately and efficiently.

Table 1: Average word count analysis of question answering dataset

Attributes	Question	Context	Answer
Average word numbers	18.25	110.67	4.25

4.2 Experimental Implementation

Chapters 5 and 10 of the IBC 2015, provided in PDF format, were extracted and merged into a single PDF document. A PDF parser, Marker, was utilized to convert this document into a markdown file, which was subsequently processed into a plain text format. Marker is a simple yet efficient tool for PDF conversion, facilitating the extraction of structured text for further analysis (Paruchuri, 2023/2025). The converted text was then embedded using text-embedding-3-small, a lightweight embedding model developed by OpenAI (*New Embedding Models and API Updates*, 2024). This model transforms text into dense vector representations that capture both semantic and syntactic relationships, enabling effective information retrieval. User queries were similarly embedded using the same model to ensure consistency in the retrieval process. To optimize the search and retrieval of relevant content from the building code, the framework employed GraphRAG (*GraphRAG*, 2025), an advanced RAG architecture that integrates graph-based data structures with traditional RAG frameworks. GraphRAG represents knowledge as interconnected nodes (entities) and edges (relationships), allowing for more accurate context retrieval by capturing complex dependencies within the data. This graph-enhanced retrieval ensures that related nodes and edges are identified and used to generate a comprehensive context for each query. Once the relevant content was retrieved and the contextual text generated, it was combined with the user's query and fed into another LLM, GPT-4o (*GPT-4o*, 2025), for response generation. This process ensures that the generated answers are both contextually accurate and semantically aligned with the building code.

During the framework setup, the embedding process for Chapters 5 and 10 was conducted only once. After this initial setup, 150 selected queries were individually processed through the framework to retrieve relevant content and generate corresponding answers. Table 2 presents an example entry from the experimental process, focusing on regulations for revolving doors used as components of a means of egress. Compared to the reference answer, the generated answer is longer and more detailed, providing a fuller explanation. Additionally, the generated context is not merely a direct retrieval from the building code but is synthesized using multiple nodes and relationships extracted from the original building code, offering a more comprehensive and interconnected view of the regulations.

Table 2: Example entry from the question answering dataset

Attributes	Content
Question	What type of door is used as a component of a means of egress and must comply with specific conditions regarding width, occupant load, and breakout force?
Context	1010.1.4.1.1 Egress component. A revolving door used as a component of a means of egress shall comply with Section 1010.1.4.1 and the following three conditions: 1. Revolving doors shall not be given credit for more than 50 percent of the minimum width or required capacity. 2. Each revolving door shall be credited with a capacity based on not more than a 50-person occupant load. 3. Each revolving door shall provide for egress in accordance with BHMA A156.27 with a breakout force of not more than 130 pounds (578 N).
Answer	Revolving doors
Generated answer	Revolving doors are used as a component of a means of egress and must comply with specific conditions regarding width, occupant load, and breakout force.
Generated context	Revolving doors must have a side-hinged swinging door nearby as per Section 1010.1, and they must comply with the requirements in Table 1010.1.4.1(2) for automatic or power-operated revolving doors.

To evaluate the generated answer, an independent LLM, GPT-4o, was used to determine whether the generated answer fully covered the reference answer. A score of 1 was assigned if the generated answer completely matched the reference answer; otherwise, a score of 0 was given. In the example shown in Table 2, the generated answer fully covered the reference answer, “Revolving doors”, resulting in a score of 1. To further assess the semantic accuracy of the generated answer, semantic similarity was calculated using a cross-encoder model. The process involved three steps. First, the reference answer was vectorized using the specified embedding model. Second, the generated answer was vectorized using the same embedding model to ensure consistency. Finally, the cosine similarity between the two vectors was computed to quantify their semantic alignment. This evaluation was implemented using a Python program. For the example in Table 2, the semantic similarity between the generated answer and the reference answer was 0.8738, indicating a high level of semantic similarity. The efficiency of the framework was also evaluated by measuring the completion time, which records the duration (in seconds) from query input to answer generation, as tracked by GraphRAG. For the example in Table 2, the completion time was 3.11 seconds. This evaluation process was applied to all 150 queries, and the results are discussed in the subsequent section.

5. RESULTS

The generated answers and contexts for all 150 queries were recorded and analyzed. Table 3 presents the average word count for both. Compared to the reference answers, the generated answers are significantly longer, with an average of 24.37 words. This difference arises because the reference answers are intentionally concise, containing only the essential information needed to address the query, while the generated answers are designed to be more complete and explanatory. In contrast, the generated contexts are shorter on average than the reference contexts, with 33.72 words compared to the often-lengthier building code provisions. This is because the reference contexts are direct excerpts from the building code, often covering broader sections that may address multiple queries. The generated contexts, however, are synthesized summaries that draw from multiple related sources within the building code, providing concise, query-specific information that directly supports the generated answers.

Table 3: Average word count for generated answers and contexts in dataset

Attributes	Generated answer	Generated context
Average word numbers	24.37	33.72

For the evaluation of the framework across the selected queries, Table 4 presents key performance metrics, including answer accuracy, semantic similarity, and completion time. The framework achieved an average answer accuracy of 0.747, outperforming the state-of-the-art deep learning and NLP-based question-answering (QA) system, which reports a top-1 exact match accuracy of 0.63 (Xue et al., 2024). This indicates the framework's superior capability in producing precise answers. Additionally, the average semantic similarity score of 0.857 demonstrates the framework's effectiveness in generating answers that closely align with the reference answers in terms of meaning and context. The average completion time of 2.610 seconds highlights the system's efficiency in generating timely responses, making it suitable for real-time or near-real-time applications.

Table 4: Performance metrics for question answering system

Metrics	Answer accuracy	Semantic similarity	Completion time (s)
Score/Duration	0.747	0.857	2.610

These results confirm that the proposed framework not only improves answer accuracy compared to existing QA systems but also maintains a high level of semantic relevance while delivering responses efficiently. Future work could focus on further optimizing completion time and exploring methods to enhance the contextual depth of generated answers without sacrificing conciseness. The strong performance across all metrics suggests the framework's potential for broader application in regulatory document interpretation and other complex domains requiring accurate and contextually grounded question answering.

6. CONCLUSIONS

This study presents an LLM-based framework enhanced with RAG for building code interpretation, addressing the complexities and challenges associated with understanding and applying regulatory documents like the IBC2015. By integrating domain-specific knowledge retrieval with the natural language understanding capabilities of LLMs, the proposed framework effectively mitigates common issues found in universal LLMs, such as hallucination and irrelevant responses, thereby improving both the accuracy and reliability of the answers generated.

The evaluation of the framework, conducted using a dataset of 150 queries, demonstrated its strong performance across multiple metrics. The framework achieved an average answer accuracy of 0.747, surpassing the performance of state-of-the-art deep learning and NLP-based question-answering systems. Additionally, the semantic similarity score of 0.857 highlights the framework's ability to produce contextually accurate answers that closely align with the intended meaning of the reference responses. The average completion time of 2.610 seconds further emphasizes the system's efficiency, making it viable for real-time applications where timely decision-making is critical. Beyond its strong quantitative performance, the framework offers practical advantages in building code interpretation by generating concise, query-specific contexts that streamline complex regulatory information. This not only reduces reliance on human experts but also lowers operational costs and enhances accessibility for users with varying levels of expertise. The system's ability to synthesize and summarize information from multiple sources within the building code ensures that responses are both comprehensive and directly relevant to user queries.

Future work will focus on several key areas to further enhance the framework's performance and adaptability. One priority is optimizing the completion time to improve efficiency, particularly for large-scale or real-time applications. Additionally, efforts will be made to refine the generation of context by incorporating more advanced graph-based retrieval methods, enabling deeper contextual understanding and more nuanced responses. To address the current limitation of dataset scope, the authors plan to expand the dataset beyond the initial 150 Q&A pairs by incorporating a wider range of questions and answers derived from multiple building codes, including region-specific regulations and their updates. This expansion will facilitate more comprehensive validation and strengthen the framework's generalizability and relevance across diverse compliance scenarios. Furthermore, integrating user feedback mechanisms could help fine-tune the system's accuracy and relevance over time. Finally, extending the framework's capabilities to other domains that require precise interpretation of complex regulatory documents, such as legal codes or environmental regulations, represents a promising direction for broader impact.

ACKNOWLEDGMENTS

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.

REFERENCES

BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (2025). Retrieved April 7, 2025, from https://research.google/pubs/bert-pre-training-of-deep-bidirectional-transformers-for-language-understanding/

Eastman, C., Lee, J., Jeong, Y., & Lee, J. (2009). Automatic rule-based checking of building designs. *Automation in Construction*, *18*(8), 1011–1033. https://doi.org/10.1016/j.autcon.2009.07.002

Eddy, S. R. (1996). Hidden Markov models. *Current Opinion in Structural Biology*, *6*(3), 361–365. https://doi.org/10.1016/S0959-440X(96)80056-X

Hello GPT-4o. (2025). Retrieved February 24, 2025, from https://openai.com/index/hello-gpt-4o/

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., & Liu, T. (2025). A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems*, *43*(2), 1–55. https://doi.org/10.1145/3703155

International Code Council (Ed.). (2014). *International Building Code 2015 IBC*. International Code Council.

Jang, S., Lee, G., Oh, J., Lee, J., & Koo, B. (2024). Automated detailing of exterior walls using NADIA: Natural-language-based architectural detailing through interaction with AI. *Advanced Engineering Informatics*, *61*, 102532. https://doi.org/10.1016/j.aei.2024.102532

Jeong, M., Kim, T., Kim, S., & Kim, H. (2024). Retrieval-Augmented Generation-based Question Answering Technology for Construction Safety. *International Conference on Construction Engineering and Project Management*, 439–446. https://doi.org/10.6106/ICCEPM.2024.0439

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, *33*, 9459–9474. https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). *A Critical Review of Recurrent Neural Networks for Sequence Learning* (arXiv:1506.00019). arXiv. https://doi.org/10.48550/arXiv.1506.00019

Models—OpenAI API. (2025). Retrieved April 7, 2025, from https://platform.openai.com

Naveed, H., Khan, A. U., Qiu, S., Saqib, M., Anwar, S., Usman, M., Akhtar, N., Barnes, N., & Mian, A. (2024). *A Comprehensive Overview of Large Language Models* (arXiv:2307.06435). arXiv. https://doi.org/10.48550/arXiv.2307.06435

New embedding models and API updates. (2024, March 13). https://openai.com/index/new-embedding-models-and-api-updates/

Paruchuri, V. (2025). *VikParuchuri/marker* [Python]. https://github.com/VikParuchuri/marker (Original work published 2023)

Pipitone, N., & Alami, G. H. (2024). *LegalBench-RAG: A Benchmark for Retrieval-Augmented Generation in the Legal Domain* (arXiv:2408.10343). arXiv. https://doi.org/10.48550/arXiv.2408.10343

Pulkkinen, T. (2024). *Generative AI for identifying conflicts in construction industry documents*. https://aaltodoc.aalto.fi/handle/123456789/130112

Rajpurkar, P., Zhang, J., Lopyrev, K., & Liang, P. (2016). *SQuAD: 100,000+ Questions for Machine Comprehension of Text* (arXiv:1606.05250). arXiv. https://doi.org/10.48550/arXiv.1606.05250

Shuai, B., & Caldas, C. H. (2024). *A Case-Based Rag Methodology to Analyze Contract Risks for Construction Projects* (SSRN Scholarly Paper 4946907). Social Science Research Network. https://doi.org/10.2139/ssrn.4946907

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is All you Need. *Advances in Neural Information Processing Systems*, *30*. https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

Welcome—GraphRAG. (2025). Retrieved February 24, 2025, from https://microsoft.github.io/graphrag/

Wu, C., Ding, W., Jin, Q., Jiang, J., Jiang, R., Xiao, Q., Liao, L., & Li, X. (2025). Retrieval augmented generation-driven information retrieval and question answering in construction management. *Advanced Engineering Informatics*, *65*, 103158. https://doi.org/10.1016/j.aei.2025.103158

Xu, Z., Cruz, M. J., Guevara, M., Wang, T., Deshpande, M., Wang, X., & Li, Z. (2024). Retrieval-Augmented Generation with Knowledge Graphs for Customer Service Question Answering. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2905–2909. https://doi.org/10.1145/3626772.3661370

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

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

Zelin, C., Chung, W. K., Jeanne, M., Zhang, G., & Weng, C. (2024). Rare disease diagnosis using knowledge guided retrieval augmentation for ChatGPT. *Journal of Biomedical Informatics*, *157*, 104702. https://doi.org/10.1016/j.jbi.2024.104702

Zhang, J., & 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), 04015014. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000346

Zhang, R., & El-Gohary, N. (2019). A machine learning-based method for building code requirement hierarchy extraction. *Proceedings, Annual Conference-Canadian Society for Civil Engineering*, 1–10. https://legacy.csce.ca/elf/apps/CONFERENCEVIEWER/conferences/2019/pdfs/PaperPDFversion_147_0423081134.pdf

Zheng, J., & Fischer, M. (2023). Dynamic prompt-based virtual assistant framework for BIM information search. *Automation in Construction*, *155*, 105067. https://doi.org/10.1016/j.autcon.2023.105067