A Traceable Clause-Level Retrieval and Evidence-Based Question Answering System for Tunnel Specifications

Yafei Sun[1], Xuesong Shen[1,*], Sisi Zlatanova[2], Khalegh Barati[1], Milad Mousavi[1], and James Linke[3]

[1]School of Civil and Environmental Engineering, University of New South Wales, Sydney, NSW 2052, Australia
[2]School of Built Environment, University of New South Wales, Sydney, NSW 2052, Australia
[3]GeoAI, Sydney, NSW 2019, Australia
yafei.sun@unsw.edu.au, x.shen@unsw.edu.au, s.zlatanova@unsw.edu.au, khalegh.barati@unsw.edu.au, milad.mousavi@unsw.edu.au, j.linke@geoai.au

Abstract

Tunnel engineering specifications are critical normative documents for compliance verification and decision support. Existing text-based retrieval methods do not sufficiently exploit the hierarchical structure between clauses, making it difficult to reliably locate evidence and prioritize core clauses. This paper proposes a clause-level retrieval and evidence localization method for tunnel specifications that explicitly leverages document structure as retrieval evidence. A two-stage framework is adopted, where text-based candidate retrieval is followed by structure-aware reranking. A lightweight structural knowledge graph is constructed to encode clause hierarchy and document membership, enabling the derivation of interpretable document-level aggregation and clause-level structural cues. These features provide an explainable structural bias for ranking without training reranking models or requiring additional annotation. Experiments on actual tunnel specifications with 50 engineering queries show improved top-ranking accuracy and evidence coverage over lexical and semantic baselines, while keeping millisecond-level retrieval latency. The results indicate that modeling structural cues is an effective foundation for evidence-based question answering.

Keywords –

Tunnel specifications; Clause retrieval; Structural knowledge graph; Structure-aware reranking.

1 Introduction

Tunnel engineering is characterized by strong compliance requirements and high complexity, involving the coordination of multiple systems such as electrical and ventilation systems [1]. Misinterpretation or omission of any key clauses can lead to significant safety consequences. Thus, tunnel specifications and technical standards documents play a crucial role as "traceable compliance evidence" throughout the entire operations management process [2]. In current practice, engineers typically rely on manual browsing or keyword search within document management systems to locate relevant clauses, a process that becomes increasingly time-consuming as specification sets grow in size and complexity. The risk of overlooking a critical clause or retrieving a similar but incorrect provision is amplified when related requirements are distributed across multiple hierarchical levels within a single document. Given the volume and depth of such documents, quickly locating the specific clauses that directly support decision-making remains a persistent challenge. Traditional information retrieval based on chapters or documents is insufficient to meet the needs of refined compliance retrieval and specification question answering (QA) [3, 4]. Therefore, clause-level retrieval and evidence location have become foundational for tunnel intelligent applications.

Existing document retrieval methods for specific domains often rely on keyword matching or semantic similarity calculations, treating clauses as independent text units [5, 6]. When dealing with tunnel specifications, these methods lack the application of explicit hierarchical organization and document-level aggregation patterns. At the clause level, simply ranking based on text similarity easily leads to scattered evidence clauses and core clauses being placed at the end; while retrieval granularity at the document or chapter level is insufficient to directly support the output of locatable and citable clause-level evidence. Specification clauses carry implicit structural context—such as document membership, hierarchical depth, and clause adjacency [7]—that, if left unexploited, limits the interpretability and traceability of retrieval results [8]. However, how to transform the specification structure into a computable and interpretable ranking criterion without introducing costly annotation or complex semantic modeling remains the core problem facing tunnel specification retrieval.

To address this issue, this paper proposes a clause-

level retrieval method for tunnel specifications that explicitly models document hierarchy as structural evidence for ranking. The method adopts a two-stage pipeline: (i) text-based candidate recall over clause content, followed by (ii) structure-aware reranking within the candidate set. We construct a lightweight structural knowledge graph (KG) with clauses as nodes to encode document membership and hierarchical position, from which interpretable structural cues are derived to bias ranking toward coherent and citable evidence. This design avoids complex semantic KG and trained reranking models, while capturing the "structure-as-evidence" property of specification texts.

The main contributions of this paper are as follows:

- A clause-level retrieval and evidence localization framework for tunnel specifications that outputs traceable clauses.
- A lightweight structural KG with clauses as nodes to encode document membership and hierarchy as a unified source of provenance information.
- A structure-aware reranking scheme, which uses document-level aggregation and clause-level structural cues.

The remainder of this paper is organized as follows. The following sections review related work, describe the proposed method, present the experimental results, and conclude with key findings and future directions.

2 Related Works

This section reviews prior work on tunnel specification retrieval, retrieval and reranking methods for domain-specific documents, and summarizes the key research gaps motivating this study.

2.1 Tunnel Specification Retrieval

Existing research mainly focuses on the digital management of specification texts, keyword retrieval, and rule-based clause location [9, 10]. Some works have integrated with engineering management systems to support the querying, browsing, and citation of specification clauses. However, these methods often use documents or chapters as the basic retrieval unit, emphasizing information organization and retrieval efficiency, and making limited use of the explicit hierarchical organization and clause structure within the specification texts. For specification of QA and compliance verification scenarios addressing specific engineering problems, these methods often struggle to directly output locatable and citable clause-level evidence, thus limiting their applicability in complex tunnel engineering practices.

2.2 Retrieval and Reranking Methods

In the broader processing of technical documents, information retrieval research often employs a two-stage retrieval paradigm [11, 12]. This involves initial candidate recall through keyword matching or semantic similarity models, followed by finer-grained reranking on the candidate set to improve ranking performance. This paradigm has been widely validated in balancing retrieval efficiency and result quality. Regarding retrieval tasks involving long documents and normative texts, some research has begun to focus on the role of document structure information (such as chapters, levels, or metadata) in evidence localization. It has attempted to improve retrieval or QA performance through structure-aware mechanisms [13-16]. However, it should be noted that research explicitly leveraging document structure for clause-level retrieval in normative texts is still relatively limited. Existing studies mainly explore the use of structural information as auxiliary context or organizational metadata, such as for text segmentation, navigation, or result presentation, rather than treating document structure as a computable and interpretable ranking signal. In addition, the retrieval granularity in most existing works is typically defined at the document or chapter level, which poses challenges when directly addressing fine-grained, hierarchically organized clauses in specification documents.

2.3 Research Gaps

Despite these initial efforts, research on structure-aware clause-level retrieval for normative engineering documents remains limited, leaving several open challenges. The review of existing research reveals several shortcomings in retrieval methods for tunnel engineering specifications. First, retrieval granularity is generally limited to the document or chapter level, failing to directly support the location and citation of clause-level evidence required in engineering practice. Second, while structure-aware retrieval is gaining attention, most methods rely on learned reranking models or complex semantic modeling, failing to explicitly incorporate structural cues into the ranking decision process in a low-cost, highly interpretable manner. Third, in applications such as retrieval-augmented generation, structural information is primarily used for text segmentation or organization, lacking research on systematically modeling document-level aggregation and clause-level hierarchical cues on candidate clause sets. Unlike approaches that rely on learned reranking models or complex semantic knowledge graph construction, this study treats structural cues as directly computable ranking features derived from document metadata, without model training or manual ontology design. This positions the approach as an interpretable and low-cost

alternative suited to engineering applications where deployment simplicity and model transparency are practical priorities.

3 Research Methodology

This section introduces the proposed clause-level retrieval method and system framework, including the overall process, clause-level structural representation, and structure-aware reranking strategy.

As shown in Figure 1, the system takes natural language queries as input, performs candidate recall and structure-aware reranking on a clause-level corpus, and outputs the ranked relevant clauses and their evidence fragments. The overall process adopts a two-stage design.

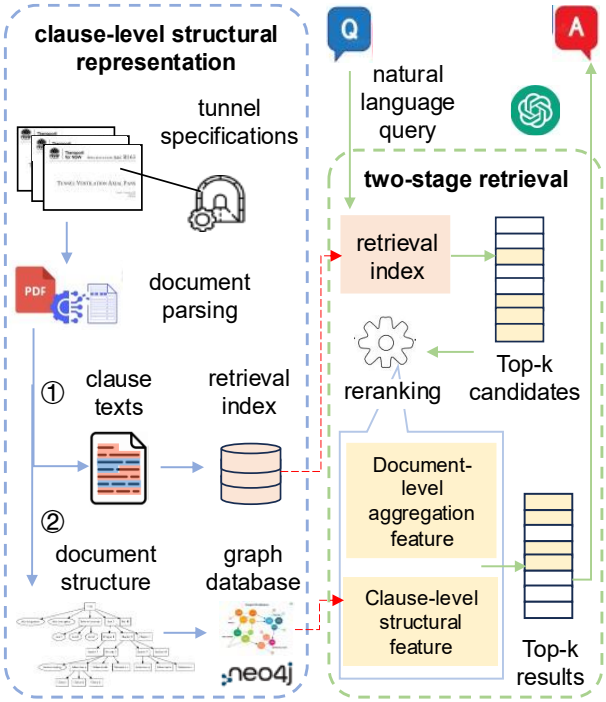

Figure 1. Framework of the proposed clause-level retrieval and evidence-based question answering system.

3.1 Clause-Level Structural Representation

The first step is parsing the specification document into a corpus with clauses as the basic unit. Each clause corresponds to an explicit entry in the specification. It is associated with structural meta-information such as its text content, clause number, the specification document it belongs to, and its hierarchical depth. The clause number and hierarchical depth are automatically inferred from the original chapter structure, used to characterize the relative position of the clause in the overall specification. Based on the above information, a lightweight structural knowledge representation is built, in which clauses are treated as nodes, and their hierarchical and membership relationships are explicitly encoded as node attributes. This structural representation can be regarded as a structural KG of the specification, serving as a unified structural basis in the subsequent ranking stage, without relying on complex semantic relation extraction or manual ontology modeling. Engineering specifications frequently contain explicit internal cross-references between clauses (e.g., "refer to Clause 6.3"). The current implementation does not parse such references at the indexing stage. However, the document-level aggregation mechanism in the reranking stage partially addresses this limitation: when multiple related clauses from the same specification appear in the candidate set, their shared document membership reinforces their ranking, indirectly capturing within-document co-referencing patterns. Explicit cross-reference parsing and graph-based linking represent a direction for future structural enrichment.

3.2 Candidate Retrieval

During the candidate retrieval phase, the system performs text-based retrieval over the clause corpus and returns the Top-k candidate clauses, narrowing the search space while maintaining high recall for subsequent structure-aware reranking. Each candidate corresponds to a complete clause text at the smallest granularity, ensuring a uniquely locatable and citable unit. All subsequent reranking operations are conducted on this unified Top-k candidate set.

3.3 Structure-Aware Reranking

Specification documents exhibit a highly explicit hierarchical structure, with relevant evidence typically distributed in “clause groups” within the same document or adjacent levels, rather than appearing as isolated clauses. When ranking solely based on text similarity, models often struggle to leverage this structured evidence distribution, leading to core clauses with stronger responsiveness failing to consistently appear at the top.

To address this, the reranking strategy incorporates document-level and clause-level structural cues from the KG into the initial text relevance ranking, applying a structured evidence bias that guides results toward more representative clauses without altering the candidate set.

The reranking process utilizes two complementary structural signals: first, document-level aggregation signals, where multiple candidate clauses originate from the same specification, indicating stronger overall relevance within the current query context. Therefore, candidate clauses within the same document are weighed to reinforce the “evidence clustering within the document” pattern. Second, clause-level structural signals,

leveraging attributes such as the hierarchical depth of clause nodes in the structural KG to characterize the functional differences of clauses in normative expression. Thus, they can be seen as structural constraints alongside textual relevance in the ranking process. Through the incorporation of document-level and clause-level structural cues, the reranking mechanism adjusts the relative ordering of candidate clauses in a structure-aware manner without changing the underlying retrieval results.

4 System Evaluations

To evaluate the proposed method, this paper conducts experimental analysis from aspects such as components ablation and external comparison and discusses the retrieval effectiveness and answer quality.

4.1 Experimental Setup

The corpus used is the design compliance specifications from New South Wales, Australia, covering multiple subsystems in tunnels, including electrical and lighting systems. A total of six specifications were selected as the corpus source for the system's retrieval and QA. The documents involved are shown in Table 1.

Table 1. Summary of specifications used in evaluation.

Code	Name	Scope	Clause number	Pages
D&C R153	Tunnel and underpass main switchboard, distribution boards and control panels	electrical boards	33	13
QA R154	Tunnel and underpass electrical services works	electrical services	46	13
QA R158	Road tunnel and underpass lighting	lighting equipment	45	17
D&C R163	Tunnel ventilation axial fans	ventilation equipment	61	27
D&C TS931	Tunnel electrical boards	electrical boards	35	16
D&C R164	Tunnel jet fans	ventilation equipment	65	27

The evaluation experiments were run in a WSL2 (Ubuntu) environment on a Windows host machine. During the experiments, the WSL environment had 12 logical processors (Intel® Core™ i7-8700 @ 3.20 GHz) and approximately 7.7 GB of memory (plus 2.0 GB swap). No graphics processing unit acceleration or model training was used during the experiments; retrieval and reranking were performed on the local central processing unit, and QA generation was completed online through OpenAI large language model interface. This configuration demonstrates the lightweight nature and reproducibility of the proposed method in engineering applications.

The system is evaluated from three aspects: retrieval effectiveness, system efficiency, and QA quality. Retrieval is assessed using standard information retrieval metrics, including Hit@k and Mean Reciprocal Rank (MRR). Hit@k indicates whether at least one relevant evidence clause appears in the Top-k results; MRR measures the reciprocal rank of the first relevant clause. The system efficiency is reported as average query latency. QA quality is evaluated against a small set of reference answers. Over generation, evidence-attribution constraints are enforced to ensure traceability and auditability. Unless otherwise stated, k is set to 5 in all evaluation experiments; the term Top-k is used to describe the general retrieval framework, while Top-5 refers to the specific cutoff adopted for quantitative evaluation.

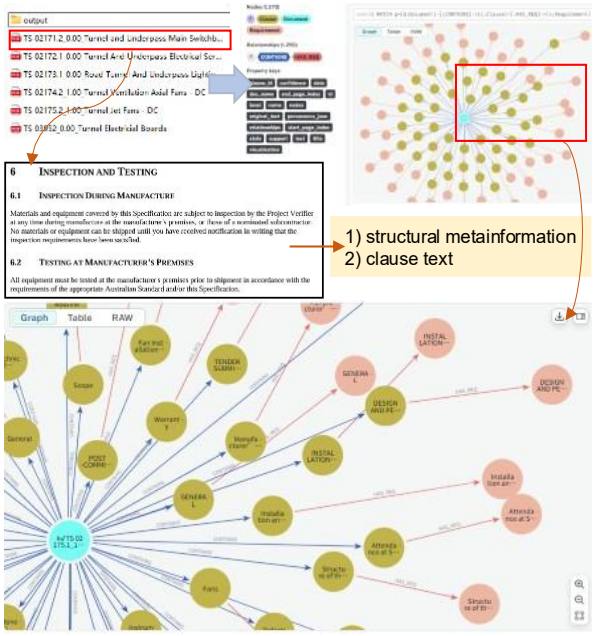

Figure 2. Clause-level KG scaffold constructed from specification documents.

4.2 Results

This subsection evaluates the proposed approach from three aspects: end-to-end functional performance, key component ablation, and comparative analysis.

4.2.1 Functionality Demonstration

This part uses representative cases to demonstrate key system outputs. The proposed system automatically parses specification PDFs into a clause-level index and constructs a KG scaffold encoding clauses, requirement entities, and their hierarchical relationships, as shown in Figure 2.

For the retrieval phase, taking specification QA R154 as an example, the query “What are the requirements for cable routing in tunnel electrical systems?” is selected. The system returned the Top-5 evidence clauses, as shown in Figure 3. Among them, Clause 6.2 “Cable Routing” was consistently ranked first, and its content directly specifies aspects such as cable route selection and support; the remaining returned clauses provide relevant information from aspects such as conduit specifications.

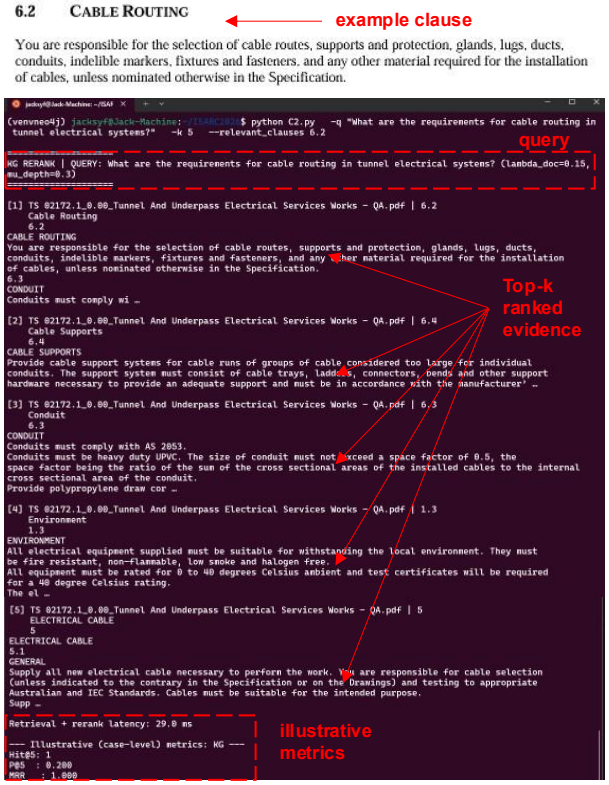

Figure 3. Example clause retrieval results.

Based on this representative case, exemplary (case-level) retrieval metrics are further reported: the system successfully covered the core relevant clauses in the Top-5 results (Hit@5 = 1), and the first relevant clause was ranked first (MRR = 1.00). In addition, the response latency for one retrieval and reranking was approximately 29.0ms, indicating that the system can efficiently return a set of evidence highly relevant to the specification requirements in interactive use scenarios.

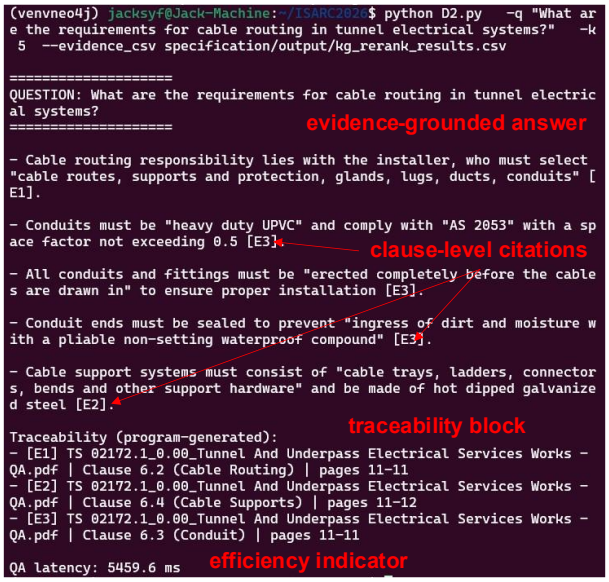

Figure 4. Evidence-based QA output.

During the QA phase, the system generates an answer from the Top-k evidence clauses, as shown in Figure 4. Each conclusion must cite its source clause, and a traceability list maps answers to specific documents, clause numbers, and page numbers, forming a verifiable "answer-evidence" closed loop. The end-to-end QA latency for this example was approximately 5.46 seconds.

4.2.2 Ablation Experiment

This subsection evaluates the impact of two types of rearrangement features derived from the KG structure design, document aggregation features and clause depth features, on the ranking performance of evidence clause retrieval through component ablation experiments.

The experiment was built upon 50 engineering queries designed around the typical information requirements of tunnel specifications. These queries were divided by topic and sampled evenly (e.g., electrical and mechanical, fire protection and safety) to avoid bias caused by queries focusing on a single chapter or term. Each query was expressed as a natural language question, consistent with the clause-level evidence retrieval task. In terms of the experimental procedure, the system first used BM25 [17], a widely used keyword-based ranking function, to generate a candidate clause list. Then, under the same candidate set and parameter settings, the BM25 baseline, a complete model incorporating only a single KG structural feature, and a complete model incorporating both types of KG structural cues were compared. For clarity, we denote the document-level aggregation feature as Doc, and the clause-level hierarchical depth feature as Depth in the ablation variants. Evaluation used Hit@1 and Hit@5, where @1 checks whether the top-ranked clause is relevant and @5 checks whether at least one relevant clause appears

among the top five results; all automatic judgments were manually reviewed and confirmed.

Table 2 shows a comparison of retrieval performance across different ablation variants. The BM25 baseline achieves a Hit@5 score of 0.84, indicating that keyword-based candidate recall is highly usable given the highly structured nature of the specification clause titles. However, its Hit@1 score is only 0.56, suggesting that clauses that directly answer the query are not always prioritized. Introducing document aggregation features derived from the KG structure design improves Hit@1 to 0.62, demonstrating that document-level aggregation using document-clause attribution relationships helps prioritize valid evidence within the same specification document.

Table 2. Ablation experiment results

Variant	Hit@1	Hit@5
BM25	0.56	**0.84**
Depth	0.54	0.82
Doc	0.62	0.80
Full	**0.62**	0.80

In contrast, introducing only clause depth features has limited effect, and the complete model does not improve over the Doc-only variant, indicating that document-level aggregation is the primary source of ranking gain in this task setting. Hit@5 shows little variation across variants, suggesting that the main benefit of structural cues lies in improving top-ranking quality rather than expanding candidate coverage.

4.2.3 Comparative Experiment

This subsection compares the proposed method against lexical, semantic vector, and pretrained embedding baselines under a unified setting to examine the effectiveness of structured evidence bias. A more challenging and application-oriented query set consisting of 50 natural-language specification questions is adopted. These queries were formulated based on domain consultation to reflect typical information needs in tunnel specification compliance workflows, covering electrical, lighting, ventilation, and control panel subsystems. While the queries were authored by the research team rather than extracted from actual engineer search logs, they were designed to span diverse subsystem topics and avoid single-chapter concentration. Integrating real-world search logs from engineering practice is recognized as a direction for future validation.

Compared with the ablation queries, these questions exhibit weaker lexical overlap with clause titles/headings, longer semantic expressions, and less explicit structural cues, making purely keyword-based retrieval substantially harder.

In the comparative experiments, the lexical retrieval method used the standard BM25 model as the baseline; the semantic retrieval method was based on latent semantic analysis, representing clause texts as low-dimensional semantic vectors and ranking them according to cosine similarity [18, 19]; our proposed method, based on the initial BM25 ranking results, introduced document-level aggregation features and clause-level features to perform structure-aware reranking of candidate clauses. All methods returned results under a unified Top-k setting, evaluated using the same weakly supervised judgment rule as the ablation experiments, with all relevance labels manually verified.

As shown in Table 3, our proposed method outperforms the comparative methods on both evaluation metrics. Compared to the Lexical retrieval baseline, the structure-aware reranking strategy achieves a stable improvement in the Hit@1 metric, indicating that introducing document-level and clause-level structural cues helps to prioritize clauses with greater judgment value. On the Hit@5 metric, our proposed method achieves near-saturation retrieval coverage, demonstrating that structured evidence bias can effectively reduce the omission of key clauses.

Table 3. Comparative experiment results

Method	Hit@1	Hit@5
Lexical retrieval	0.16	0.82
Latent semantic vector retrieval	0.12	0.60
Pretrained embedding (text-embedding-3-small)	0.04	0.26
Pretrained embedding (MiniLM-L6-v2)	0.06	0.30
Ours (structure-aware reranking)	**0.18**	**1.00**

To examine whether pretrained text embedding models offer advantages in this setting, two additional baselines were evaluated: OpenAI text-embedding-3-small and all-MiniLM-L6-v2 [20]. Clause texts were encoded into dense vectors and retrieval was performed via cosine similarity ranking. As shown in Table 3, both pretrained models achieved the lowest performance among all methods, falling below even the unsupervised LSA baseline. This result indicates that general-purpose pretrained embeddings, without domain-specific adaptation, do not transfer effectively to normative specification retrieval in the current zero-shot setting, where clause-level functional distinctions are encoded in domain-specific terminology rather than general semantic proximity. Together with the LSA result, these findings confirm that purely semantic approaches—whether corpus-specific or pretrained—are insufficient to capture the functional differences between specification clauses in this retrieval task. Overall, the

results support the effectiveness of structural information in normative clause retrieval tasks.In addition, the Hit@1 performance of the BM25 baseline drops significantly compared to the ablation setting, indicating that lexical matching alone is insufficient for evidence-oriented clause retrieval when queries deviate from structured specification terminology.

4.3 Discussion

The experimental results confirm that the structured organization of specification documents provides natural contextual constraints for retrieval, while purely semantic approaches are insufficient to characterize clause-level functional differences. Incorporating structural information into the ranking process improves both coverage and ranking reliability.

From a practical standpoint, the improvement in Hit@1 from 0.56 to 0.62 in the ablation setting means that for an additional 6% of queries, the most relevant clause is ranked first, reducing the need for engineers to manually inspect lower-ranked results. In the comparative evaluation, the proposed method achieved Hit@5 = 1.00, meaning that for all 50 queries, at least one relevant clause appeared within the top five results. In compliance verification workflows, where overlooking a key regulatory clause can carry safety and legal consequences, this near-complete evidence coverage directly supports the practical goal of reducing clause omission risk in engineering decision-making.

To assess QA generation quality beyond case-level demonstration, ROUGE scores [21] were computed for 44 of the 50 queries for which both retrieval results and matching ground-truth clause labels were available. The scores were computed by comparing system-generated answers against the full text of their ground-truth relevant clauses. The mean ROUGE-L precision reached 0.468, indicating that approximately 47% of generated answer content can be traced back to the source clause text. The corresponding ROUGE-L recall (0.053) and F1 (0.090) were lower, which reflects the expected asymmetry between concise bullet-point answers and full-length reference clauses—the system is designed to extract key provisions rather than reproduce entire clause content. These results confirm that the evidence-grounding constraints imposed during answer generation are effective in maintaining content fidelity to the source specifications.

Overall, the experiments provide consistent evidence supporting the design rationale and practical applicability of the proposed system in specification clause retrieval scenarios. Moreover, while this study focuses on tunnel specifications, the approach can be extended to other road-infrastructure normative documents with similar hierarchical structures to improve evidence-oriented operations management [22].

5 Conclusions

This paper developed a provenance-aware, clause-level retrieval and evidence localization system for tunnel engineering specifications. The proposed approach adopts a two-stage pipeline and a lightweight structural KG. The main contribution of this study lies in explicitly treating document structure as retrieval evidence rather than auxiliary context. Two structural signals, i.e. document-level aggregation and clause-level hierarchical cues, are incorporated into the ranking process without relying on trained reranking models or additional annotation, enabling a lightweight yet explainable retrieval mechanism suitable for engineering applications.

Experimental results demonstrate that the proposed method improves top-ranking reliability, with document-level aggregation increasing Hit@1 from 0.56 to 0.62 and the full approach achieving near-saturated Hit@5 coverage in comparative evaluations. Overall, the results confirm that explicitly modeling structural cues enhances ranking consistency while maintaining low computational overhead.

Future work will explore richer structural signals, extend the evaluation to broader categories of normative documents, and conduct more rigorous assessments of evidence-based QA quality under real-world engineering scenarios.

Acknowledgment

This research is supported by funding from the Australian Research Council (ARC) Research Hub for Resilient and Intelligent Infrastructure Systems (IH210100048), and the ARC Research Hub for Human-Robot Teaming for Sustainable and Resilient Construction (IH240100016).

References

[1] Bjelland, H., Gehandler, J., Meacham, B., Carvel, R., Torero, J. L., Ingason, H., & Njå, O. (2024). Tunnel fire safety management and systems thinking: Adapting engineering practice through regulations and education. Fire Safety Journal, 146, 104140.

[2] Chen, L., Shi, P., Tang, Q., Liu, W., & Wu, Q. (2020). Development and application of a specification-compliant highway tunnel facility management system based on BIM. Tunnelling and Underground Space Technology, 97, 103262.

[3] Lee, S. H., Choi, S. W., & Lee, E. B. (2023). A question-answering model based on knowledge graphs for the general provisions of equipment purchase orders for steel plants

maintenance. *Electronics*, *12*(11), 2504.

[4] Du, H., Feng, Y., Li, C., Li, Y., Lan, Y., & Zhao, D. (2023, July). Structure-discourse hierarchical graph for conditional question answering on long documents. In *Findings of the Association for Computational Linguistics: ACL 2023* (pp. 6282-6293).

[5] Yousefi, S., & Dami, S. (2024, February). Medical Documents Search Engine in the Comprehensive Hospital System Using Ontology-Based Semantic Similarity Measurement. In *2024 20th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP)* (pp. 1-6). IEEE.

[6] Yang, Y., Tang, W., Lu, Y., Wang, J., & Jiang, Y. (2024, October). Similarity Measurement Model for Standard Clauses Based on RAG. In International Conference on Computer Engineering and Networks (pp. 319-330). Singapore: Springer Nature Singapore.

[7] Chen, H., Yang, Y., Li, Y., Zhang, M., & Zhang, M. (2025). DISRetrieval: Harnessing Discourse Structure for Long Document Retrieval. *arXiv preprint arXiv:2506.06313*.

[8] Ma, Y., Wu, Y., Ai, Q., Liu, Y., Shao, Y., Zhang, M., & Ma, S. (2023). Incorporating structural information into legal case retrieval. *ACM Transactions on Information Systems*, *42*(2), 1-28.

[9] Ling, J., Li, X., Li, H., An, Y., Rui, Y., Shen, Y., & Zhu, H. (2024). Hybrid NLP-based extraction method to develop a knowledge graph for rock tunnel support design. *Advanced Engineering Informatics*, *62*, 102725.

[10] Yu, G., Wang, Y., Mao, Z., Hu, M., Sugumaran, V., & Wang, Y. K. (2021). A digital twin-based decision analysis framework for operation and maintenance of tunnels. *Tunnelling and underground space technology*, *116*, 104125.

[11] Dang, V., Bendersky, M., & Croft, W. B. (2013, March). Two-stage learning to rank for information retrieval. In *European Conference on Information Retrieval* (pp. 423-434). Berlin, Heidelberg: Springer Berlin Heidelberg.

[12] Ren, R., Ma, J., & Zheng, Z. (2025). Large language model for interpreting research policy using adaptive two-stage retrieval augmented fine-tuning method. *Expert Systems with Applications*, *278*, 127330.

[13] Jia, R., Zhang, B., Méndez, S. J. R., & Omran, P. G. (2025, May). StructRAG: Structure-Aware RAG Framework with Scholarly Knowledge Graph for Diverse Question Answering. In *Companion Proceedings of the ACM on Web Conference 2025* (pp. 2567-2573).

[14] Wu, Chengke, et al. "Retrieval augmented generation-driven information retrieval and question answering in construction management." *Advanced Engineering Informatics* 65 (2025): 103158.

[15] Xu, L., Feng, C., Zhang, K., Zhengyong, L., Xu, W., & Meng, F. (2025, October). Equipping Retrieval-Augmented Large Language Models with Document Structure Awareness. In *Findings of the Association for Computational Linguistics: EMNLP 2025* (pp. 24608-24631).

[16] Wang, S., Zhou, Y., & Fang, Y. (2025). BookRAG: A Hierarchical Structure-aware Index-based Approach for Retrieval-Augmented Generation on Complex Documents. *arXiv preprint arXiv:2512.03413*.

[17] Akkuş, Ş. G., & Emekci, H. (2025, August). Ranking Matters: A Comparative Study of BM25, BERT-E5, and Hybrid (BM25+ BERT-E5) Retrival Systems on the SQuAD 2.0 Dataset. In *2025 International Conference on Artificial Intelligence, Computer, Data Sciences and Applications (ACDSA)* (pp. 1-6). IEEE.

[18] Li, B., Wen, H., Wang, S., Hu, T., Liang, X., & Luo, X. (2025). An Optimized Semantic Matching Method and RAG Testing Framework for Regulatory Texts. *Electronics*, *14*(14), 2856.

[19] Collini, E., Kurniadi, F. I., Nesi, P., & Pantaleo, G. (2025). Context-Aware Retrieval Augmented Generation Using Similarity Validation to Handle Context Inconsistencies in Large Language Models. *IEEE Access*.

[20] Reimers, Nils, and Iryna Gurevych. "Sentence-bert: Sentence embeddings using siamese bert-networks." *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. 2019.

[21] Lin, Chin-Yew. "Rouge: A package for automatic evaluation of summaries." *Text summarization branches out*. 2004.

[22] Sun, Y., Shen, X., Zlatanova, S., Barati, K., & Linke, J. (2026). Text-based automatic knowledge graph construction for road infrastructure operations management. *Automation in Construction*, *182*, 106733.