eISSN 2783-6851

VILNIUS TECH
Vilnius Gediminas Technical University

NEW TRENDS in COMPUTER SCIENCES

2026
Volume 4
Issue 1
Pages 1–19
https://doi.org/10.3846/ntcs.2026.27121

EVALUATING KNOWLEDGE GRAPH ENHANCED RETRIEVAL AUGMENTED GENERATION FOR AUTOMATED FUNCTIONAL REQUIREMENTS EXTRACTION

Dilki Sandunika RATHNAYAKE, Asta SLOTKIENĖ ✉

Department of Information Systems, Vilnius Gediminas Technical University, Vilnius, Lithuania

Article History:
- received 19 May 2026
- accepted 3 June 2026

Abstract. The automation of requirements engineering using large language models offers significant potential for efficiency but struggles with hallucinations and a lack of domain-specific precision in highly regulated fields such as healthcare. While retrieval augmented generation (RAG) addresses some of these issues, standard vector-based retrieval often fails to capture complex semantic relationships required for strict compliance. This research assesses the effectiveness of automated functional requirement extraction by comparing two distinct retrieval architectures: a baseline vector-only RAG versus a knowledge graph-enhanced RAG with three distinct prompt strategies. We implemented an end-to-end automated extraction pipeline applied to a corpus of heterogeneous healthcare documents. The study constructed two separate knowledge bases to perform a comparative analysis. Under the experimental conditions (seven healthcare documents, three security-focused testing areas, and Claude-3-Haiku at temperature zero), knowledge base curation was found to be a stronger determinant of extraction quality than retrieval architecture, though the relative impact may vary with corpus scale, domain, and base LLM. With a curated knowledge base, GraphRAG eliminated the systematic relevance collapse observed under standard preprocessing, achieved significant structural traceability validity in independent Neo4j cross-checking, and produced prompt-stable performance across three prompt strategies. Without rigorous preprocessing, graph augmentation amplified rather than mitigated retrieval noise.

Keywords: requirements extraction, Large Language Models, prompt strategies, retrieval augmented generation, GraphRAG.

✉Corresponding author. E-mail: ***asta.slotkiene@vilniustech.lt***

1. Introduction

Functional Requirements (FRs) describe the basic tasks and functionalities of a software system and play a critical role in driving its systematic design, development, and validation. These requirements must be well-defined and unambiguous for software projects to succeed. However, extracting FRs from various domain sources, including Software Requirements Specifications (SRS), user stories, backlogs, and regulatory documents, remains a major challenge.

Traditional manual extraction methods are labour-intensive, error-prone, and struggle to scale effectively for complex projects (Alhoshan et al., 2023). While Natural Language Processing (NLP) and Artificial Intelligence (AI) have advanced significantly, fully automated solutions for effectively and consistently extracting FRs in accordance with industry standards remain elusive. A primary limitation of current Large Language Models (LLMs) approaches is hallucination, where models produce plausible but factually incorrect outputs, and a lack of domain adaptation in specialised fields like healthcare.

RAG has emerged as a solution for grounding LLMs' outputs in verified documentation. However, standard RAG implementations relying solely on vector similarity often miss the intricate semantic relationships necessary for compliance-relevant requirements practices (e.g., Implementing the Health Insurance Portability and Accountability Act (HIPAA) (Marron, 2024); IEEE 830 is historically influential but superseded by ISO/IEC/IEEE 29148) (IEEE Computer Society, 1998; International Organization for Standardization [ISO], 2011; Ji et al., 2024).

In this research, the main contributions are as follows:

1. Our research shows that knowledge graph augmentation eliminates the systematic relevance degradation observed under standard vector-based retrieval, while improving requirement faithfulness and regulatory compliance when applied over a curated knowledge base (KB).
2. Empirical results indicates that, within the experimental setup, KB refinement has a greater impact on RAG-based FR extraction performance than retrieval architecture alone. In the seven-document corpus, aggressive entity pruning reduces extraction volume but improves requirement faithfulness, compliance, and structural traceability when combined with GraphRAG. The generalizability of this pattern to larger or non-healthcare corpora remains to be determined.

The remainder of this paper is structured as follows. Section 2 presents related works on the subject. Section 3 shows the research methodology. Section 4 provides an evaluation of the framework. The experiment results are presented in Section 5. Section 6 reports the ablation study. Conclusions follow.

2. Related work

The evolution of automated Requirements Engineering (RE) has shifted from classical rule-based systems to sophisticated generative and retrieval-based architectures. This section reviews the latest advancements in LLM applications for requirements and the emerging role of RAG.

2.1. LLMs in requirements specification and generation

Several recent studies demonstrate that LLMs can match, and in some cases exceed, the performance of junior software engineers in drafting requirements (Alhoshan et al., 2025; Arvidsson & Axell, 2023; Krishna et al., 2024; Lewis et al., 2020; Wei, 2024) employed GPT-4 to automatically draft and validate SRS, finding that while LLMs excel at comprehension and accuracy, they still suffer from hallucinations without proper context refinement. Vogelsang and Fischbach (2024) established guidelines for using LLMs in RE, emphasising the importance of prompt engineering and contextual grounding to achieve professional-quality outputs. Similarly, research by Alhoshan et al. (2023) investigated the effectiveness of generative models, such as LLaMA, for requirements classification, highlighting a growing trend toward their use for complex semantic categorisation. Researchers Arvidsson and Axell (2023) and Wei (2024) note that, despite advances in prompt engineering, LLMs operating without external knowledge sources tend to produce hallucinated content, particularly when queries reference

specialised terminology or domain-specific standards (Ji et al., 2024). This limitation motivates integrating retrieval mechanisms to ground LLM outputs in verified documentation.

2.2. Prompt engineering and contextual accuracy

To mitigate generation errors, researchers have focused on Prompt Engineering. Researchers Arvidsson and Axell (2023) established guidelines for Prompt Engineering (PE) in RE, demonstrating that few-shot and Chain-of-Thought (CoT) prompting significantly improve the clarity and completeness of extracted FRs requirements. Further, in Aishwarya's research (Aishwarya, 2023), conversational LLMs were utilised to extract structured data from unstructured text, demonstrating that well-designed prompts can effectively bridge the gap between narrative stakeholder inputs and structured technical requirements. However, inconsistency in model output remains a primary limitation of prompt-only strategies.

2.3. Retrieval augmented generation and graph-based retrieval

The emergence of RAG systems addresses the hallucination problem by grounding LLM outputs in verified documentation. Researchers (Ji et al., 2024) demonstrated that RAG architectures outperform standard generative models by leveraging external knowledge sources. In industrial settings (Arora et al., 2024), RAG-enhanced LLMs were successfully used to generate test scenarios from natural-language requirements.

However, standard vector-based RAGs face critical limitations, such as limited contextual windows and retrieval inaccuracies in complex domains. Barnett et al. (2024) identified seven key failure points in RAG implementations, emphasising that simple similarity searches often miss deep semantic connections. To resolve this, researchers are beginning to explore hybrid frameworks. In the research (Feng et al., 2024), LLMs were used to extract semantic relationships for normative requirements, showing that understanding the "graph" of relationships between system capabilities is essential for operationalisation. This study builds on this foundation by quantifying how the GraphRAG architecture affects faithfulness, relevance, and specificity compared to traditional vector-only RAG.

Recent work has begun to systematically compare the trade-offs between vector-based and graph-augmented retrieval. Microsoft's GraphRAG framework (Edge et al., 2025) showed that community-based graph summarisation outperforms vector retrieval on global sense-making queries, but behaves differently on local factoid queries. Building on this, hybrid retrieval methods that combine vector similarity with graph traversal have emerged as a complementary direction: HybridRAG (Sarmah et al., 2024) splices Vector-RAG and GraphRAG outputs and reports complementary precision-recall behaviour on financial earnings-call corpora; LightRAG (Guo et al., 2025) introduces a dual-level retrieval scheme over graph + vector indices that prioritises efficiency and incremental update; and PathRAG (Chen et al., 2026) explicitly addresses the redundancy of dense graph retrieval through flow-based path pruning, arguing that the limitation of current graph-RAG lies in retrieval redundancy rather than insufficiency. These results collectively suggest that the value of graph augmentation depends strongly on retrieval-noise control, which aligns directly with the Standard-versus-Refined KB comparison reported in Sections 5.2 and 5.3 of this paper. To date, however, these architec-

tures have been evaluated mainly on open-domain question-answering and document summarisation benchmarks; their behaviour in requirements extraction, where atomicity, traceability to source, and conformance to engineering standards (ISO, 2011) drive evaluation, has not been systematically characterised. The present study addresses that gap.

2.4. Research gap and contribution

While recent work has demonstrated LLM potential for requirements engineering (Krishna et al., 2024; Vogelsang & Fischbach, 2024) and explored RAG architecture (Arora et al., 2024; Edge et al., 2025; Ji et al., 2024), systematic comparisons of vector-only versus knowledge graph-augmented RAG specifically for FRs extraction remain absent. Existing GraphRAG evaluations focus on summarisation and question answering (Edge et al., 2025), leaving open questions about effectiveness for requirements engineering, where atomicity, traceability, and compliance with standards and guidelines (IEEE Computer Society, 1998; Marrron, 2024) are paramount. This study addresses this gap by rigorously comparing Vector-Only RAG and GraphRAG for automated extraction of FRs in healthcare systems. We adapt the RAGAS (Es et al., 2024) framework to requirements engineering contexts, introducing domain-specific metrics for technical term coverage and compliance scoring. Our work provides empirical evidence quantifying the relative benefits of Graph-augmented retrieval for regulated domains where documentation fidelity and semantic completeness are critical.

3. Methodology

This section describes the design and implementation of a graph-enhanced GraphRAG framework for automated FR extraction.

The proposed system architecture is designed to transform unstructured healthcare documentation into standardised, atomic FRs. The pipeline consists of four critical phases: (1) Document Preprocessing, (2) Knowledge Base Construction, (3) Intelligent Requirements Extraction and Processing, and (4) Quality Validation and Compliance Verification.

3.1. Document preprocessing

The preprocessing pipeline converts each PDF into a structured representation that can be indexed for retrieval and traversed as a graph. PDF parsing uses layout-aware extraction, preserving the section hierarchy and document structure rather than collapsing them into a flat token stream. Named entities are identified using spaCy's transformer-based NER model, then passed through a canonicalisation step that maps surface variants (e.g., "PHI", "Protected Health Information") to a single normalised form. A second filtering pass restricts the entity set to healthcare-relevant terms, which keeps the downstream graph focused on the domain rather than on incidental named entities such as document authors or publication dates. Subject-predicate-object triplets are then extracted from sentences that pair a requirements-engineering process verb (e.g., shall, must, requires) with a healthcare or compliance noun phrase, producing the relational backbone of the graph. The same source text is split into overlapping semantic chunks for the vector index, using a sliding window to preserve

sentence-level context across chunk boundaries. Finally, entity co-occurrence edges are added by linking any two entities that appear within a 50-token window of each other, with edge weights proportional to co-occurrence frequency.

3.2. Knowledge base construction

Two knowledge bases were constructed to validate the impact of data quality on extraction performance:

The *Standard KB* combines a vector index with a relational graph built from the same source documents. The vector index uses 384-dimensional sentence embeddings produced by the all-MiniLM-L6-v2 transformer, stored in a FAISS IndexFlatL2 structure that performs exact Euclidean nearest-neighbour search. The relational graph is built as a NetworkX MultiDiGraph from the SPO triplets extracted during preprocessing, using a deterministic regulatory heuristic that links technical requirements to their applicable compliance citations (e.g., a *user authentication* node connecting to a *HIPAA § 164.312* node). The resulting graph is dense (density = 0.97), and this density is by design: every entity is preserved, including peripheral ones. The trade-off is that some of those peripheral entities are noise rather than signal, and they create paths that the 2-hop traversal will follow without resistance. The Refined KB addresses this directly.

The *Refined KB* applies three filters to the Standard KB structure, in sequence, to remove entities that contribute noise but not retrieval value.

The first filter is a whitelist of healthcare entities. Entities are checked against a curated list of healthcare and compliance terms; entities that do not match are dropped before the graph is built. The list is conservative, broad enough to include all clinically and regulatorily relevant terms and narrow enough to exclude generic named entities (people, places, dates) that NER picks up but have no role in requirements extraction.

The second filter is degree-based pruning. Any entity with fewer than two graph connections is removed, on the rationale that a node with only one edge cannot meaningfully participate in 2-hop traversal, the path through it is a dead-end loop back to its single neighbour. The sensitivity analysis (Table 1) shows that this single threshold accounts for almost all the structural differences between the Standard and Refined KBs.

The third filter is Louvain community detection, which partitions the remaining nodes into 7 communities, each clustering around a regulatory anchor (e.g., HIPAA, HL7 FHIR, FDA guidance, etc.). The communities are not used as retrieval boundaries; traversal can still cross between them, but they provide a natural structure for monitoring whether retrieved context is regulatorily coherent.

Both KBs use the same dual-index FAISS structure, with separate IndexFlatL2 indices for document chunks and for canonicalised FRs. Using two indices allows the retrieval step to query at the granularity that suits the query, rather than retrieving large document chunks when a more targeted requirement-level match is sufficient.

3.3. KB construction parameter sensitivity analysis

The Refined KB uses three structural pruning mechanisms, and their parameter settings can affect how well it retrieves FRs. We conducted a sensitivity analysis by varying each parameter in turn while keeping the others at their default values (min_degree = 2, clique_density = 0.9, max_cooccurrence = 15,000). For each configuration, we fully rebuilt the KB from the pre-processed document corpus and recorded key graph statistics. Results are presented in Table 1.

Table 1. KB construction parameter sensitivity analysis: graph topology statistics per configuration

Configuration	Parameter	Nodes	Edges	Density	Communities	Pruned Nodes	Active?
min_degree = 1	degree ≥ 1	766	3.388	0.010	7	0	No
min_degree = 2 (default)	degree ≥ 2	451	3.073	0.026	7	315	Yes
min_degree = 3	degree ≥ 3	440	3.051	0.027	7	326	Yes
clique_density = 0.7	density > 0.7	451	3.073	0.026	7	315	No[a]
clique_density = 0.8	density > 0.8	451	3.073	0.026	7	315	No[a]
clique_density = 0.9 (default)	density > 0.9	451	3.073	0.026	7	315	No[a]
cooccur_5k	cap = 5,000	451	3.073	0.026	7	315	No[b]
cooccur_15k (default)	cap = 15,000	451	3.073	0.026	7	315	No[b]
cooccur_30k	cap = 30,000	451	3.073	0.026	7	315	No[b]

Note: [a]Clique removal inactive: no entity subgraph of size ≥ 10 exceeds density 0.7 in this corpus. [b]Co-occurrence cap inactive: post-pruning graph contains fewer than 5,000 co-occurrence edges. Default values are indicated in parentheses in the Configuration column.

Three findings were observed. First, min_degree is the only parameter that materially alters graph topology. Lowering it to 1 retains a far larger node set and produces a sparser, noisier graph; raising it to 3 prunes only marginally more nodes than the default and leaves graph density essentially unchanged. The default value sits at a structural knee in the pruning curve: most noise nodes have exactly one connection and are removed at this threshold, while substantively connected entities (those tied to multiple documents and standards) are preserved.

Second, the clique detection mechanism is inactive at all tested density thresholds. Inspection of the communities shows that no entity group of ten or more nodes reaches even the loosest density threshold in this seven-document corpus, so the glossary-clique problem that the filter is designed to remove does not occur at this scale. The filter is retained nonetheless as insurance against denser, term-heavy corpora; full regulatory libraries, for example, would be expected to contain glossary cliques.

Third, the co-occurrence edge cap is similarly inactive. After degree-based pruning, the remaining graph contains fewer co-occurrence edges than the most restrictive cap tested, so all three cap values produce identical graphs. The practical implication is that the cap parameter is irrelevant at the current corpus scale and only becomes meaningful for substan-

tially larger document sets; the default value provides comfortable headroom for corpora of roughly 50–100 documents of similar density.

Together, these findings confirm that the Refined KB construction is governed by a single active mechanism, degree pruning at threshold 2, and that the reported performance characteristics are robust to variation in the other construction parameters at this corpus scale.

3.4. Intelligent requirements extraction and processing

Final synthesis was performed by Anthropic's Claude-3-Haiku-20240307 with temperature set to zero. Three considerations motivated this choice. First, the dated snapshot (20240307) was selected over an open-version alias to ensure that the LLM behaviour reported here can be reproduced exactly: this is important when retrieval-architecture differences are small and when prompt stability is itself one of the variables under study. Second, Claude-3-Haiku occupies a price-performance segment representative of mid-tier production-grade LLMs used in practical RE deployments; selecting a fast, mid-tier model rather than a frontier model lets the retrieval-architecture effects be observed without being absorbed by the generative capacity of a much larger LLM. Third, temperature zero with a pinned version produces deterministic outputs across repeated runs; combined with the shared query encoder (all-MiniLM-L6-v2) and the dual-index FAISS structure described below, this isolates the retrieval step as the only source of architectural variation between Baseline and GraphRAG. We acknowledge that absolute scores in Section 4 will not transfer directly to other LLMs, and we discuss this external validity boundary in Section 6. Both retrieval architectures share the same query encoder (all-MiniLM-L6-v2), so any difference in output is attributable to the retrieval step rather than the query representation. Baseline retrieval performs an exact nearest-neighbour search over the FAISS index with $k = 3$. GraphRAG uses the same vector search to identify three seed nodes, then expands each seed via a 2-hop traversal of the relational graph, capped at three additional nodes per seed, and uses bidirectional substring matching with a minimum match length of 3 characters. The retrieved contexts from either path are concatenated with their source-document metadata headers and passed to the model in a single request.

3.5. Quality validation and compliance verification

Each extracted FR passes through three lightweight validation checks before being entered into the evaluation pipeline. The first check is syntactic, requiring the requirement to begin with "The system shall" or "The user shall", the IEEE 830 sentence frame for testable functional requirements. The second check confirms that at least one healthcare-domain term from the extracted vocabulary appears in the requirement text, which catches generic outputs that an LLM occasionally produces when retrieval returns sparse context. The third check verifies that the requirement contains content words that also appear in the retrieved source chunks; this is a coarse pre-filter against hallucination, distinct from the formal Faithfulness metric defined in Section 5. Requirements that fail any of these checks are flagged but not discarded; they are added to the evaluation set so the metric framework can quantify their failure mode.

4. Evaluation framework

The evaluation framework adapts the RAGAS reference-free protocol (Es et al., 2024) to FRs extraction. RAGAS was originally designed for open-domain question answering, where ground-truth answers are typically not available. The same constraint holds in requirements engineering: hand-annotated reference requirement sets at the scale needed for systematic evaluation are rare in the public literature. The four metrics defined below, Faithfulness, Answer Relevance, Technical Term Coverage, and Compliance Score, operate without a reference set and map directly to four IEEE 29148 quality attributes (correctness, unambiguity, completeness, conformance), so that automated scores reflect professionally recognised quality dimensions rather than ad-hoc proxies.

The framework consists of four quantitative metrics that provide a comprehensive assessment across multiple quality dimensions. The faithfulness measures the share of content words in an extracted requirement that also appear in the retrieved source context. The metric targets the hallucination problem directly: a requirement containing content words that do not appear in any retrieved chunk is unlikely to be grounded in the source corpus. The computation uses lowercased token-level matching with a length filter that removes function words and short articles:

$$\text{Faithfulness}\left(FR, C\right) = \frac{\left|\left\{\omega \epsilon T_{FR} : \omega \epsilon C_{lower}\right\}\right|}{\left|T_{FR}\right|}, \tag{1}$$

where $T_{FR} = \left\{\omega \in \text{tokenize}\ (FR) : |\omega| > 3 \wedge \text{isalnum}(\omega)\right\}$ and C_{lower} is the lowercase source context string.

Evaluates both semantic alignment with the query and the target testing area using embedding space similarity to ensure requirements address the intended security functionality.

$$\text{Relevance}\left(FR, A\right) = \frac{\boldsymbol{E}_{FR} \cdot \boldsymbol{E}_{A}}{\left\|\boldsymbol{E}_{FR}\right\| \cdot \left\|\boldsymbol{E}_{A}\right\|}, \tag{2}$$

where $\boldsymbol{E}_{FR}$ and $\boldsymbol{E}_{A}$ are 384-dimensional embeddings from the model. These embeddings are L2-normalised, so the dot product is equivalent to cosine similarity.

Evaluates the use of authentic domain-specific terminology density, log-normalised to prevent inflation.

$$\text{TechCoverage}\left(FR, V\right) = \min\left(1.0, \frac{\left|T_{FR} \cap V\right|}{1.5 \cdot \ln\left(\left|T_{FR}\right| + 1\right)}\right), \tag{3}$$

where T_{FR} – all tokens in the requirement, and V – healthcare security vocabulary

Assesses for adherence to industry standards like IEEE 830 through rule-based structural validation. This checks for proper requirement syntax, penalises compound clauses and ambiguous language, and quantifiable acceptance criteria (according to IEEE 830):

$$\text{Compliance}\left(FR\right) = \max\left(0, \min\left(1.0, S + A + M - P\right)\right), \tag{4}$$

where: S – structural component, A – atomicity bonus, M – measurability bonus, and P – vagueness penalty.

$$P = \sum_{v \epsilon V_{vague}} 0.1 \cdot \left[v \epsilon FR \right], \tag{5}$$

To determine the final performance, the framework calculates an Overall Quality Score:

$$\text{Overall Quality} = \frac{1}{4}\begin{pmatrix} \text{Faithfulness} + \text{Relevance} + \\ \text{TechCoverage} + \text{Compliance} \end{pmatrix}. \tag{6}$$

The four metrics correspond to established IEEE 29148 requirements quality attributes. Faithfulness operationalises correctness (traceability to source documentation), Answer Relevance operationalises unambiguity (semantic alignment with the intended domain), Technical Coverage operationalises completeness (domain vocabulary density), and Compliance operationalises conformance (structural adherence to IEEE 830 "shall" syntax, atomicity, and verifiability). This mapping ensures that automated evaluation scores reflect professionally recognised quality dimensions rather than arbitrary computational proxies.

5. Experimental results

This section presents a factorial experiment comparing two KB configurations (Standard and Refined), three prompt engineering strategies (zero-shot, few-shot, and CoT), and two retrieval methods (Baseline RAG and Graph-Augmented RAG) across three healthcare security testing areas. This experiment analyses how relational data influences requirement quality. We analyse performance across four quality dimensions as detailed in Section 4.

5.1. Experimental setup

The corpus consists of seven healthcare documents: one IEEE 830-style Software Requirements Specification for a telemedicine consultation system and six regulatory guideline documents covering HIPAA, HITECH, FDA, and HL7 FHIR provisions. The corpus is deliberately scoped. Public datasets for requirements engineering that are large enough for fine-tuning are scarce. Motger and Franch (2024) report that 31 of 62 publicly available RE datasets contain fewer than 1,000 artefacts, and recent work has shown that small, domain-focused corpora are not a limitation for LLM-based requirements work but a productive constraint when token budgets favour precision over volume (Abualhaija et al., 2024; Pasquale et al., 2025). The corpus used here is large enough to expose architectural differences between Baseline and GraphRAG retrieval and small enough to allow careful preprocessing.

We treat this corpus size as a deliberate methodological constraint rather than an incidental limitation. A small, carefully prepared corpus exposes the architectural differences between Baseline and GraphRAG cleanly, because retrieval-noise effects are not masked by the heterogeneity that a large, loosely curated corpus would introduce; this is consistent with replication guidance for NLP-based requirements engineering, which emphasises controlled, well-characterised datasets over scale alone (Abualhaija et al., 2024). At the same time, we acknowledge the corresponding cost: the empirical patterns reported below, in particular the KB-curation effect, the prompt invariance of Refined-KB GraphRAG, and the structural

trace rates, should be read as hypotheses generated under tightly controlled conditions, to be confirmed on larger and cross-domain corpora rather than treated as established general properties. The external-validity boundaries of this design are discussed further in Section 6.

The experiment varies three factors, yielding twelve configurations:

- Knowledge base. Standard KB (unfiltered entity graph) and Refined KB (whitelist + degree pruning + Louvain partitioning).
- Retrieval architecture. Baseline RAG (vector search at k = 3) and GraphRAG (vector search at k = 3 followed by 2-hop graph traversal with bidirectional substring matching).
- Prompt strategy. Zero-Shot, Few-Shot, and Chain-of-Thought.

The base LLM is Claude-3-Haiku-20240307 at temperature zero. Extraction is performed across three healthcare security testing areas: patient identity verification and authentication, audit log integrity and tamper resistance, and PHI access controls and authorisation.

All retrieval parameters were selected through a one-factor-at-a-time ablation study reported in Section 6: baseline retrieval depth k = 3, GraphRAG seed retrieval k = 3, 2-hop traversal, node expansion cap of 3, and seed term minimum length of 3 characters. Setting both architectures to k = 3 keeps the initial retrieval volume constant, so the only architectural difference between Baseline and GraphRAG is the 2-hop graph expansion performed after the seed retrieval. This isolates the graph-traversal step as the variable under comparison.

Statistical analysis uses two non-parametric tests appropriate for the small per-cell sample sizes. Within each KB, paired Wilcoxon signed-rank tests compare Baseline against GraphRAG on per-requirement scores within each prompt strategy. Across KBs, unpaired Mann-Whitney U tests compare the Standard and Refined configurations. Effect sizes are reported as Cohen's d, and 95% bootstrap confidence intervals (n = 2,000 resamples) are computed for all mean scores. Per-requirement scores are reported as mean ± SD with 95% CI in Tables 3, 4, and 5.

5.2. Knowledge base quality impact

5.2.1. Extraction completeness

Table 2 presents the FRs counts across both KBs.

Table 2. The count of extracted functional requirements

KB Type	Prompt	Baseline	GraphRAG	Total
Standard KB	Zero-Shot	7	10	17
	Few-shot	8	12	20
	CoT	9	10	19
	Subtotal	24	32	56
Refined KB	Zero-Shot	7	7	14
	Few-shot	8	7	15
	CoT	7	7	14
	Subtotal	22	21	43

Table 2 shows that GraphRAG amplifies extraction volume on the Standard KB but not on the Refined KB, indicating that aggressive entity pruning effectively closes the speculative graph paths that the 2-hop traversal would otherwise follow.

5.2.2. Quality metric analysis

Three patterns are visible across Table 3. First, Faithfulness improves under GraphRAG in both KB conditions, with a larger gain in the Refined KB condition. The mechanism is the same in both: graph traversal returns a context that contains more content words from the generated requirement, and entity filtering in the Refined KB removes noise nodes that introduce lexical mismatches between the requirement and the retrieved source.

Second, the Relevance metric sharply separates the two KBs. On Standard KB, GraphRAG shows systematic relevance degradation that does not appear in Refined KB, where Baseline and GraphRAG remain near-parity. This is the most consequential observation in Table 3 within our experimental setup: in this corpus, retrieval relevance appears to be more strongly influenced by KB quality and seed-matching than by the post-retrieval filtering mechanisms typically discussed in the GraphRAG literature. Whether the same ordering holds for larger or non-healthcare corpora remains to be studied.

Third, Technical Coverage and Compliance vary little across configurations, with GraphRAG showing marginal gains in both. The Compliance result is consistent with the Refined KB's regulatory community structure, which provides anchors that help preserve IEEE 830 syntax. The Refined KB GraphRAG configuration produces the highest Overall score of any non-Zero-Shot configuration in the study, while Refined KB Baseline takes the top score under Zero-Shot, where a prompt-only signal is sufficient to match graph augmentation.

Table 3. Mean Quality scores by knowledge base (mean ± SD)

KB Type	Mode	Faithfulness	Relevance	Technical Coverage	Compliance	Overall±SD	n
Standard KB	Baseline	0.864	0.532	0.960	0.830	0.796±0.067	24
	GraphRAG	0.933	0.443	0.960	0.870	0.802±0.052	32
Refined KB	Baseline	0.934	0.532	0.940	0.870	0.820±0.041	22
	GraphRAG	0.972	0.528	0.950	0.890	0.833±0.030	21

Note: All scores were computed as the mean across the n FRs extracted under each configuration. SD reported only for the Overall composite score. Per-prompt breakdowns are provided in Tables 4 and 5.

In this corpus, GraphRAG improves Faithfulness in both KB conditions but diverges sharply on Relevance, degrading on the Standard KB and matching Baseline on the Refined KB. The Refined-KB-with-GraphRAG configuration produces the highest Overall score in the study (0.833±0.030, n = 21), driven primarily by gains in Faithfulness rather than Relevance or Technical Coverage.

5.3. Prompt engineering performance

5.3.1. Cross-Prompt Comparison (Standard KB)

Based on the results in Table 4, Standard KB shows two distinct prompt regimes. Under Zero-Shot and Few-Shot prompting, Baseline outperforms GraphRAG because the noise in the uncurated graph leads to context dilution that simpler prompts cannot recover from. Under CoT prompting, the relationship inverts: GraphRAG outperforms Baseline by a margin

large enough to reach statistical significance on Faithfulness (p = 0.016, d = +1.610). The mechanism is that CoT's explicit reasoning chain forces the model to use the regulatory anchors provided by GraphRAG's graph traversal, even when the surrounding context contains noise. The relevance degradation that GraphRAG produces on the Standard KB is consistent across all three prompts, confirming that it is an architectural property of GraphRAG over an uncurated KB, not an effect of any specific prompt formulation.

Table 4. Standard KB results by prompt strategy (mean ± SD)

Prompt	Metric	Baseline	GraphRAG	Δ
Zero-shot	Overall ±SD	0.833±0.034	0.804±0.057	−0.029
	Overall 95% CI	[0.813, 0.858]	[0.771, 0.837]	
Few-shot	Overall ±SD	0.797±0.049	0.795±0.045	−0.002
	Overall 95% CI	[0.764, 0.826]	[0.769, 0.818]	
CoT	Overall ±SD	0.766±0.088	0.807±0.057	+0.041
	Overall 95% CI	[0.712, 0.821]	[0.772, 0.839]	

On the Standard KB, the prompt strategy modulates the value of GraphRAG. Zero-Shot and Few-Shot favour Baseline because the uncurated graph context dilutes the prompt, whereas CoT recovers GraphRAG's advantage by forcing the model to use the regulatory anchors that graph traversal supplies. The CoT Faithfulness gain (Δ = +0.196, p = 0.016, Cohen's d = +1.610) is the only within-KB comparison in the study reaching p < 0.05.

5.3.2. Cross-Prompt Comparison (Refined KB)

The Refined KB eliminates the relevance collapse seen on the Standard KB and, in doing so, changes the prompt-architecture interaction. GraphRAG matches or outperforms Baseline across all three prompts, with the largest gain on Few-Shot, where the model appears to use graph-supplied regulatory anchors as additional exemplars alongside the prompt-supplied ones. Zero-Shot Baseline retains a marginal lead, suggesting that when the prompt itself is unstructured, KB quality alone is sufficient to drive performance, and graph augmentation provides no additional gain. The most striking property of the Refined KB GraphRAG configuration is its prompt invariance: Overall scores across the three prompt strategies fall within a 0.001 range (SD = 0.0008), an order of magnitude below the variance under both

Table 5. Refined KB results by prompt strategy (mean ± SD)

Prompt	Metric	Baseline	GraphRAG	Δ
Zero-shot	Overall ±SD	0.835±0.033	0.834±0.031	−0.001
	Overall 95% CI	[0.815, 0.859]	[0.815, 0.857]	–
Few-shot	Overall ±SD	0.800±0.050	*0.834±0.032*	*+0.034*
	Overall 95% CI	[0.766, 0.829]	[0.815, 0.857]	–
CoT	Overall ±SD	0.827±0.031	0.833±0.031	+0.006
	Overall 95% CI	[0.810, 0.849]	[0.814, 0.855]	–

Standard KB GraphRAG and Refined KB Baseline. This near-zero variance suggests that a curated KB removes prompt formulation as a meaningful design lever for graph-augmented retrieval; that the architecture and data quality together determine performance; and that the prompt strategy has little incremental effect.

Once the KB is refined, the prompt–architecture interaction collapses. GraphRAG matches or exceeds Baseline across all three prompts, and the three Refined-KB GraphRAG Overall scores fall within a 0.001 range (SD = 0.0008), suggesting that, in this configuration, prompt formulation is no longer a primary performance lever.

5.4. Prompt engineering with knowledge base interaction

Table 6 captures the prompt-architecture interaction in compact form. The pattern across both KBs is that the value of graph augmentation increases with prompt structure: GraphRAG provides no advantage under Zero-Shot, a modest advantage under Few-Shot, and a clear advantage under CoT. Under the Standard KB, this interaction is masked by noise except in the CoT case, where the explicit reasoning chain forces the model to attend to graph-supplied regulatory anchors. Under the Refined KB, where noise is removed, the same interaction surfaces are consistently observed across Few-Shot and CoT, suggesting that graph-based regulatory anchoring and structured prompting are complementary mechanisms that reinforce each other rather than substitute for each other.

Table 6. Best configuration per prompt

Prompt	Best Config	Overall Score	Key Strength
Zero-shot	*Refined KB + Baseline*	*0.835*	KB quality dominant; near-tie with GraphRAG (0.834)
Few-shot	Refined KB + GraphRAG	0.834	Largest faithfulness and compliance gains
CoT	*Refined KB + GraphRAG*	*0.833*	GraphRAG advantage on faithfulness

Across both KBs, the magnitude of GraphRAG's advantage scales with prompt structure: no benefit under Zero-Shot, a modest benefit under Few-Shot, and the clearest benefit under CoT. This is consistent with graph-supplied regulatory anchors acting as additional reasoning scaffolds that more structured prompts can exploit.

5.5. Statistical significance analysis

Table 7 presents Wilcoxon signed-rank test results for paired Baseline vs. GraphRAG comparisons within each KB configuration, alongside Cohen's d effect sizes and 95% bootstrap confidence intervals.

Only one within-KB comparison reached $p < 0.05$ (Standard KB CoT Faithfulness, $d = +1.610$). The remaining findings are supported by medium-to-large practical effect sizes rather than p-values, consistent with the limited statistical power of small per-cell sample sizes ($n = 6–11$). The systematic Relevance degradation under Standard KB GraphRAG ($d = −0.45$ to $−0.55$ across all three prompts) confirms that the degradation is an architectural property, not a prompt-specific effect.

Table 7. Wilcoxon signed-rank test: Baseline vs. GraphRAG within each knowledge base configuration (selected metrics)

KB	Prompt	Metric	Baseline Mean	GraphRAG Mean	Δ	p-value	Cohen's d
Standard	Zero-Shot	Faithfulness	0.962	0.916	−0.045	0.313	−0.525 (medium)
	CoT	Faithfulness	0.770	0.966	+0.196	0.016[a]	+1.610 (large)
	CoT	TechCoverage	1.000	0.961	−0.039	0.500	−0.704 (large)
	CoT	Relevance	0.529	0.464	−0.065	0.301	−0.527 (medium)
Refined	Few-Shot	Faithfulness	0.895	0.973	+0.077	0.500	+0.635 (medium)
	Few-Shot	Overall	0.798	0.834	+0.036	0.125	+0.823 (large)

Note: [a]Significant at $p < 0.05$. All other comparisons did not reach conventional significance thresholds, reflecting the limited statistical power of the small per-group sample sizes (n = 6–11). Cohen's d magnitude conventions: $|d| < 0.2$ = negligible; 0.2–0.5 = small; 0.5–0.8 = medium; > 0.8 = large.

Only one within-KB comparison reaches conventional significance, but the medium-to-large effect sizes, especially the consistent negative d for Relevance under Standard-KB GraphRAG, indicate that the observed patterns reflect systematic effects attenuated by small sample size rather than noise. Cross-KB Mann–Whitney comparisons reach significance for two configurations (Baseline CoT Faithfulness and GraphRAG Few-Shot Overall), confirming that the KB-quality effect persists across both retrieval architectures.

5.6. Structural traceability validation

The four metrics in Section 4 assess requirement quality through token overlap, semantic similarity, vocabulary density, and rule-based syntactic checks. To complement these textual signals with a structural cross-check, the 99 extracted FRs and the 13 in-scope SRS source requirements were imported into a Neo4j graph database, and traceability relationships were established using bidirectional keyword matching against the SRS requirement content. Each extracted FR was classified into one of five categories: fully traced (score 1.0), partially traced (0.5–0.8), off-area (correct SRS domain but assigned to the wrong testing area), out-of-scope (no valid SRS match), or untraced. Results are presented in Table 8.

Table 8. Neo4j structural traceability analysis per configuration

Configuration	FRs	Full Trace	Partial	Out-of-Scope	Off-area	Valid %	Full-trace ratio
Standard KB + Baseline	24	15	7	1	0	92%	68% of valid
Standard KB + GraphRAG	32	17	3	9	3	63%	85% of valid
Refined KB + Baseline	22	17	5	0	0	*100%*	77% of valid
Refined KB + GraphRAG	21	17	4	0	0	*100%*	81% of valid

Note: Each FR was classified into one of five categories based on bidirectional keyword matching against the in-scope SRS: Full Trace (score 1.0), Partial (0.5–0.8), Out-of-Scope (no SRS match), Off-area (matches an SRS requirement in a different testing area), or Untraced. Valid Rate = (Full + Partial) / FRs. Full-Trace Rate = Full / (Full + Partial), expressed as a percentage of valid traces.

As shown in Table 8, both Refined KB configurations reach the maximum valid trace rate, while the Standard KB GraphRAG falls substantially below both Baseline and the Refined KB pair. The 12 invalid traces under Standard KB GraphRAG (9 out-of-scope, 3 off-area) confirm the relevance collapse identified in Table 3 as a structural rather than purely metric phenomenon; these requirements are not merely scoring lower on automated relevance, they are extracting content from outside the in-scope SRS domain. The Refined KB pruning eliminates this failure mode entirely. The full-trace ratio (the percentage of valid FRs achieving score 1.0) is 81% for Refined KB GraphRAG vs 77% for Refined KB Baseline, indicating that graph augmentation does not merely retain validity but preferentially surfaces the most strongly grounded requirements.

6. Ablation study

To verify that the reported results are not artefacts of specific retrieval hyperparameter choices, we conducted a systematic one-factor-at-a-time ablation study. Each parameter was varied independently while all others were held at initial default values (baseline k = 5, GraphRAG seed k = 3, 2-hop traversal, node expansion cap = 5, seed term minimum length = 4). The ablation results informed the selection of optimised parameters k = 3 for both architectures, cap = 3, and minlen = 3, which were applied to the main experimental run reported in Section 4, ensuring equal initial retrieval volume for a fair architectural comparison. Extraction and evaluation were performed using the Zero-Shot prompt strategy on the Refined KB, producing per-area FR sets evaluated with the same four-metric framework. Results are summarised in Table 9.

Table 9. Retrieval parameters ablation: mean quality scores per configuration

Factor / Config	Variable Value	Faithfulness	Relevance	Tech Coverage	Compliance	Overall
Baseline depth k	k = 3	0.976	0.532	0.865	0.886	0.814
	k = 5[a]	0.989	0.508	0.844	0.850	0.798
	k = 7	1.000	0.483	0.819	0.831	0.783
Graph seed k	k = 2	1.000	0.500	0.760	0.867	0.782
	k = 3[a]	1.000	0.496	0.752	0.886	0.783
	k = 5	1.000	0.488	0.766	0.850	0.776
Hop depth	1-hop	1.000	0.431	0.676	0.867	0.743
	2-hop[a]	1.000	0.496	0.752	0.886	0.783
Node expansion cap	cap = 3	1.000	0.496	0.752	0.886	0.784
	cap = 5[a]	1.000	0.496	0.752	0.886	0.783
	cap = 8	1.000	0.496	0.752	0.886	0.783
Seed min length	min = 3	1.000	0.539	0.842	0.900	0.820
	min = 4[a]	1.000	0.496	0.752	0.886	0.783
	min = 5	1.000	0.496	0.752	0.886	0.783

Note: [a]Default initial value; varied for ablation. Ablation conducted on Refined KB with Zero-Shot prompt strategy. Each row varies one parameter while holding the others at their default values.

Four findings emerge from the ablation. First, baseline retrieval depth shows a consistent trade-off: smaller k yields higher Relevance and Overall quality, because additional low-similarity chunks dilute the context passed to the LLM. The default value (k = 5) is therefore mildly suboptimal for this corpus, and adaptive k selection is a candidate for future work. Second, the move from 1-hop to 2-hop traversal is the single largest gain in the table, driven by simultaneous improvements in Relevance and Technical Coverage. This confirms that the value of GraphRAG over a vector-only baseline lies not in the graph itself, but in the multi-hop expansion it enables; a 1-hop graph traversal is essentially equivalent to vector retrieval. Third, the node expansion cap is completely inactive across the tested range, with identical scores for cap values from 3 through 8. This indicates that the effective number of useful graph-expanded nodes per query is consistently below the lowest cap tested, so the parameter is non-critical at the current corpus scale and would only become relevant at substantially larger graph sizes. Fourth, the seed term minimum character length shows a modest improvement from min_len = 3 to min_len = 4, attributable to the inclusion of short but meaningful acronyms (PHI, MFA) that a four-character minimum filters out. This is a deployment-relevant finding for healthcare and other acronym-dense domains, but it falls outside the scope of the main comparison.

Together, the four findings confirm that 2-hop traversal is the architectural choice that drives GraphRAG's quality gains, while the remaining parameters are nearly insensitive within the tested ranges. The configuration applied in the main experiment (k = 3 for both architectures, cap = 3, min_len = 3) follows directly from these results.

7. Threats to validity

Our research threats to validity are based on construct, internal and external validity. The first of these is the applied metrics. All four metrics, adapted from RAGAS, approximate IEEE 29148 quality attributes (correctness, unambiguity, completeness, and conformance), but operate at the token and embedding levels rather than at the semantic or specification levels. Faithfulness relies on word-overlap matching, which can underestimate paraphrased grounding (e.g., "physician" vs. "doctor"). Section 5.6's Neo4j check addresses this by validating grounding at the requirement level rather than the token level.

Internal validity. Statistical tests were conducted on small samples (n = 6–11). Two of six within-KB Wilcoxon comparisons reached $p < 0.05$; the remaining findings are supported by medium-to-large practical effect sizes, rather than conventional significance thresholds. Cross-KB Mann–Whitney tests reached significance for two of five planned comparisons. For more evidence, the experiment should be replicated across larger document corpora.

External validity. The corpus comprises seven healthcare documents covering a single regulatory domain (HIPAA, HITECH, FDA, HL7 FHIR), three security-focused testing areas, and a single base LLM (Claude-3-Haiku-20240307 at temperature zero). The specific empirical patterns reported here, the Standard-vs-Refined KB divergence, the prompt invariance of Refined-KB GraphRAG, and the 100% structural trace rate under both Refined-KB configurations, are therefore directly tied to this corpus, this regulatory scope, and this base model. We do

not claim that the same ordering, the same effect magnitudes, or the same prompt–architecture interaction will hold on larger document sets, different regulatory domains, or different LLMs. The qualitative direction of the Standard-vs-Refined KB effect, however, is consistent with prior findings on retrieval precision in RAG systems (Barnett et al., 2024), suggesting that the underlying mechanism, noise propagation through multi-hop graph traversal, may not be corpus-specific, even if the precise numerical magnitudes reported here are.

8. Conclusions

This study reports three main empirical observations, all derived from a factorial comparison of two KB configurations, two retrieval architectures, and three prompt strategies applied to a seven-document healthcare corpus.

First, the Refined-KB GraphRAG configuration achieved the highest Overall quality score in the study (0.833±0.030, n = 21), with gains over the Refined-KB Baseline driven primarily by Faithfulness (+0.038) and Compliance (+0.020). Second, GraphRAG on the Standard KB produced a systematic Relevance degradation (Cohen's d = −0.45 to −0.55 across all three prompts) that the same architecture eliminated when applied to the Refined KB; the independent Neo4j cross-check confirmed this as a structural effect, with 12 out-of-scope or off-area traces under Standard-KB GraphRAG and zero under both Refined-KB configurations. Third, the Refined-KB GraphRAG configuration exhibited near-zero variance across the three prompt strategies (SD = 0.0008 on the Overall score), suggesting that, in this setting, prompt formulation ceases to act as a primary design lever once KB curation and graph augmentation are applied jointly.

Taken together, these observations reframe GraphRAG, within the bounds of this experimental setup, as a precision multiplier whose benefit is conditional on KB curation rather than as an unconditional improvement over vector retrieval. The 2-hop traversal that gives GraphRAG its expressive power is also the mechanism that propagates retrieval noise when the underlying graph is uncurated, which is consistent with recent observations on RAG failure modes (Barnett et al., 2024).

The findings reported above apply to a seven-document healthcare corpus, three security-focused testing areas, and a single base LLM at temperature zero. Confirmation on larger corpora, broader regulatory scopes, and additional LLMs is required before the empirical patterns above can be treated as general properties of graph-augmented retrieval for requirements engineering. Future work will investigate adaptive traversal depth, hybrid retrieval architectures that combine graph-augmented discovery with vector-only verification, and the regulatory-anchor salience effect identified during analysis.

Acknowledgements

The authors thank the anonymous reviewers of an earlier version of this work for constructive comments that significantly improved the manuscript.

Author contributions

Conceptualization: R.M.D.S. Rathnayake, Asta Slotkienė; methodology: R.M.D.S. Rathnayake, Asta Slotkienė; software: R.M.D.S. Rathnayake; validation: R.M.D.S. Rathnayake; formal analysis: R.M.D.S. Rathnayake; investigation: R.M.D.S. Rathnayake; data curation: R.M.D.S. Rathnayake; writing, original draft: R.M.D.S. Rathnayake; writing, review and editing: R.M.D.S. Rathnayake, Asta Slotkienė; visualization: R.M.D.S. Rathnayake; supervision: Asta Slotkienė; project administration: Asta Slokienė. All authors have read and agreed to the published version of the manuscript.

Disclosure statement

The authors declare no conflict of interest.

References

Abualhaija, S., Basak Aydemir, F., Dalpiaz, F., Dell'Anna, D., Ferrari, A., Franch, X., & Fucci, D. (2024). Replication in requirements engineering: The NLP for RE case. *ACM Transactions on Software Engineering and Methodology*, *33*(6), Article 151. https://doi.org/10.1145/3658669

Aishwarya, V. (2023). A prompt engineering approach for structured data extraction from unstructured text using conversational LLMs. In *Proceedings of the 2023 6th International Conference on Algorithms, Computing and Artificial Intelligence* (*ACAI '23*) (pp. 183–189). Association for Computing Machinery. https://doi.org/10.1145/3639631.3639663

Alhoshan, W., Ferrari, A., & Zhao, L. (2023). Zero-shot learning for requirements classification: An exploratory study. *Information and Software Technology*, *159*, Article 107202. https://doi.org/10.1016/j.infsof.2023.107202

Alhoshan, W., Ferrari, A., & Zhao, L. (2025). *How effective are generative large language models in performing requirements classification?* arXiv. https://doi.org/10.48550/arXiv.2504.16768

Arora, C., Herda, T., & Homm, V. (2024). Generating test scenarios from NL requirements using retrieval-augmented LLMs: An industrial study. In *Proceedings of the IEEE International Conference on Requirements Engineering* (pp. 240–251). IEEE. https://doi.org/10.1109/RE59067.2024.00031

Arvidsson, S., & Axell, J. (2023). *Prompt engineering guidelines for LLMs in Requirements Engineering* (pp. 1–19). Göteborgs universitet. https://gupea.ub.gu.se/handle/2077/77967

Barnett, S., Kurniawan, S., Thudumu, S., Brannelly, Z., & Abdelrazek, M. (2024). Seven failure points when engineering a retrieval augmented generation system. In *Proceedings of the IEEE/ACM 3rd International Conference on AI Engineering – Software Engineering for AI* (*CAIN '24*) (pp. 194–199). Association for Computing Machinery. https://doi.org/10.1145/3644815.3644945

Chen, B., Guo, Z., Yang, Z., Chen, Y., Chen, J., Liu, Z., Shi, C., & Yang, C. (2026). PathRAG: Pruning graph-based retrieval augmented generation with relational paths. *Proceedings of the AAAI Conference on Artificial Intelligence*, *40*(36), 30183–30191. https://doi.org/10.1609/aaai.v40i36.40268

Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., & Larson, J. (2025). *From local to global: A graph RAG approach to query-focused summarization*. arXiv. https://doi.org/10.48550/arXiv.2404.16130

Es, S., James, J., Espinosa-Anke, L., & Schockaert, S. (2024). RAGAS: Automated evaluation of retrieval augmented generation. In *EACL 2024 – 18th Conference of the European Chapter of the Association for Computational Linguistics*, *Proceedings of System Demonstrations* (pp. 150–158). Association for Computational Linguistics. https://doi.org/10.18653/v1/2024.eacl-demo.16

Feng, N., Marsso, L., Yaman, S. G., Standen, I., Baatartogtokh, Y., Ayad, R., De Mello, V. O., Townsend, B., Bartels, H., Cavalcanti, A., Calinescu, R., & Chechik, M. (2024). Normative requirements operationaliza-

tion with large language models. In *Proceedings of the IEEE International Conference on Requirements Engineering* (pp. 129–141). IEEE. https://doi.org/10.1109/RE59067.2024.00022

Guo, Z., Xia, L., Yu, Y., Ao, T., & Huang, C. (2025). LightRAG: Simple and Fast retrieval-augmented generation. In *Findings of the Association for Computational Linguistics*: *EMNLP 2025* (pp. 10746–10761). Association for Computational Linguistics. https://doi.org/10.18653/v1/2025.findings-emnlp.568

IEEE Computer Society. (1998). *IEEE Recommended Practice for Software Requirements Specifications* (Standard No. IEEE Std 830-1998). IEEE. https://doi.org/10.1109/IEEESTD.1998.88286

International Organization for Standardization. (2011). *Systems and software engineering – Life cycle processes – Requirements engineering* (Standard No. ISO/IEC/IEEE 29148:2011). https://doi.org/10.1109/IEEESTD.2011.6146379

Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y., Chen, D., Dai, W., Chan, H. S., Madotto, A., & Fung, P. (2024). Survey of hallucination in natural language generation. *ACM Computing Surveys*, *55*(12), Article 248. https://doi.org/10.1145/3571730

Krishna, M., Gaur, B., Verma, A., & Jalote, P. (2024). Using LLMs in Software Requirements Specifications: An Empirical Evaluation. In *Proceedings of the IEEE International Conference on Requirements Engineering* (pp. 475–483). IEEE. https://doi.org/10.1109/RE59067.2024.00056

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W. T., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems* (*NIPS '20*) (pp. 9459–9474). Curran Associates Inc.

Marron, J. A. (2024). *Implementing the Health Insurance Portability and Accountability Act* (*HIPAA*) *security rule*: *A cybersecurity resource guide* (NIST Special Publication 800-66r2). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-66r2

Motger, Q., & Franch, X. (2024). *NLP-based Relation Extraction Methods in Requirements Engineering*. arXiv. https://doi.org/10.48550/arXiv.2401.12075

Pasquale, L., Ragone, A., Piemontese, E., & Darban, A. A. (2025). Exploring the Use of LLMs for Requirements Specification in an IT Consulting Company. In *Proceedings of the IEEE International Conference on Requirements Engineering* (pp. 389–399). IEEE. https://doi.org/10.1109/RE63999.2025.00045

Sarmah, B., Hall, B., Rao, R., Patel, S., Pasquali, S., & Mehta, D. (2024). HybridRAG: Integrating knowledge graphs and vector retrieval augmented generation for efficient information extraction. In *Proceedings of the 5th ACM International Conference on AI in Finance* (pp. 608–616). Association for Computing Machinery. https://doi.org/10.48550/arXiv.2408.04948

Vogelsang, A., & Fischbach, J. (2024). *Using large language models for natural language processing tasks in requirements engineering*: *A systematic guideline*. arXiv. https://doi.org/10.48550/arXiv.2402.13823

Wei, B. (2024). Requirements are all you need: From requirements to code with LLMs. In *Proceedings of the IEEE International Conference on Requirements Engineering* (pp. 416–422). IEEE. https://doi.org/10.1109/RE59067.2024.00049