TelcoAI: Advancing 3GPP Technical Specification Search through Agentic Multi-Modal Retrieval-Augmented Generation

Rahul Ghosh[1], Chun-Hao Liu[1], Gaurav Rele[1], Vidya Sagar Ravipati[1], Hazar Aouad[2]
[1]Generative AI Innovation Center, Amazon Web Services (AWS)
[2]Bouygyes Telecom
{rahulgh, chunhaol, grele, ravividy}@amazon.com
haouad@bouyguestelecom.fr

Abstract

The 3rd Generation Partnership Project (3GPP) produces complex technical specifications essential to global telecommunications, yet their hierarchical structure, dense formatting, and multi-modal content make them difficult to process. While Large Language Models (LLMs) show promise, existing approaches fall short in handling complex queries, visual information, and document interdependencies. We present TelcoAI, an agentic, multi-modal Retrieval-Augmented Generation (RAG) system tailored for 3GPP documentation. TelcoAI introduces section-aware chunking, structured query planning, metadata-guided retrieval, and multi-modal fusion of text and diagrams. Evaluated on multiple benchmarks—including expert-curated queries—our system achieves 87% recall, 83% claim recall, and 92% faithfulness, representing a 16% improvement over state-of-the-art baselines. These results demonstrate the effectiveness of agentic and multi-modal reasoning in technical document understanding, advancing practical solutions for real-world telecommunications research and engineering.

1 Introduction

The 3rd Generation Partnership Project (3GPP) [1] is responsible for developing and publishing technical specifications for technologies like 3G, 4G, 5G, and beyond, that form the foundation of global mobile telecommunications standards. These specifications are created by various Technical Specification Groups (TSGs) and their associated Working Groups (WGs). The resulting documentation is vast and complex, often spanning hundreds of pages, multiple releases, and containing thousands of equations, tables, figures, and references. This complexity poses significant challenges for engineers and researchers tasked with interpreting, implementing, and staying current with these standards. The emergence of Large Language Models (LLMs) has created new opportunities in the telecommunications domain (Zhou et al., 2024), particularly in their ability to process and understand technical documentation for network design and operations. With the advent of Generative AI (GenAI) and LLMs, there is an unprecedented opportunity to develop a Question-Answering (QA) system specifically tailored to 3GPP documentation (Huang et al., 2025). Such a system would efficiently process and retrieve relevant information from this extensive corpus, significantly reducing the cognitive load on engineers.

LLMs offer multiple approaches for developing effective QA systems for 3GPP documentation, with ongoing debates centering around the optimal implementation strategy. The two primary methodologies under consideration are Fine-Tuning and Retrieval-Augmented Generation (RAG), each presenting distinct advantages and challenges. Fine-tuning involves adapting pre-trained LLMs specifically to the telecommunications domain (Erak et al., 2024), potentially offering more specialized and nuanced responses, but faces limitations due to the scarcity of high-quality training data in the technical domain like 3GPP specifications. RAG, on the other hand, represents a more dynamic non-parametric approach by combining LLMs with real-time information retrieval from vectorized 3GPP documentation (Huang et al., 2025), enabling up-to-date and verifiable responses while maintaining the context of the vast technical specifications. Recent research leveraging RAG-enabled AI platforms shows promising results in reducing human cognitive load through comprehensive solutions for 3GPP documentation, spanning from intelligent digital assistance to automated issue detection (Bornea et al., 2024), knowledge management in telecommunications (Lin, 2023), and 3GPP-aware agentic actors capable of issue-detection and problem-solving (Roychowdhury et al., 2024).

[1] https://www.3gpp.org

Several approaches—such as TSpec-LLM (Nikbakht et al., 2024), Chat3GPP (Huang et al., 2025), Telco-RAG (Bornea et al., 2024), Telco-DPR (Saraiva et al., 2024), and TelecomGPT (Zou et al., 2024)—have explored techniques including domain-specific dataset construction, advanced chunking strategies, hybrid search pipelines, glossary-guided query reformulation, dense passage retrieval, and instruction-tuned LLMs tailored for telecommunications. These methods underscore the importance of specialized data curation, structured retrieval, and domain adaptation to boost performance in technical question answering. However, several key challenges remain. Existing systems often apply flat or generic chunking strategies that fail to leverage the deeply hierarchical and highly structured nature of 3GPP documents, resulting in the loss of crucial contextual cues. Moreover, current approaches largely ignore the multi-modal nature of specifications, neglecting technical diagrams, tables, and visual schematics that are vital for accurate interpretation. From a reasoning perspective, most implementations rely on simple, single-step retrieval patterns and lack agentic capabilities such as structured query decomposition, multi-hop planning, and cross-document information fusion. These limitations are further amplified by the use of evaluation datasets focused on isolated, atomic queries, which do not reflect the complex, multi-faceted information needs of real-world researchers and engineers. Thus there is a need for a more advanced, multi-modal, and agentic framework capable of handling the intricacies of 3GPP documentation.

To address the unique challenges we introduce TelcoAI, an advanced multi-stage RAG system grounded in agentic reasoning and tailored document understanding. Our framework is specifically designed to leverage the hierarchical structure and visual-rich content of 3GPP documents, integrating structural parsing, section-aware chunking, metadata enrichment, and image-text fusion. Through an agentic architecture, the system performs intelligent query decomposition, multi-step planning, and dynamic retrieval refinement to provide accurate, context-aware responses to domain-specific queries. To assess the effectiveness of our approach, we conduct rigorous evaluations across multiple datasets representative of real-world 3GPP use cases. Results demonstrate strong improvements over state-of-the-art baselines for complex, multi-document and version-sensitive queries. Our key contributions are as follows:

Agentic multi-stage RAG architecture: We design a modular retrieval and generation pipeline that integrates hierarchical reasoning, query planning, and answer synthesis—mimicking expert decision-making in telecommunications document navigation.

Multi-modal document understanding: We build a custom ingestion pipeline that fuses visual (e.g., technical diagrams) and textual content using metadata-aware alignment, enabling comprehensive interpretation of 3GPP specs.

Section-aware chunking and structured retrieval: Our method leverages the inherent document hierarchy to preserve semantic coherence across retrieved contexts, improving downstream answer fidelity.

Agentic query decomposition and fusion: We introduce a planner that decomposes complex queries into manageable sub-queries and performs answer fusion over diverse content slices to capture nuanced technical insights.

Comprehensive evaluation: We conduct extensive experiments on curated and synthetic datasets[2] tailored to the 3GPP domain, demonstrating superior performance across multiple retrieval and reasoning tasks.

Together, these contributions represent a significant step toward domain-specialized, multi-modal language systems for high-stakes technical environments such as telecommunications standards development.

2 Related Works

LLMs in 3GPP Domain: Several works suggest that carefully prepared small models or RAG systems can significantly enhance LLMs for the 3GPP domain. In the realm of customized LLMs, researchers have explored various approaches: fine-tuning Microsoft's Phi-2 with LoRA for 3GPP MCQs (Erak et al., 2024), creating Tele-LLM through full fine-tuning on curated datasets (Maatouk et al., 2024) , and fine-tuning BERT variants and GPT-2 for 3GPP documents (Bariah et al., 2023). RAG systems have shown particular promise in the 3GPP domain, with (Nikbakht et al., 2024) demonstrating the potential of naive RAG approaches, while (Huang et al., 2025) enhanced

[2]We are working on releasing the datasets as part of the published paper.

capabilities through hierarchical chunking and hybrid search. Further advancements include Telco-RAG utilizing technical glossaries and neural routing (Bornea et al., 2024) and Telco-DPR offering a curated dataset (Saraiva et al., 2024). Our approach differs from existing works by emphasizing enhanced retrieval and context processing techniques, leveraging general-purpose LLMs while addressing domain-specific challenges through architectural innovations such as comprehensive query reformulation, planning strategies, and integrated image understanding to handle the multi-modal nature of 3GPP technical specifications.

General RAG Architectures: The RAG paradigm has evolved significantly from Naive RAG's straightforward indexing and retrieval (Lewis et al., 2020) to more sophisticated approaches. Advanced RAG (Ma et al., 2023; Zheng et al., 2024) introduced pre-retrieval enhancements like query rewriting and post-retrieval strategies such as re-ranking, improving output quality. Modular RAG further extended these concepts, supporting multiple data modalities (Wang et al., 2023b), leveraging multi-query mechanisms (Rackauckas, 2024), and incorporating memory modules (Cheng et al., 2024) and routing techniques (Mu et al., 2024). Recent developments include task adapters for few-shot adaptation (Cheng et al., 2023; Dai et al., 2022), dynamic frameworks to reduce hallucination (Ma et al., 2023; Khattab et al., 2022), and iterative methods for enhanced context integration (Shao et al., 2023; Jiang et al., 2023; Asai et al., 2023). Our method aligns with recent RAG advancements while addressing unique 3GPP challenges, incorporating Advanced RAG elements in query reformulation, adopting Modular RAG principles for multi-modal handling, echoing iterative retrieval developments through hierarchical chunking and hybrid search, and innovating in document structure preservation for technical content.

3 Method

We present our TelcoAI system architecture (illustrated in Figure 1) which consists of two main components: (1) Document Ingestion, which processes and indexes the technical specifications, and (2) Question Answering, which handles user queries and generates responses.

3.1 Document Ingestion

Document ingestion processes 3GPP technical specifications in Microsoft Word (.docx) format and prepares them for efficient and accurate retrieval. This process involves several key steps: metadata extraction, section-based chunking, table understanding and image tagging.

3.1.1 Metadata Extraction

For each document, we extract structured metadata to facilitate targeted retrieval and filtering. Specifically, we extract *Release number* often prefixed with "R" (e.g., "R16"), *Series number* indicating the document series (e.g., "23"), *Specification number* that is the complete identifier (e.g., "23548-i30"), and a three part *Version* (e.g., "V18.3.0"). This metadata is stored alongside each document chunk in the vector database, enabling filtering of retrieval results based on user queries.

3.1.2 Chunking

Our system implements an innovative structural chunking approach that preserves the natural organization and semantic coherence of 3GPP technical specifications through section-based decomposition. This approach fundamentally differs from traditional RAG systems' fixed-size or basic hierarchical chunking strategies by maintaining the document's inherent structure. The chunking process is formally defined as a recursive function $\mathcal{C}(d)$ that produces a set of chunks $c_1, c_2, ..., c_n$ for a document d. The algorithm implements a depth-first traversal through the document's hierarchical structure:

$$c_i = f_{chunk}(d, h_i, h_{i+1}, s_{max}, o), \quad (1)$$

where h_i and h_{i+1} represent consecutive section headings, s_{max} is the maximum chunk size, and o is the overlap percentage. The algorithm recursively processes sections and subsections until reaching either: (1) a section that fits within s_{max}, or (2) the most granular subsection level.

The chunking process is formally defined as a function $\mathcal{C}(d)$ that produces a set of chunks $c_1, c_2, ..., c_n$ for a document d, where each chunk corresponds to a meaningful section or subsection in the document's structure. The algorithm implements a chunk creation function:

$$c_i = f_{chunk}(d, h_i, h_{i+1}), \quad (2)$$

where h_i and h_{i+1} represent consecutive section headings. To maintain the document's structural

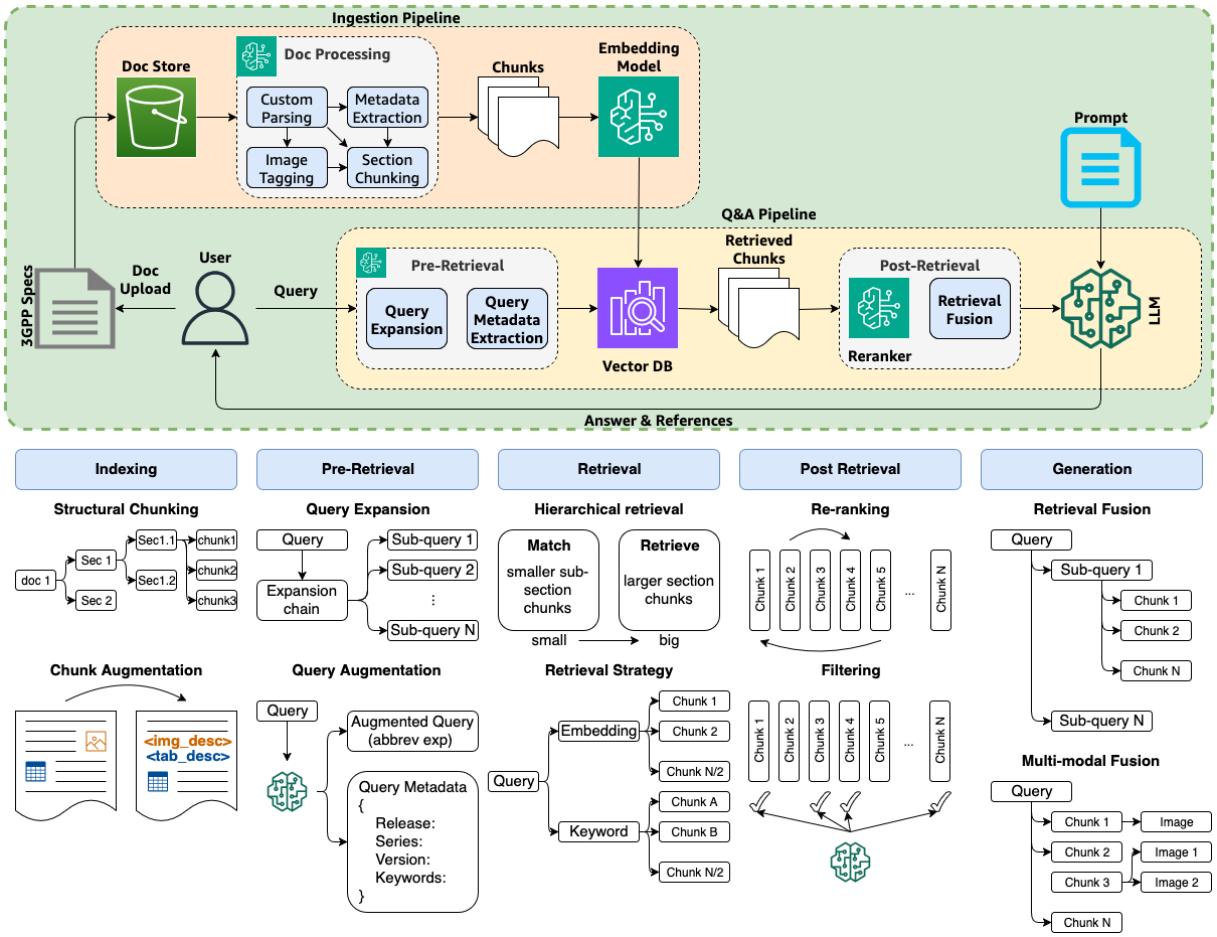

Figure 1: *Overview of TelcoAI's architecture showing both ingestion and Q&A pipelines. Below, detailed workflows show the key components of each stage: indexing, pre-retrieval, context retrieval, post-retrieval and answer generation.*

integrity, each chunk is mapped to its position in the document hierarchy through function $\mathcal{H}(c_i) = (l_i, p_i)$, where l_i represents the heading level and p_i indicates the position within its parent section. This structured approach ensures that semantically related content remains cohesive, even when technical concepts span multiple paragraphs within a logical section. The preserved hierarchical information is crucial for retrieval and answer generation phases, where understanding the context and relationships between different sections becomes essential.

3.1.3 Image Tagging and Integration

To integrate visual and tabular information from 3GPP technical specifications, TelcoAI implements a comprehensive multi-modal processing strategy. This approach embeds visual and tabular information directly into text chunks through natural language descriptions generated using an LLM (we use Anthropic's Claude 3.5 Sonnet v2 (Anthropic, 2024) in our implementation). During indexing, we transform chunks containing visual elements using:

$$c_i' = f_{tag}(c_i, V_j) = c_i \oplus [\text{DESC}]_j \oplus m_j, \quad (3)$$

where c_i' is the transformed chunk, V_j represents either an image I_j or table T_j, $[\text{DESC}]_j$ is the corresponding natural language description, m_j is a unique element marker ([IMG_XXX] - Simple sequential numbering for each document), and $\oplus$ represents concatenation. The description generation follows:

$$[\text{DESC}]_j = \begin{cases} f_{img2text}(I_j) & \text{if } V_j \text{ is an image} \\ f_{tab2text}(T_j) & \text{if } V_j \text{ is a table} \end{cases} \quad (4)$$

This approach preserves semantic relationships between text and visual elements while enabling natural language matching for multi-modal queries. During retrieval and generation, the embedded descriptions participate in semantic matching alongside textual content, while element markers provide access to original visual elements for comprehensive response generation.

3.2 Question Answering

Our question answering pipeline consists of four stages: pre-retrieval processing, context retrieval, post-retrieval processing, and answer generation.

3.2.1 Pre-retrieval

The pre-retrieval stage prepares user queries for effective retrieval through two main techniques: a) Query Reformulation and Planning and b) Query metadata extraction

Query Reformulation and Planning: Real-world questions on 3GPP specifications are often complex, requiring synthesis of information across multiple releases, specifications, and sections, as well as iterative exploration to fully address the query intent.

Example Complex Query

"What are the changes to the standard related to MEC interfaces between versions R17 and R18 and between R18 and R19 of specification 23.558? What are the functional additions of these new interfaces?"

This complexity motivates our advanced query reformulation and planning strategy, designed to bridge the gap between complex user queries and the structured nature of technical documentation. Given a user query q, our query reformulation process leverages an LLM with a planning-aware prompt to develop a question-answering plan and generate a sequence of sub-queries:

$$f_{qr}(q) = \mathcal{M}_{LLM}(p_{qr}, q) = \{q_1, ..., q_m, f_1, ..., f_n\} \quad (5)$$

where q_1 to q_m ($m \leq 3$) are the core sub-queries that directly address the main question, and f_1 to f_n ($m, n \leq 5$) are suggested follow-up queries planned by the LLM to enable deeper exploration of related aspects. Here p_{qr} is a carefully designed query reformulation and planning prompt, and $\mathcal{M}_{LLM}$ is the LLM. Each query is crafted to be standalone, retaining the full context of the original question, including crucial elements such as specification references and release information. This structured approach enables more precise retrieval and allows our system to systematically explore complex technical concepts through planned, iterative questioning.

Query Metadata Extraction: To leverage the 3GPP technical structure, we extract metadata and expand abbreviations from user queries to enable focused retrieval. Our metadata extraction function processes each query to extract release numbers (e.g., "R16", "R17"), series numbers (e.g., "23"), and specification identifiers (e.g., "23.588.h00"). The extraction process is implemented using an LLM with a carefully designed metadata extraction and abbreviation extraction prompt. For example, from the query "Compare the architecture changes in specification 23.558 between R17 and R18", the system extracts "release": ["17", "18"], "series": ["23"], "specification": ["23.558"], enabling focused retrieval by filtering candidate chunks to those matching the specific releases and specifications mentioned in the query.

3.2.2 Context Retrieval

We perform context retrieval from our knowledge base through a multi-stage process that combines dense vector representation and hybrid search strategies. For each sub-query q_i from the reformulation step, we perform context retrieval from our knowledge base through hybrid search, and hierarchical retrieval. First, we encode the query into a dense vector representation, $\mathbf{v}_q = \mathcal{E}(q_i)$. We perform a hierarchical hybrid search combining dense vector similarity and sparse lexical matching:

$$sim(q_i, c) = \alpha \cdot cos(\mathbf{v}_q, \mathbf{v}_c) + (1 - \alpha) \cdot lex(q_i, c), \quad (6)$$

where $cos(\mathbf{v}_q, \mathbf{v}_c)$ is the cosine similarity between query and chunk vectors, $lex(q_i, c)$ is a lexical similarity function based on BM25 (Robertson and Zaragoza, 2009), and α is a weighting parameter (set to 0.5 in our experiments). The search operates hierarchically, first retrieving smaller sub-section chunks and then expanding to include their parent sections.

3.2.3 Post-Retrieval Processing

The retrieved chunks, initially consisting of the top 40 matches, undergo metadata-based filtering using attributes extracted during pre-retrieval, keeping only chunks with matching metadata. The remaining chunks are then reranked based on similarity scores, with the top 5 selected for the final result, ensuring both content relevance and metadata alignment.

3.2.4 Answer Generation

The answer generation phase of our system synthesizes information from multiple retrieved chunks and modalities to produce comprehensive responses to user queries. This process involves three

key components: retrieval fusion, multi-modal fusion, and structured response generation.

Retrieval Fusion: For each sub-query q_i, multiple chunks are retrieved, potentially containing complementary or overlapping information. Our retrieval fusion mechanism combines these chunks into a coherent context:

$$\mathcal{C}(q) = \bigcup_{q_i \in f_{qr}(q)} \mathcal{R}(q_i). \quad (7)$$

The chunks are ordered by relevance score to prioritize the most salient information. For chunks containing image references, we retrieve the associated images using our image mapping:

$$I_c = I_j | \exists \mathcal{T}(I_j) = (c, cap_j, ref_j). \quad (8)$$

Multi-modal Fusion: Our system performs multi-modal fusion by combining textual and visual information:

$$\mathcal{M}(q_i) = f_{multimodal}(\mathcal{C}(q), I_1, ..., I_m), \quad (9)$$

where $I_1, ..., I_m$ are the images referenced by the markers in the retrieved chunks. This fusion process ensures that technical concepts are explained with appropriate visual support, particularly crucial for architectural descriptions and complex technical diagrams.

Response Generation: The final response is generated using the Anthropic Claude 3.5 Sonnet v2 model (Anthropic, 2024):

$$y = \mathcal{M}_{LLM}(p_{gen}, q, \mathcal{C}(q), I_{\mathcal{C}(q)}), \quad (10)$$

where p_{gen} is a carefully designed prompt that instructs the model to analyze the question and context thoroughly, identify relevant information across multiple chunks, incorporate visual elements with appropriate context, perform necessary technical reasoning, and generate a comprehensive answer with explicit references to specifications and sections. For queries requiring multiple steps or comparisons across releases, the system maintains coherence while building a comprehensive answer that addresses all aspects of the original question.

4 Experiments

4.1 Datasets

Our evaluation utilizes four distinct datasets designed for 3GPP technical specifications: Human-QA, Synthetic-QA, Telco-DPR, and TSpec-LLM.

Human-QA, our expert-curated dataset, contains 41 technical questions spanning 12 specifications from Releases 16-19, with questions categorized by complexity and requiring single-document retrieval (60%), cross-document synthesis (25%), or version comparisons (15%). **Synthetic-QA** (Guinet et al., 2024) uses the same document corpus but employs Anthropic Claude 3.5 Sonnet v2 (Anthropic, 2024) with custom prompting to generate multiple-choice questions, validated using Mistral Large 2 (Mistral AI team, 2024) and human experts. **TSpec-LLM** (Nikbakht et al., 2024) offers the most comprehensive coverage with 30,137 documents spanning Releases 8-19 (535 million words), preserving original structure and content, and includes an evaluation set of 100 multiple-choice questions categorized by difficulty level.

4.2 Evaluation Metrics

For the Human-QA and Synthetic-QA datasets, we evaluate using RAGChecker (Ru et al., 2024) metrics, which provide both overall performance assessment and detailed diagnostic insights. The evaluation is based on claim-level analysis, where responses and ground-truth answers are decomposed into individual claims for precise comparison. We compute claim-level metrics as follows. Given a model response m and a ground-truth answer gt, we first extract their respective claims:

$$Claims_{model} = c_i^{(m)}, Claims_{gt} = c_i^{(gt)}. \quad (11)$$

We then compute recall that evaluates the proportion of ground-truth claims present in the model response:

$$R = \frac{|c_i^{(gt)} | c_i^{(gt)} \in m|}{|c_i^{(gt)}|}. \quad (12)$$

We focus on recall as our primary metric since human-generated ground truth answers tend to be concise and targeted, while model-generated responses often include additional contextual details. In this scenario, recall effectively measures whether the model captures all information from the ground truth, without penalizing the inclusion of relevant supplementary details. For TSpec-LLM, we evaluate accuracy across difficulty levels (Overall, Easy, Intermediate, and Hard) to assess the system's performance on questions of varying complexity.

4.3 Baselines

Our evaluation compares our method against several baselines, including base LLMs and special-

ized RAG systems for 3GPP documentation. **Base LLMs** like Gemini 1.0 (Gemini Team, Google, 2023), GPT-4 (OpenAI, 2023), Mistral Large 2 (Mistral AI team, 2024), Llama 3.3 70B Instruct (Meta AI, 2024), and Anthropic Claude 3.5 Sonnet v2 (Anthropic, 2024), while demonstrating broad language understanding, lack crucial components for effectively handling 3GPP technical documentation, such as access to the latest specifications, metadata awareness, and multimodal fusion capabilities. **Naive RAG** implements the standard retrieve-then-generate pattern without any domain-specific optimizations. **Telco-RAG** (Bornea et al., 2024) implements a dual-stage pipeline with query enhancement and retrieval stages. **Chat3GPP** (Huang et al., 2025) uses hierarchical chunking and hybrid retrieval. While these baseline methods were originally evaluated with specific generation models in their respective papers, we extend their evaluation by testing each method with multiple state-of-the-art generation models (Gemini 1.0, GPT-4, Mistral Large 2, Llama 3.3 70B Instruct, and Claude 3.5 Sonnet v2) to ensure a comprehensive comparison across different language models.

4.4 Implementation Details

We implement our system using Anthropic Claude 3.5 Sonnet v2 (Anthropic, 2024) as the base LLM and Amazon Titan Text Embeddings v2 (1024 dimensions) (Amazon Web Services, 2024) for embeddings. For comparison, we evaluate against baseline RAG systems combining two retrievers (BM25 and E5-Mistral (Wang et al., 2023a)) with five generation models (Gemini 1.0, GPT-4, Mistral Large 2, Llama 3.3 70B Instruct, and Claude 3.5 Sonnet v2). Our implementation uses a temperature of 0.7, retrieves top-5 chunks with size of 512 tokens and 20% overlap. BM25 uses Elasticsearch (Elastic NV, 2010) with default parameters, while dense retrieval uses Titan Text Embeddings v2. All systems use identical chunk size, overlap settings, and retrieval parameters for fair comparison. Evaluation employs RAGChecker with Llama 3.3 70B Instruct as both claim extractor and checker models for detailed performance analysis, providing metrics for overall effectiveness, retrieval quality, and generation reliability.

Table 1: *Performance comparison (in Recall R and Claim Recall CR) across different methods and generation models on Human-QA and Synthetic-QA datasets. Human-QA contains expert-curated complex questions requiring multi-document synthesis, while Synthetic-QA contains LLM generated questions validated by experts. Recall (R) measures overall performance while Claim Recall (CR) evaluates retriever effectiveness. Base models have no CR values as they operate without retrievers. Within each method category (Naive RAG, Telco-RAG, and Chat3GPP), the retriever remains constant, resulting in a single CR value per category.*

Method	Generation Model	Human QA		Synthetic QA	
		R	CR	R	CR
Base Model	Gemini 1.0	0.34	↑	0.31	↑
	GPT 4	0.38	|	0.35	|
	Mistral Large 2	0.22	NA	0.2	NA
	Llama 3.3 70B Instruct	0.35	|	0.32	|
	Claude 3.5 Sonnet v2	0.42	↓	0.45	↓
Naive RAG	Gemini 1.0	0.56	↑	0.51	↑
	GPT 4	0.61	|	0.55	|
	Mistral Large 2	0.54	0.6	0.49	0.57
	Llama 3.3 70B Instruct	0.6	|	0.54	|
	Claude 3.5 Sonnet v2	0.63	↓	0.57	↓
Telco-RAG	Gemini 1.0	0.62	↑	0.56	↑
	GPT 4	0.65	|	0.59	|
	Mistral Large 2	0.64	0.67	0.58	0.62
	Llama 3.3 70B Instruct	0.68	|	0.61	|
	Claude 3.5 Sonnet v2	0.72	↓	0.65	↓
Chat 3GPP	Gemini 1.0	0.64	↑	0.58	↑
	GPT 4	0.71	|	0.64	|
	Mistral Large 2	0.68	0.73	0.61	0.72
	Llama 3.3 70B Instruct	0.72	|	0.65	|
	Claude 3.5 Sonnet v2	0.75	↓	0.73	↓
TelcoAI	Claude 3.5 Sonnet v2	0.87	0.83	0.85	0.83

5 Results

5.1 Real-world Results

Table 1 presents the performance comparison of our method against various baselines on the Human-QA and Synthetic-QA datasets, which represent real-world scenarios in querying 3GPP technical specifications. As expected, the results show a clear progression in performance from base models to specialized RAG systems. **Base Models** demonstrate limited effectiveness in handling 3GPP technical queries, with performance varying significantly across different LLMs. **Naive RAG** implementations show marked improvements over base models. The consistent improvement across both datasets (average improvement of 20-25% over base models) demonstrates the fundamental value of retrieval augmentation in processing technical specifications. **Specialized Systems** (Telco-RAG and Chat3GPP) further enhance performance through domain-specific optimizations. Chat3GPP with Claude 3.5 Sonnet v2 achieves the highest baseline performance (Human-QA R : 0.75, CR : 0.73; Synthetic-QA R : 0.73, CR : 0.72),

Table 2: *Ablation study results (in Recall R and Claim Recall CR) on Human-QA dataset showing the incremental performance improvement with the addition of each component. Starting from Naive RAG baseline, each row represents the cumulative effect of adding the corresponding component. The final row shows the overall improvement percentage over the baseline in parentheses.*

Stage	Method	R	CR
	Baseline (Naive RAG)	0.63	0.6
Indexing	Chunking	0.72	0.73
Pre-retrieval	Query Expansion	0.74	0.77
Retrieval	Hierarchical Retrieval	0.75	0.78
	Hybrid Retrieval	0.78	0.79
Post Retrieval	Reranking	0.79	0.8
	Filtering	0.82	0.81
Generation	Multi-modal Fusion	0.87 (38.1%)	0.83 (38.3%)

Table 3: Performance of TelcoAI across different generation models on the Human-QA dataset, demonstrating consistent effectiveness independent of the underlying LLM.

Generation Model	**Recall**
Claude 3.5 Sonnet v2	**0.87**
Claude 3 Opus	0.85
Llama 3.3 70B Instruct	0.84
Mistral Large 2	0.83

representing a significant improvement over naive RAG approaches. **TelcoAI** achieves the highest performance across both datasets, with consistent improvements over the best baseline. Specifically, our method achieves 16% and 13.7% improvement in Recall (0.87 vs 0.75) and Claim Recall (0.83 vs 0.73) respectively in the Human-QA dataset, and 16.4% and 15.3% improvement in Recall (0.85 vs 0.73) and Claim Recall (0.87 vs 0.72) respectively in the SyntheticQA dataset.

5.2 Ablation Studies

We conducted comprehensive ablation studies on the Human-QA dataset to understand each component's contribution to our system's performance. Starting from a Naive RAG baseline ($R : 0.63$, $CR : 0.6$), we incrementally added components across different pipeline stages, measuring their impact on both Recall (R) and Claim Recall (CR). Our section-based chunking strategy in the indexing stage provides the first significant improvement ($R : 0.72$, $CR : 0.73$), showing a 14.3% increase in Recall over the baseline. This demonstrates the importance of preserving document structure and semantic coherence in technical specification processing. Adding query expansion techniques in the pre-retrieval stage further enhances performance ($R : 0.74$, $CR : 0.77$), highlighting the value of comprehensive query understanding.

Table 4: Latency comparison across different methods and pipeline stages. All measurements in seconds, using Claude 3.5 Sonnet v2 as the generation model.

Method	**Pre-Ret. (s)**	**Retrieval (s)**	**Post-Ret. (s)**	**Generation (s)**	**Overall (s)**
Base LLM	-	-	-	6.73	6.73
Naive RAG	-	0.5	-	6.26	6.76
Telco-RAG	2.5	0.8	-	6.61	9.91
Chat3GPP	-	1.0	-	7.07	8.07
TelcoAI	**2.7**	**1.2**	**0.7**	**7.21**	**11.81**

In the retrieval stage, hierarchical retrieval first improves performance ($R : 0.75$, $CR : 0.78$), followed by additional gains from hybrid retrieval ($R : 0.78$, $CR : 0.79$). Post-retrieval processing, comprising re-ranking and filtering, contributes substantial improvements ($R : 0.82$, $CR : 0.81$) by refining the retrieved context. The final addition of multi-modal fusion in the generation stage yields the highest performance ($R : 0.87$, $CR : 0.83$), representing a total improvement of 38.1% in Recall and 38.3% in Claim Recall over the baseline. This significant gain underscores the importance of integrating visual information with textual content in technical documentation understanding. The consistent improvements across both metrics throughout the ablation study validate our system's architecture and the contribution of each component to the overall performance.

To assess the robustness of TelcoAI across different generation models, we extended our evaluation beyond Claude 3.5 Sonnet v2 to include several state-of-the-art LLMs. Table 3 presents the performance comparison across four diverse models spanning different architectures and parameter scales. TelcoAI demonstrates robust performance across all evaluated models, with Recall scores ranging from 0.83 to 0.87 on the Human-QA dataset. The variation of only 2-4% across different generators indicates that our framework's effectiveness is not overly dependent on any specific generation model.

5.3 Latency Analysis

To evaluate the practical feasibility of our approach, we conducted comprehensive latency benchmarking across all pipeline modules. Table 4 presents the performance breakdown for different methods using Claude 3.5 Sonnet v2 as the generation model. TelcoAI achieves a total latency of 11.81s, which represents a reasonable trade-off for the significant accuracy improvements obtained. While this is higher than base LLM (6.73s) and Naive

RAG (6.76s), it remains comparable to other specialized systems such as Telco-RAG (9.91s) and Chat3GPP (8.07s). Importantly, the 3-5s additional latency over competing specialized systems translates to substantial accuracy gains: +16% Recall and +13.7% Claim Recall over Chat3GPP on the Human-QA dataset.

The latency breakdown reveals several insights. Pre-retrieval processing (2.7s) remains close to Telco-RAG (2.5s), indicating that our query expansion and decomposition techniques add minimal overhead. Retrieval time increases slightly (1.2s vs 0.8-1.0s for baselines) due to our hierarchical and hybrid retrieval mechanisms, but this investment enables more comprehensive context gathering. Post-retrieval processing (0.7s) introduces additional overhead for re-ranking and filtering, but proves essential for context quality improvement as demonstrated in our ablation studies. Generation times remain stable across all methods (6-7s), confirming that our performance gains stem from improved context preparation rather than extended generation processes.

5.4 Benchmarking Results

To further validate our system's factual accuracy, we evaluated its performance on the TSpec-LLM benchmark dataset. While this benchmark primarily consists of single-chunk, fact-based multiple-choice questions (simpler than the multi-document, multi-modal queries encountered in real-world scenarios) it provides a valuable test of precise information retrieval. Table 5 shows our method achieving state-of-the-art performance with an overall accuracy of 93%, demonstrating consistent performance across difficulty levels (Easy: 93%, Intermediate: 93%, Hard: 92%).

The results show a clear progression of performance improvements across different approaches. Base models demonstrate limited effectiveness, with accuracies ranging from 30% (Mistral Large 2) to 67% (Claude 3.5 Sonnet v2). The introduction of Naive RAG significantly improves performance across all models, with accuracies increasing to 73-82%. Specialized systems like Telco-RAG and Chat3GPP further enhance performance, with Chat3GPP using Claude 3.5 Sonnet v2 achieving 92% overall accuracy.

Thus, these results demonstrate that TelcoAI maintains high precision in fact-based scenarios where exact accuracy is crucial. The consistent improvement over strong baselines across all difficulty levels validates the robustness of our approach in both accuracy over fact-based and comprehensiveness in real-world QA.

Table 5: *Performance (in accuracy %) comparison on TSpec-LLM benchmark across different methods and generation models. Questions are categorized by difficulty level: Easy, Intermediate (Inter.), and Hard.*

Method	Generation Model	Overall	Easy	Inter.	Hard
Base Model	Gemini 1.0	46	67	37	36
	GPT 4	51	80	47	26
	Mistral Large 2	30	37	26	26
	Llama 3.3 70B Instruct	47	63	45	26
	Claude 3.5 Sonnet v2	67	73	71	47
Naive RAG	Gemini 1.0	75	93	65	66
	GPT 4	82	89	85	61
	Mistral Large 2	73	77	75	63
	Llama 3.3 70B Instruct	81	87	82	68
	Claude 3.5 Sonnet v2	82	90	80	74
Telco RAG	Gemini 1.0	83	87	87	64
	GPT 4	84	90	87	64
	Mistral Large 2	85	90	84	79
	Llama 3.3 70B Instruct	85	93	86	79
	Claude 3.5 Sonnet v2	91	90	92	90
Chat 3GPP	Gemini 1.0	85	90	82	76
	GPT 4	86	92	87	75
	Mistral Large 2	87	88	86	82
	Llama 3.3 70B Instruct	87	90	84	90
	Claude 3.5 Sonnet v2	92	91	92	91
TelcoAI	Claude 3.5 Sonnet v2	93	93	93	92

6 Conclusions

We presented TelcoAI, an advanced RAG system specifically designed for processing 3GPP technical specifications, incorporating sophisticated query planning, hierarchical retrieval, and multi-modal fusion capabilities. Our comprehensive evaluation demonstrates significant improvements over existing approaches, achieving 16% improvement in recall on complex real-world queries and 93% accuracy on benchmark tasks. The system shows strong performance in both complex scenarios and simpler benchmark tasks, with high faithfulness (0.92) and minimal hallucination (0.05) on multi-document queries. The ablation studies validate our architectural choices, showing meaningful contributions from each component. While opportunities for future work remain, such as handling more diverse visual elements and developing sophisticated cross-version comparison capabilities, our current implementation represents a substantial step forward in making 3GPP technical documentation more accessible and efficiently processable, potentially reducing the cognitive load on engineers working with these complex specifications.

Limitations

While our agentic multi-modal RAG system advances the state-of-the-art in 3GPP technical specification search, it has several limitations. First, the current implementation is tailored to .docx documents containing both text and embedded diagrams, which aligns with the format of 3GPP specifications. However, other relevant multi-modal sources in the telecommunications domain—such as PDFs with scanned images, Markdown files with external media, and presentation slides—are not yet supported. Expanding multi-modal ingestion to accommodate these diverse formats remains future work. Second, our system is limited to single-turn queries and does not model multi-turn interactions or conversational history, which are important for complex, context-dependent tasks in technical support scenarios. Future work should explore multi-step search agents that can decompose complex queries, maintain conversational state, and iteratively refine searches based on intermediate results. Incorporating agentic workflows—such as an agentic planner for action planning and execution monitoring—could enable sophisticated reasoning chains where the system autonomously decides when to retrieve additional context, cross-reference multiple specifications, or request clarification from users. Extending our framework to handle multi-turn, multi-modal dialogue is a promising direction. Lastly, while we emphasize accuracy and capability in this research prototype, we have not yet optimized for system efficiency, inference latency, or integration overhead, which are critical for real-world deployment. Addressing these engineering aspects is necessary for transitioning our system from research to production.

References

Amazon Web Services. 2024. Amazon Titan Text Embeddings V2. https://aws.amazon.com/about-aws/whats-new/2024/04/amazon-titan-text-embeddings-v2-amazon-bedrock/. General availability announcement for Titan Text Embeddings V2 on Amazon Bedrock.

Anthropic. 2024. Claude 3.5 sonnet model card addendum.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

Lina Bariah, Hang Zou, Qiyang Zhao, Belkacem Mouhouche, Faouzi Bader, and Merouane Debbah. 2023. Understanding telecom language through large language models. In *GLOBECOM 2023-2023 IEEE Global Communications Conference*, pages 6542–6547. IEEE.

Andrei-Laurentiu Bornea, Fadhel Ayed, Antonio De Domenico, Nicola Piovesan, and Ali Maatouk. 2024. Telco-rag: Navigating the challenges of retrieval-augmented language models for telecommunications. *arXiv preprint arXiv:2404.15939*.

Daixuan Cheng, Shaohan Huang, Junyu Bi, Yuefeng Zhan, Jianfeng Liu, Yujing Wang, Hao Sun, Furu Wei, Weiwei Deng, and Qi Zhang. 2023. UPRISE: Universal prompt retrieval for improving zero-shot evaluation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 12318–12337, Singapore. Association for Computational Linguistics.

Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. 2024. Lift yourself up: retrieval-augmented text generation with self-memory. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*, NIPS '23, Red Hook, NY, USA. Curran Associates Inc.

Zhuyun Dai, Vincent Y. Zhao, Ji Ma, Yi Luan, Jianmo Ni, Jing Lu, Anton Bakalov, Kelvin Guu, Keith B. Hall, and Ming-Wei Chang. 2022. Promptagator: Few-shot dense retrieval from 8 examples. *arXiv preprint arXiv:2209.11755*.

Elastic NV. 2010. Elasticsearch: A Distributed RESTful Search and Analytics Engine. https://www.elastic.co/elasticsearch/.

Omar Erak, Nouf Alabbasi, Omar Alhussein, Ismail Lotfi, Amr Hussein, Sami Muhaidat, and Merouane Debbah. 2024. Leveraging fine-tuned retrieval-augmented generation with long-context support: For 3gpp standards. *arXiv preprint arXiv:2408.11775*.

Gemini Team, Google. 2023. Gemini: A Family of Highly Capable Multimodal Models. arXiv preprint arXiv:2312.11805.

Gauthier Guinet, Behrooz Omidvar-Tehrani, Anoop Deoras, and Laurent Callot. 2024. Automated evaluation of retrieval-augmented language models with task-specific exam generation. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Long Huang, Ming Zhao, Limin Xiao, Xiujun Zhang, and Jungang Hu. 2025. Chat3gpp: An open-source retrieval-augmented generation framework for 3gpp documents. *arXiv preprint arXiv:2501.13954*.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Active retrieval augmented generation. In *Proceedings of the 2023*

Conference on Empirical Methods in Natural Language Processing, pages 7969–7992, Singapore. Association for Computational Linguistics.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive NLP. *arXiv preprint arXiv:2212.14024*.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

Xingqin Lin. 2023. Artificial intelligence in 3gpp 5g-advanced: A survey. *arXiv preprint arXiv:2305.05092*.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting in retrieval-augmented large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 5303–5315, Singapore. Association for Computational Linguistics.

Ali Maatouk, Kenny Chirino Ampudia, Rex Ying, and Leandros Tassiulas. 2024. Tele-llms: A series of specialized large language models for telecommunications. *arXiv preprint arXiv:2409.05314*.

Meta AI. 2024. Llama 3.3 70B Instruct. `https://huggingface.co/meta-llama/Llama-3.3-70B-Instruct`. Instruction-tuned, multilingual; released Dec 6, 2024.

Mistral AI team. 2024. Large Enough. `https://mistral.ai/news/mistral-large-2407`.

Feiteng Mu, Yong Jiang, Liwen Zhang, Liuchu Liuchu, Wenjie Li, Pengjun Xie, and Fei Huang. 2024. Query routing for homogeneous tools: An instantiation in the RAG scenario. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10225–10230, Miami, Florida, USA. Association for Computational Linguistics.

Rasoul Nikbakht, Mohamed Benzaghta, and Giovanni Geraci. 2024. Tspec-llm: An open-source dataset for llm understanding of 3gpp specifications. *arXiv preprint arXiv:2406.01768*.

OpenAI. 2023. GPT-4 Technical Report. arXiv preprint arXiv:2303.08774.

Zackary Rackauckas. 2024. Rag-fusion: A new take on retrieval augmented generation. *International Journal on Natural Language Computing*, 13(1):37–47.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Sujoy Roychowdhury, Sumit Soman, H G Ranjani, Neeraj Gunda, Vansh Chhabra, and Sai Krishna Bala. 2024. Evaluation of rag metrics for question answering in the telecom domain. *arXiv preprint arXiv:2407.12873*.

Dongyu Ru, Lin Qiu, Xiangkun Hu, Tianhang Zhang, Peng Shi, Shuaichen Chang, Cheng Jiayang, Cunxiang Wang, Shichao Sun, Huanyu Li, and 1 others. 2024. Ragchecker: A fine-grained framework for diagnosing retrieval-augmented generation. *Advances in Neural Information Processing Systems*, 37:21999–22027.

Thaina Saraiva, Marco Sousa, Pedro Vieira, and António Rodrigues. 2024. Telco-dpr: A hybrid dataset for evaluating retrieval models of 3gpp technical specifications. *arXiv preprint arXiv:2410.19790*.

Zhihong Shao, Yeyun Gong, Yelong Shen, Minlie Huang, Nan Duan, and Weizhu Chen. 2023. Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9248–9274, Singapore. Association for Computational Linguistics.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2023a. Improving text embeddings with large language models. *arXiv preprint arXiv:2401.00368*.

Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. 2023b. Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases. *arXiv preprint arXiv:2308.11761*.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024. Take a step back: Evoking reasoning via abstraction in large language models. In *The Twelfth International Conference on Learning Representations*.

Hao Zhou, Chengming Hu, Ye Yuan, Yufei Cui, Yili Jin, Can Chen, Haolun Wu, Dun Yuan, Li Jiang, Di Wu, Xue Liu, Charlie Zhang, Xianbin Wang, and Jiangchuan Liu. 2024. Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities. *Preprint*, arXiv:2405.10825.

Hang Zou, Qiyang Zhao, Yu Tian, Lina Bariah, Faouzi Bader, Thierry Lestable, and Merouane Debbah. 2024. Telecomgpt: A framework to build telecom-specfic large language models. *arXiv preprint arXiv:2407.09424*.

A Appendix

A.1 Tools

A.1.1 Query Reformulation

Prompt

<instruction> You are a helpful assistant that breaks a user question into multiple consequent sub queries for search, if required. Write the subqueries so that they are standalone with full context and can be understood with just by themselves. Do not create new queries with facts other than what is mentioned in the original question Do not break a query into subqueries and lose the context. do not lose the context of specification and release. Create subqueries only when necessary. if not needed keep the query as is. Generate sub queries, one on each line, related to the query given in <query> section. Do not break a query into subqueries and lose the context. do not lose the context of specification and release. Be judicious with the number of subqueries generated. if the user question is simple enough to not be broken into sub queries return the original question. Do not use your self-knowledge about the domain. Only state in the sub-queries what is present in the original user question. Do not go beyond 3 sub queries. Do not ask questions about specifications. </instruction>
<query> query </query>
<format> "sub_queries": ['Your search question 1 here.', 'Your search question 2 here.', ... 'Your search question n here.'] </format>
Provide your response immediately without any preamble/explain/additional text, enclosed in valid JSON format.

Example :

Example Complex Query

"How has the handling of network slicing evolved in the core network authentication process between R16, R17 and R18 of specification 33.501? What security enhancements were introduced in each version?"

Generated Query Plan

Initial Queries:
What changes were made to network slice authentication between R16 and R17 in specification 33.501?
What changes were made to network slice authentication between R17 and R18 in specification 33.501?
What specific security enhancements for network slicing were added in each release?
Follow-up Analysis:
Analyze the key security mechanisms introduced in each release
Compare authentication procedures across different slice types
Investigate backward compatibility considerations

A.1.2 Query Metadata Extraction

Prompt

You are an expert in 3GPP (3rd Generation Partnership Project) Technical Specifications; these are specifications relevant to mobile communications.
<instruction> - Parse a 3GPP query to extract the three entities contained therein, i.e., release, series, and specification. - A 'release' (may also be referred to as a 'version') is a number such as 16, 17, ... 19, etc. A release may also be preceded by an 'R', like this: R16, R17, etc. - A 'series' is typically a number, such as 23. - A specification is string, typically starting with a series followed by a more digits, with possibly a sub-identifier, like this: 23588, 23.588, or 23.588.h00. - Hint: if you are confident of the specification (e.g., 23588), then the release is the first two digits, like this: 23. - If none of the three can be extracted from the queries, return with null. - Return your answer as JSON with no other commentary. - Always put the values in a list - use the following databank if some information is missing </instruction>
<query> query </query>

A.2 Prompts

A.2.1 Answer Generation

Prompt

You are a 3GPP Question Answering agent specialized in providing accurate and substantiated responses to technical queries based on given context. Your persona is that of a knowledgeable and meticulous expert who carefully analyzes the question, relevant context, and performs necessary calculations to arrive at a precise and well-reasoned answer.
<Task> Given a user question enclosed in <question> tags and potentially relevant context provided in <context> tags, your task is to provide a complete and correct answer to the question based on the given context. </Task>
<instruction> To answer the question, think step by step: 1. Read the question carefully and thoroughly understand its requirements. 2. Review the provided context paragraph(s) and identify all information relevant to answering the question. 3. If required, perform any necessary calculations such as sums, divisions, or other operations to derive the answer. Show your work and math clearly. 4. Formulate a final answer that directly responds to the question, grounded in and substantiated by the given context. 5. Present any numerical values in your answer in a rounded and easy-to-read format with appropriate units. 6. Enclose your final answer in an <answer></answer> tag and all the "doc_name" used to answer the questions in <docs></docs> tag. 7. Include all relevant and exhaustive information from the context to support your answer. Provide a clear explanation justifying your response. 8. Do not answer the question with "Based on the provided context" or anything similar. Just providing answer is enough. 9. If image is not provided, try to answer using the textual context that should contain figure explanation. </instruction>
<question> question </question>
<context> context </context>

Table 6: *Detailed performance analysis on Human-QA dataset using RAGChecker metrics.*

Overall Metrics	Precision	0.35
	Recall	0.87
	F1	0.46
Retriever Metrics	Claim Recall	0.83
	Context Precision	0.34
Generator Metrics	Context Utilization	0.71
	Hallucination	0.05
	Self Knowledge	0.03
	Faithfulness	0.92

A.3 Detailed Performance Analysis

Table 6 provides comprehensive insights into our method's performance on the Human-QA dataset using RAGChecker metrics. The overall effectiveness metrics show a high Recall (0.87) combined with moderate Precision (0.35), yielding an F1 score of 0.46. Note, that the low precision is choice to prioritize comprehensive information retrieval, often providing additional relevant context beyond the ground truth answers, rather than indicating inaccuracy in the responses.

The retrieval quality metrics demonstrate strong performance in information gathering, with high Claim Recall (0.83) paired with lower Context Precision (0.34). This pattern indicates successful capture of essential information while including broader contextual content from the technical specifications. Such behavior is particularly beneficial in the 3GPP domain, where technical concepts are often interconnected and additional context enhances understanding.

Our system exhibits exceptional reliability in response generation, as evidenced by the generator metrics. Strong Context Utilization (0.71) demonstrates effective use of retrieved information, while minimal Hallucination (0.05) and Self Knowledge (0.03) scores indicate strict grounding in the source documentation. The high Faithfulness score (0.92) further confirms that our generated responses accurately represent the technical specifications. These metrics validate our system's effectiveness in handling complex 3GPP documentation tasks, successfully balancing comprehensive coverage with accurate information representation.