Title

DEEPSPECS Expert-Level Questions Answering in 5G

ISBN

9798265483416

Author

Manvattira, Aman Ganapathy

Publication Date

2025-12-11

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA

Los Angeles

DEEPSPECS

Expert-Level Questions Answering in 5G

A thesis submitted in partial satisfaction of the

requirements for the degree Master of Science in Computer Science

by

Aman Ganapathy Manvattira

2025

ABSTRACT OF THE THESIS

DEEPSPECS

Expert-Level Questions Answering in 5G

by

Aman Ganapathy Manvattira

Master of Science in Computer Science

University of California, Los Angeles, 2025

Professor Songwu Lu, Chair

5G technology enables mobile Internet access for billions of users. Answering expert-level questions about 5G specifications requires navigating thousands of pages of cross-referenced standards that evolve across releases. Existing retrieval-augmented generation (RAG) frameworks, including telecom-specific approaches, rely on semantic similarity and cannot reliably resolve cross-references or reason about specification evolution. We present DEEPSPECS, a RAG system enhanced by structural and temporal reasoning via three metadata-rich databases: SpecDB (clause-aligned specification text), ChangeDB (line-level version diffs), and TDocDB (standardization meeting documents). DEEPSPECS explicitly resolves crossreferences by recursively retrieving referenced clauses through metadata lookup, and traces specification evolution by mining changes and linking them to Change Requests that

document design rationale. We curate two 5G QA datasets: 573 expert-annotated real-world questions from practitioner forums and educational resources, and 350 evolution-focused questions derived from approved Change Requests. Across multiple LLM backends, DEEPSPECS outperforms base models and state-of-the-art telecom RAG systems; ablations confirm that explicit cross-reference resolution and evolution-aware retrieval substantially improve answer quality, underscoring the value of modeling the structural and temporal properties of 5G standards.

The thesis of Aman Ganapathy Manvattira is approved.

Lixia Zhang

Nanyun Peng

Songwu Lu, Committee Chair

University of California, Los Angeles

2025

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGEMENTS

The content for Chapter 3, Chapter 5, and the Appendices is drawn from my preprint for the same submitted to arXiv: "Aman Ganapathy Manvattira, Yifei Xu, Ziyue Dang, Songwu Lu. 2025. DeepSpecs: Expert-Level Questions Answering in 5G. *arXiv preprint arXiv:2511.01305*", which was co-authored by Yifei Xu, Songwu Lu, and Ziyue Dang. While the paper itself was co-authored by the other authors, the architecture, design, and code for DEEPSPECS and the experiments were all created entirely by me. Yifei Xu, Songwu Lu, and Ziyue Dang helped with idea refinement.

The datasets used for evaluation were partially populated and annotated by Yifei Xu and Ziyue Dang along with me.

CHAPTER 1

Introduction

5G technology represents a massive shift in the delivery, access, robustness, and usability of mobile broadband services. System architects, field engineers, and vendor developers across the globe have leveraged the new standard to great effect in the design, development, and deployment of their products and services, resulting in an estimated net contribution to global gdp of 2.3 trillion dollars by 2035 (IHS Markit, 2020). Underpinning this success is a sprawling ecosystem of interdependent technical specifications and design documents, with mature processes for proposing and adopting changes. This ecosystem is governed by the 3rd Generation Partnership Project, an alliance of several different telecommunications standard development organizations (3rd Generation Partnership Project (3GPP), 2024a).

5G practitioners often need to consult these technical specifications in order to understand what features to implement and how they should be structured. In this sense, technical specifications are prescriptive and authoritative, and therefore vital to the telecom workflow. Owing to their prescriptive nature, technical specifications do not contain information on why a particular change or feature was implemented: such design considerations are best captured in meeting notes or change requests. Therefore, when searching for the answer to questions like “Why was Feature A introduced” or “What are the parameters that need to be configured for Feature B”, practitioners need to navigate these complex and scattered documents and trace changes across different releases amidst thousands of specifications and change

requests. This process is time-consuming and requires substantial domain expertise (Chen et al., 2022; Rodriguez Y, 2025).

1.1 Problem Statement

Large Language Models (hereafter LLMs), owing to the statistical patterns observed in the large amount of data encountered during pretraining and the large amount of parameters, have demonstrated the ability to perform question answering tasks (Brown et al., 2020). Due to this, they can provide quick answers to questions asked to them. However, LLMs can struggle with providing factual and high quality answers to highly technical questions. This is because of the risk of hallucinations present in LLMs (Huang et al., 2024) and because the data used for pretraining may not be inclusive of all the necessary context to answer the questions.

Attempts to address this have spurred the development of Retrieval Augmented Generation (hereafter RAG) systems, where a retriever fetches relevant context to the question being asked and makes it available to the LLM during the question-answering process. They have shown promise in reducing factual hallucinations (Huang et al., 2024) and therefore in automating technical question answering (Lewis et al., 2020; Guu et al., 2020; Izacard et al., 2023). However, their efficacy in improving the factuality of the generated answers is contingent on the quality of retrieval. If the retrieval does not fetch all the germane context to the question, the quality of the answer may be compromised. Additionally, complex queries that encompass multiple aspects of a feature or domain may also be difficult to parse for retrievers not optimized for them, also hurting the retrieval. (Huang et al., 2024). Consequently, great care must be taken to ensure that the retrieval strategies are designed with reference to the relationship between context chunks and the questions being asked.

There have been efforts made to utilize LLMs in the telecommunications domain for answering questions. These range from fine-tuning LLMs on 3GPP specifications (Maatouk et al., 2024) to employing RAG by incorporating domain specific retrieval (Bornea et al., 2024; Huang et al., 2025). However, these approaches tend to follow conventional RAG strategies, without considering the unique relationships between documents that characterize the 3GPP domain. This exposes them to the same potential issues regarding answer quality that were described above.

Traditional RAG strategies struggle with 3GPP documents because of two core issues. First is the highly referential and hierarchical nature of the technical specifications. Technical specifications are organized into clauses and sub clauses with unique identifiers. Content from one clause often references content from other clauses either within that specification or within another one. (Qualcomm, 2017; 3rd Generation Partnership Project (3GPP), 2024b) This helps ensure consistency and modularity. Consequently, traditional retrieval and chunking strategies may fail to get all the relevant context. Consider a clause numbered X about certain parameter configurations. This clause X may mention that the configurations are in line with a standard applied elsewhere, in clause Y (which may or may not be in the same document). Traditional chunking strategies split context into chunks, and semantic retrieval would miss out on the critically important context from clause Y thus not providing the full picture to the LLM. Therefore, references must first be understood and then resolved in order to address this problem.

The second core issue is with specification evolution. 5G specifications evolve through a system of approved change requests and meetings. New features can be added or removed, and new specifications or releases can represent different standards. The rationales behind new features and architectures are not present in the technical specifications themselves owing to

their prescriptive nature, but rather in several technical documents (TDocs) ranging from change requests to meeting documents of the 3GPP Alliance. A technical specification mainly covers "what is" rather than "why it is". This makes it challenging to trace when and how a feature was introduced, modified, or deprecated without referencing the aforementioned TDocs (Baron and Gupta, 2018; Chen et al., 2022). Beyond just access to these change requests however, there is the issue of linking the specific change to the relevant change request, a non-trivial problem exposed by the fact that there are multiple versions of each technical specification.

To address these issues, we present DEEPSPECS, a novel telecoms domain expert-level questions answering system. Through its structurally and temporally aware retriever, DEEPSPECS brings 3GPP domain insights to the traditional RAG paradigm. It combines its retrieval strategies with three domain-specific, metadata rich logical databases to allow it to perform i) Cross-Reference Resolution and ii) Specification-Evolution Reasoning.

1.2 Contributions

We introduce DEEPSPECS, a novel framework for 5G QA that enables clause-level cross-reference resolution and specification-evolution reasoning, supporting expert-level question answering over technical standards. Fundamentally, DEEPSPECS contributes a novel retrieval mechanism well tailored to 3GPP Domain Question Answering.

We present, to our knowledge, the first expert-annotated 5G QA dataset, comprising 573 question-answer pairs collected from real-world practitioner forums and blog posts.

Through automated and human evaluation, we show that DEEPSPECS outperforms strong LLM baselines combined with state-of-the-art retrieval, and demonstrate the benefit of cross-reference resolution and specification-evolution reasoning.

1.3 Scope

DEEPSPECS introduces a specialized methodology for handling the complex landscape of 3GPP technical specifications. Although the underlying framework is capable of scaling across diverse document types and releases, this thesis focuses on a foundational implementation. Consequently, we have delimited the scope regarding data coverage, structural assumptions, and evaluation metrics as follows:

1.3.1 Coverage of Specifications and TDocs

Our study focuses on English NR (3GPP) specifications, primarily Releases 17–18, and question types skewed toward RAN/PHY procedures. For TDocs, we currently include only CR-type documents. We have not evaluated other TDoc types, earlier or future releases, or multilingual corpora, so our claims should not be over-generalized.

1.3.2 Structural Assumptions About Documents

The method assumes consistent clause numbering, machine-parseable cross-references, and semitemplated Change Requests (CRs). While this holds for most 3GPP documents, exceptions do occur (e.g., atypical numbering, unusual references, or editorial inconsistencies). Such cases can reduce the reliability of linking and reasoning across documents.

1.3.3 Dependence on LLM-Based Processing

Our pipeline makes extensive use of large language models for chunking, extraction, and metadata handling. This introduces potential errors and computational or API calling cost. However, most of the processing is one-time or incremental with new releases, and we found that budget-oriented models are sufficient for these tasks. Still, downstream accuracy ultimately depends on model quality, which may vary with version updates.

1.3.4 Evaluation

(1) Our annotated datasets are limited in size due to the need for careful expert curation and limited number of high-quality data sources. This constraint may affect statistical significance, and performance estimates should be interpreted with caution.

(2) For our evaluation, some evaluation metrics rely on LLM-based judges or scorers. Although we include human validation, LLM evaluators can reward fluency or plausible reasoning over our desired factors, and prompt bias may persist. We view LLM-based evaluation as a useful complement rather than a full substitute for human evaluation.

1.4 Overview

We organize the rest of this thesis in the following manner:

- In Chapter 2, we go over related work. We trace the history of both Retrieval Augmented Generation for QA and telecom-domain specific approaches to Question Answering. Here, we discuss the limitations of the existing approaches.

- In Chapter 3, we discuss our architecture and approach. This includes an overview of our goals, how we constructed and populated the databases, our approach to cross-reference resolution and to specification-evolution reasoning, and how we generate answers.
- In Chapter 4, we go into detail on our datasets. Namely, how we constructed them and what their characteristics are.
- In Chapter 5, we detail our evaluation results. We compare and contrast DEEPSPECS' performance on our microbenchmarks and datasets versus that of the baselines we chose.
- In Chapter 6, we present our conclusions and insights, and indicate directions for future work.

CHAPTER 2

Related Work

2.1 Other RAG Approaches to Advanced QA

As discussed earlier, RAG improves the factuality of LLM generated answers via the retrieval of relevant context. However, improper retrieval due to complex or ambiguous queries or domain-specific considerations can weaken the factuality of an answer. We have already outlined how the self-and-extra referential nature of the 5G 3GPP corpus can confound traditional semantic retrieval.

There have been other attempts to move beyond pure semantic retrieval. GraphRAG (Edge et al., 2024) constructs a knowledge graph out of the corpus using an LLM, and models relationships between nodes. It then partitions the graph and attempts to generate summaries of the different “communities” of nodes. This approach suffers from an inability to recognize “references” as relations, owing to it still relying on semantic or string matching for forming relations. Additionally, the summarization process can obscure the proper, referential relationships between clauses by bringing in other clauses into the summarization process. There are also issues with modeling the relationships between different versioned corpora, from different specifications.

Some works like FinSage (Wang et al., 2025) and Chat3GPP (Huang et al., 2025) have attempted to address the context deficiency by employing multiple retrieval mechanisms together and aggregating and reranking the results. These methods have merit in mitigating the issues

with any one retrieval mechanism (like pure similarity based retrieval), but ultimately still lack the ability to reason across versioned corpora and navigate the referential relationship between clauses in technical specifications. Other works do try to navigate references within their domain like Legal Document RAG (Zhang et al., 2025a) by iteratively improving queries with knowledge of the retrieved context at each stage. However, these approaches are still too reliant on semantic similarity as a metric for resolving references, and do not account for fetching relevant reasoning context via appropriate heuristics.

While temporal RAG frameworks like E2RAG (Zhang et al., 2025b) and ConQRet (Dhole et al., 2025) offer mechanisms for reasoning over evolving text, they fail to account for the structural complexity of telecommunications standards. Consequently, they cannot resolve the version-dependent cross-references or link specific Change Requests to their resulting specification snapshots—the core 'inter-connectedness' challenge identified in our problem statement.

2.2 Attempts at Domain-Specific Telecom QA Systems

Recent advances in LLMs have spurred efforts to democratize access to the dense and complex landscape of LTE/5G standard specifications. These efforts generally fall into two categories: domain-specialized models and domain-specific retrieval.

(1) **Domain-Specialized Models:** One approach is continual pretraining of existing models with telecommunications context. This is the approach followed by Tele-LLMs (Maatouk et al., 2024). Here, the extra context is cleaned and processed and used to fine-tune and continually pretrain models to enhance their QA capabilities. While this approach does

improve the factuality of the answers, it does not allow for the LLM to distinguish between different versions of the same specification nor let it connect specific clauses to relevant referenced clauses or motivational context

(2) **RAG-based Systems:** To address the need for grounded answers, systems like Chat3GPP (Huang et al., 2025), Telco-RAG (Bornea et al., 2024), and Telco-oRAG (Bornea et al., 2025) employ retrieval-augmented generation. These architectures utilize hybrid dense-sparse retrieval, neural routing, and query reformulation to better locate relevant text within 3GPP documents. While these do improve the quality of the answers, they fundamentally still rely on semantic search without addressing the core issues of cross-references and specification evolution.

DeepSpecs distinguishes itself from both approaches by introducing a **multi-stage structure aware architecture,** with clear metadata search based mechanisms to identify and parse the relationships between technical specification clauses and the TDocs that provide information about them. We combine our structural, temporal, and argumentative insights to allow for cross-reference resolution and specification-evolution reasoning.

2.3 Attempts to Integrate Search with LLMs to Improve Relevancy of Queries and Quality of Retrieved Information

There has been recent work on integrating search with LLMs to improve their QA capabilities. One approach has been teaching LLMs to work with search-as-a-tool architectures, either by sophisticated prompting and subsequent iterative reasoning like with ReAct (Yao et al., 2023), or with multi-stage reasoning and search approaches that employ reinforcement learning to come up with better quality queries (Jin et al., 2025).

However, these search based approaches are reliant on high-quality search repositories that are easily parseable and accessible. 3GPP technical specifications are stored in word or pdf documents on sources that do not lend themselves to searching. Moreover, the structure of these documents are highly technical, hierarchical, and self-referential, meaning that search based approaches that are still reliant on dense semantic retrieval will not be efficient at parsing the connections between clauses and specifications, nor sophisticated enough to reason across evolving corpora.

CHAPTER 3

Architecture and Approach

3.1 Goal

Our goal is to build a system that answers expert-level questions over 5G specifications by producing accurate and helpful responses. This requires structural and temporal understanding beyond surface semantics, in particular: (i) cross-reference resolution, which follows spec IDs and clause numbers to integrate information scattered across modular, non-redundant documents; and (ii) specification-evolution reasoning, which traces when a feature was introduced, how it changed across releases, and why. In line with practitioner use, our goal emphasizes not only factual accuracy but also explanation helpfulness, reflecting how experts contextualize and justify their answers.

3.2 Overview

DEEPSPECS answers expert-level questions over 5G specifications by extending a RAG framework with structural and temporal reasoning. Following human expert practice, it supports clause-level cross-reference resolution and specification-evolution reasoning.

DEEPSPECS operates in four stages: (1) Database construction, where specification content and TDocs are indexed into three databases: SpecDB, serving as the primary context provider enriched with metadata for accurate reference retrieval; ChangeDB and TDocDB, which capture the temporal evolution of specifications and support reasoning over changes and design decisions; (2) Cross-reference resolution, which extracts specification and clause IDs

using rule-based patterns and retrieves the corresponding clauses from filtered chunks with metadata; (3) Specification-evolution reasoning, which maps the queried feature to relevant entries in ChangeDB and uses the associated metadata to retrieve design-phase reasoning from TDocDB; and (4) Answer generation, which produces the final response. Each component is described in detail below.

3.3 Database Construction

As illustrated in Figure 1, we construct three structured databases that serve as the core retrieval sources for DEEPSPECS.

3.3.1 SpecDB

This database stores the core content of 3GPP specification documents. We download official DOCX-formatted specifications from the 3GPP official archive[1] . We chunk documents by atomic clauses (e.g., “7.4.1.1.2 Mapping to physical resource”) and annotate said chunks with metadata including specification ID, clause ID, version number, and timestamp, which are extracted based on structural patterns followed consistently across 3GPP specs (OpenAirInterface, 2022). This clause-aligned indexing enables fine-grained retrieval and also serves as the basis for constructing ChangeDB. We employ an LLM for metadata extraction.

3.3.2 ChangeDB

This database records line-level differences between adjacent versions of each specification clause. Using the clause-aligned chunks derived from SpecDB, we sort all versions of a given clause by timestamp and skip missing versions for robustness. We then compute diffs between

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

each pair of adjacent versions and extract added, removed, or modified lines. Each change entry is annotated with metadata from both versions, including timestamps, and is indexed for temporal reasoning. We also treat the first observed version of a clause as its initial addition, allowing us to track the introduction of new features.

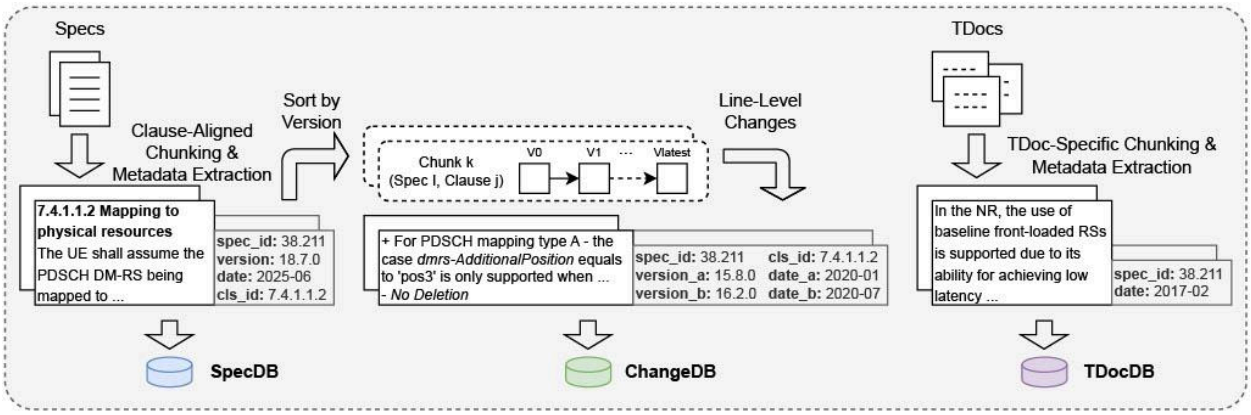

Figure 1: Database construction of DEEPSPECS. The system builds three vector databases from 5G specs and TDocs, each embedded with rich metadata: SpecDB provides the backbone context; ChangeDB tracks the temporal evolution of specs; TDocDB stores the reasoning behind each change.

3.3.3 TDocDB

This database stores 3GPP meeting documents (TDocs), which capture design-phase discussions explaining feature additions, protocol tradeoffs, and specification ambiguities. Although such content is essential for understanding the rationale behind changes, it is omitted from the formal specifications for reasons of brevity and standardization (ShareTechnote; Baron and Gupta, 2018). In this work, we focus on Change Requests (CRs), which document the reasoning behind each proposed modification. We collect DOCX-formatted CRs from the official archive.

These CRs follow a consistent structure. To process them, we employ an LLM extractor for CR-specific chunking and metadata extraction. In our parsing strategy for change requests

(CRs), we extract content from three key sections: the summary of changes, the reasons for the change, and the consequences of non-approval. Each individual change listed in the summary is segmented into a separate chunk, within which it is then paired with its corresponding rationale and the potential consequences if the change is not adopted. All three databases are indexed for both exact metadata filtering and dense semantic retrieval using OpenAI's text-embedding-3-large embeddings (OpenAI, 2024b). We use these indices for hybrid retrieval during the reference resolution and evolution reasoning stages.

3.4 Cross-Reference Resolution

5G specifications are intentionally modular. To reduce redundancy and maintain formality, clauses frequently cite other clauses within the same document or across different documents. For instance, a clause in TS 38.211 may define a feature but defer key operational conditions to TS 38.214. As a result, answering a single technical query often requires chaining together multiple cross-referred clauses (OpenAirInterface, 2022).

To support this, DEEPSPECS implements clauselevel cross-reference resolution, as shown in Figure 2. The resolution process consists of the following steps:

(1) Initial Retrieval:

Given a user query, we first retrieve top-k_1 ranked passages from SpecDB according to semantic similarity. Specifically, we employ HyDE (Gao et al., 2023) in this phase since practical user queries in the 5G domain often lack sufficient context for effective zero-shot retrieval.

(2) Reference Extraction:

From the initially retrieved passages, we extract all cited specification and clause ID pairs . This is done using regular expressions based on citation patterns commonly found in 3GPP documents, such as “See clause 5.1.6.4 of TS 38.214.”

(3) Reference Retrieval: Each pair extracted is used to locate the corresponding clause in SpecDB by matching its spec ID and clause ID through metadata lookup. If the retrieved chunk contains additional references, the system recursively resolves them to build a deeper context chain. To avoid over-expansion, we use a configurable maximum recursion depth. This process enables the system to construct a retrieval trace that reflects the true dependency structure of the initially retrieved passage.

(4) Reference Ranking: All referenced chunks, including those obtained recursively, are re-ranked according to their semantic relevance to the original question. The top-k_2 chunks are returned for final answer generation.

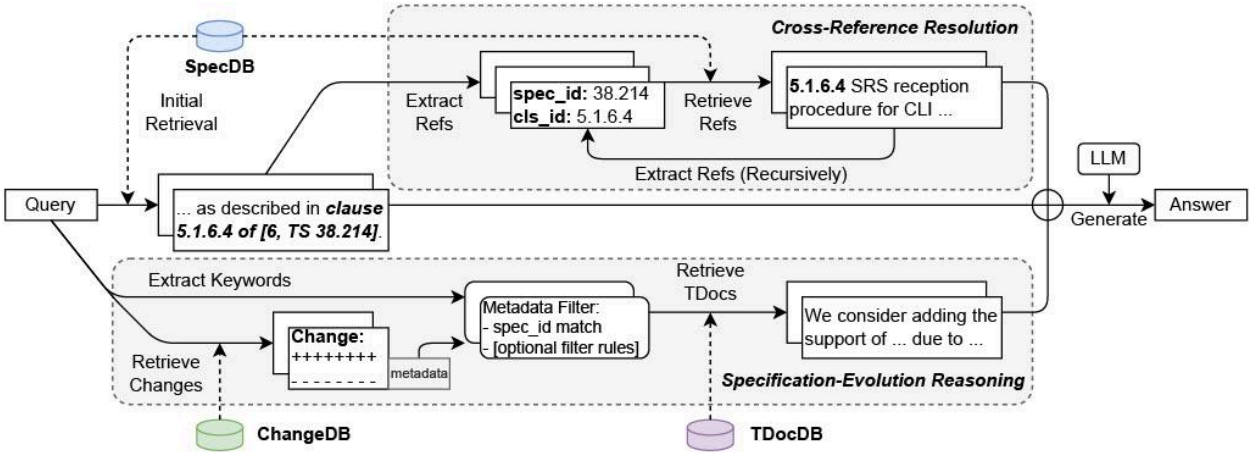

Figure 2: The retrieval process of DEEPSPECS. The system leverages the metadata associated with the chunks in each DB to resolve cross-references and trace changes of specs. This allows DEEPSPECS to provide informative context with structural and temporal understanding of 5G specs for question answering.

3.5 Specification-Evolution Reasoning

Many expert-level queries require understanding how a feature has changed across specification versions and why certain changes were introduced. While 3GPP specifications describe the current intended behavior with precision, they omit the rationale behind changes for conciseness and neutrality (Baron and Gupta, 2018). This rationale is often recorded instead in 3GPP meeting documents (TDocs), which capture discussions, debates, and design motivations during the standardization process.

As illustrated in Figure 2, DEEPSPECS leverages the TDocs and traces the evolution of a feature with the following steps:

(1) Direct Extraction: DEEPSPECS first attempts to extract explicit mentions of specification IDs from the user query (practically, users often name a spec directly). The extracted spec IDs feed a later metadata-based filter. We employ an LLM extractor for this step.

(2) Change Retrieval: If no specification is mentioned explicitly, DEEPSPECS queries CHANGEDB using dense semantic retrieval (along with HyDE) to locate the most relevant change entry. This step discovers additions, removals, or modifications related to the queried feature.

(3) Metadata-Based Filtering: Using the metadata (e.g., date, spec ID) of the identified change, DEEPSPECS maps the query or retrieved change (if any) to a narrowed set of candidate TDocs. While our current implementation matches by spec ID only, which is sufficiently accurate for CRs, the DEEPSPECS framework also supports filtering with additional metadata (e.g., working group, feature tags) and rules when other TDoc types are included.

(4) TDoc Ranking: All filtered TDocs from TDOCDB are re-ranked according to their semantic relevance to the HyDE aligned hypothetical document generated from the original query. The top-k_3 chunks are returned for final answer generation.

This hybrid strategy, supported by ChangeDB, not only grounds the query-to-TDoc mapping in explicit specification metadata but also enhances the interpretability and transparency of retrieval compared to purely semantic methods, which is especially valuable for practitioners.

3.6 Answer Generation

After retrieving relevant specification clauses, resolving cross-references, and identifying supporting TDocs, DEEPSPECS assembles a generation prompt for a general-purpose language model to synthesize the final answer. From the stage of initial retrieval, reference resolution, and specification-evolution reasoning, it selects the top-k_1, k_2, and k_3 chunks, respectively, ranked by semantic similarity.

CHAPTER 4

Datasets

There are very few telecommunications domain QA datasets that have been published in recent years. Of the few that exist, many restrict evaluation to multiple-choice answers. Additionally, the questions in the dataset are artificially created from small context chunks and tend to be surface level queries, thus not being aligned with the kinds of questions that practitioners face in the real-world (Maatouk et al., 2024; Huang et al., 2025; Bornea et al., 2024, 2025; Nikbakht et al., 2024).

Consequently, there is a need for telecommunications question-answer datasets that are i) better aligned with practitioner experience and expectations and ii) reflective of the reasoning behind specification implementation choices. To that end, we construct two new datasets.

4.1 Real-World QA Dataset

In order to ensure that the benchmark is aligned with the kinds of questions that 5G practitioners face in the real world, we compiled a dataset derived from real practitioner inquiries. Sourced from publicly available telecommunications forums and educational resources where practitioners exchange information,, this corpus captures the authentic information-seeking behaviors of 5G professionals, ensuring our evaluation aligns with actual engineering and instructional requirements.

4.1.1 Data Collection

Data collection spanned a three-year window (August 2022–August 2025), targeting two authoritative sources. We mined *telecomHall* (telecomHall), a global practitioner community

established in 1999, specifically isolating question–answer exchanges regarding 3GPP standards that met strict criteria for technical accuracy and structural completeness. Complementing this, we utilized ShareTechnote (Ryu), a comprehensive educational archive focused on 5G documentation. From 236 constituent webpages, we formulated precise technical queries and mapped them to their corresponding verified answers. The resulting dataset is stratified across five technical domains (see Appendix B for the full taxonomy)

4.1.2 Human Annotation

Three volunteers with advanced backgrounds in mobile networks and cellular systems reviewed each QA pair to validate technical correctness, clarity, and relevance. The curation, data collection, and verification process took approximately 120 hours of annotation effort across a 3-month period.

4.2 CR-Focused QA Dataset

To support targeted evaluation of DEEPSPECS specification-evolution reasoning, we construct another QA dataset derived from real 3GPP specification changes with rich supervision signals.

4.2.1 Data Collection

We aggregated a corpus of 997 approved Change Requests (CRs) from 3GPP Releases 17 and 18. Thematically, the collection focuses predominantly on physical layer procedures, channel multiplexing, and dual connectivity enhancements. Crucially, each CR was harvested alongside its complete Technical Specification Group (TSG) documentation package, preserving the associated change rationale, technical implementation details, and impact analysis.

4.2.2 QA Generation

To guarantee dataset depth, we implemented a rigorous pre-processing strategy before generation. We extracted key narrative fields—specifically the change summary, rationale, and impact analysis—and filtered out low-complexity entries (typically editorial corrections) defined by a word count below 200 per field. For the remaining substantive CRs, we used a prompt-engineered LLM (see Appendix A.3) to synthesize question-answer pairs. All generated QA pairs are manually reviewed to verify technical accuracy, question practicality, and answer quality.

4.3 Dataset Characteristics

Table 1 summarizes statistics of the final collected real-world and CR-focused QA datasets.

Source	#	Avg. Q Len	Avg. A Len
Real-world QA Dataset			
telecomHall	102	46.9 tokens	126.9 tokens
ShareTechnote	471	30.3 tokens	102.7 tokens
Total	573	33.2 tokens	107.0 tokens
CR-focused QA Dataset			
3GPP Change Requests	350	45.4 tokens	287.7 tokens

Table 1: Statistics of the QA datasets, including real-world sources and CR-based pairs.

CHAPTER 5

Evaluation Results

We evaluate DEEPSPECS on its ability to answer expert-level questions about 5G specifications. Specifically, we ask:

1. How well does DEEPSPECS answer practical questions encountered by practitioners, compared to the state-of-the-art baselines?
2. How does semantic retrieval differ from crossreference resolution performed by experts?
3. How well does DEEPSPECS perform specification-evolution reasoning on tasks targeting evolution-related reasoning?

5.1 Experimental Setup

5.1.1 Retrieval Databases

For SPECDB, we collect all Release 17 and 18 3GPP technical specifications, in accordance with prior work (Huang et al., 2025). In total, we obtain 2137 specifications as input for database construction. We download Change Requests from 3GPP Releases 17 and 18 that satisfy "approved" status requirements, yielding 997 TDocs in total.

5.1.2 Baselines

We compare DEEPSPECS against two categories of baselines: (1) Base Model: Direct prompting of the base LLMs without any retrieval context. For consistency, we use the same answer-generation prompt, but replace the context section with “NO CONTEXT.” (2) Chat3GPP: A state-of-the-art RAG system tailored for telecom-domain question answering

(Huang et al., 2025) that employs hybrid retrieval, specialized chunking, and efficient indexing, and demonstrates superior performance over existing methods. Nevertheless, Chat3GPP still largely follows the vanilla RAG paradigm. We use it as our primary baseline to demonstrate the limitations of standard RAG approaches on expert-level 5G questions.

For both baselines, we test a collection of generation backends, including GPT4o/4.1/4.1 mini (OpenAI, 2024a, 2025), Qwen3-4/8/14/32B (Yang et al., 2025), and Claude 3.5 Haiku (Anthropic, 2024), all using their default or recommended hyperparameter settings.

5.1.3 Evaluation Metrics

Pairwise Win Rate: We employ an LLM evaluator validated with human evaluation to compute head-to-head win rates between system outputs. The evaluator is given the question, a gold answer, and two candidate answers, and is instructed to select the preferred one based on accuracy and helpfulness. The evaluation prompt is provided in Appendix A.1. To mitigate positional bias, evaluation is repeated with reversed answer order. We validate the alignment with human evaluators by asking domain experts to re-judge a subset of 144 pairs. The LLM-based judgments are aligned with human assessments in 92.4% of cases, with most disagreements occurring in close comparisons.

Rubric-Based Scoring: To complement pairwise win rates with absolute quality scores, we design question-specific rubrics. For each QA pair, rubrics are first generated with an LLM, then curated and validated by domain experts. During evaluation, an LLM applies these rubrics to score each candidate answer along multiple dimensions. The rubric design and evaluation prompts are provided in Appendix A.2.

We provide more implementation details for our experiments in Appendix D.

Primarily in this paper, we use pairwise win rate as the main metric. Pairwise win rates have been shown to be more reliable (Liusie et al., 2024) and to align better with human judgments (Liu et al., 2024). Our additional results for rubric-based scoring can be found in Appendix D.2 and E. These results are consistent with those obtained from pairwise comparison.

5.2 Results on Real-World QA

Table 2 summarizes results across multiple model families. For DEEPSPECS, we fix retrieval parameters (k_1=4, k_2=3, k_3=3), and have Chat3GPP return the top-10 chunks to ensure that both DEEPSPECS and Chat3GPP return the same amount of context. We report both win rates and output lengths to ensure that observed improvements are not confounded by verbosity bias.

Retrieval consistently helps, with DEEPSPECS providing the strongest gains. Both Chat3GPP and DEEPSPECS improve over base models, underscoring the importance of retrieval augmentation for navigating 3GPP specifications. While Chat3GPP offers decent improvements, it remains limited by its inability to resolve cross-references or track evolutions of specifications.

In contrast, DEEPSPECS leverages structure and evolution aware retrieval to deliver consistent gains across generator models and question categories, showing robustness regardless of model size or output length. Rubric-based evaluation provides further confirmation of these findings. With GPT4.1, DEEPSPECS achieves a mean score improvement of 0.62 points on a 5-point scale ($p<0.0001$, Wilcoxon signed-rank test, $dz=0.41$), while against GPT-4.1-mini, the improvement is 0.60 points ($p<0.0001$, $dz=0.40$). Detailed results on per category win rates and rubric-based scoring are provided in Appendix B.

Gains are larger on certain weaker models, but remain meaningful for stronger ones. The results indicate that even high-capacity models benefit from explicit grounding in structural and temporal aspects of the specifications. Given the rapid evolution of 3GPP standards, such grounding becomes especially valuable for maintaining accurate and helpful QA throughout.

Model	Method	Length (tokens)	Win Rate vs. GPT-4o (%)
GPT-4o	Base Model	295	(50)
	Chat3GPP	295	62.5
	DEEPSPECS	301	**68.1**
Qwen3-4B	Base Model	307	34.4
	Chat3GPP	427	67.3
	DEEPSPECS	415	**71.2**
Qwen3-8B	Base Model	346	50.8
	Chat3GPP	490	74.4
	DEEPSPECS	483	**79.8**
Qwen3-14B	Base Model	331	64.7
	Chat3GPP	459	78.9
	DEEPSPECS	460	**83.1**
Qwen3-32B	Base Model	373	73.6
	Chat3GPP	499	82.9
	DEEPSPECS	497	**88.0**
Claude 3.5 Haiku	Base Model	268	48.6
	Chat3GPP	278	56.6
	DEEPSPECS	278	**59.4**
GPT-4.1 mini	Base Model	395	82.0
	Chat3GPP	424	87.8
	DEEPSPECS	442	**90.1**
GPT-4.1	Base Model	427	87.2
	Chat3GPP	494	93.7
	DEEPSPECS	511	**95.4**

Table 2: Comparative performance on the real-world QA dataset. Both Chat3GPP and DEEPSPECS return the top-10 chunks. For Base GPT-4o, the win rate is fixed at 50%.

5.3 Evaluating Cross-Reference Resolution

We quantify the gap between semantic retrieval and cross-reference resolution using a microbenchmark over 10 specifications (TS 38.181–TS 38.304 from release 18). In total, we sample 547 chunks and extract 903 cross-references. These are filtered with an LLM helpfulness judge (Appendix A.4) and validated by experts to form a set of helpful references, on which we calculate the precision, recall, and f1 score.

We compare the retrievals of (a) Chat3GPP and (b) DEEPSPECS's standalone cross-reference resolver to this set of helpful references. To ensure a fair comparison, we fix retrieval parameters (k_1=3, k_2=2, k_3=0) for DEEPSPECS and have Chat3GPP return the top-5 chunks. Table 3 shows substantially higher precision, recall and F1 for DEEPSPECS, indicating the limitations of semantic-only retrieval on interlinked specs and the effectiveness of DEEPSPECS's cross-reference resolution method.

Method	Precision	Recall	F1
Chat3GPP	0.0709	0.2334	0.1031
DEEPSPECS	**0.1876**	**0.7055**	**0.2837**

Table 3: Cross-reference resolution microbenchmark performance. Precision, recall, and f1 are calculated on the filtered set of "helpful" references.

5.4 Evaluating Spec-Evolution Reasoning

We isolate the contribution of evolution reasoning using a CR-focused QA dataset. To control for confounding variables, we disable reference resolution and fix retrieval parameters (k_1=5, k_2=0,

k_3=5), ensuring DEEPSPECS and Chat3GPP retrieve the same number of chunks. Table 4 reports GPT-4o results.

Relative to both the base model and Chat3GPP, DEEPSPECS (with reference resolution disabled) still achieves significantly higher win rates while producing outputs of comparable length. Results for additional models are provided in Appendix C and show the same trend. Rubric-based evaluation reinforces these findings. On this CR-focused subset, DEEPSPECS achieves a mean quality score of 3.05, compared to 2.48 for Chat3GPP and 2.26 for the base model (all differences significant at $p<0.0001$, detailed scores in Appendix C).

Model	Method	Length (tokens)	Win Rate vs. GPT-4o (%)
GPT-4o	Base Model	273	(50)
	Chat3GPP	265	59.3
	DEEPSPECS	269	**77.0**

Table 4: Comparative Performance on the CR-Focused QA Dataset. Even with reference resolution disabled, DEEPSPECS outperforms the base model and Chat3GPP at comparable output lengths.

CHAPTER 6

Conclusion, Insights, and Future Directions

We introduce DEEPSPECS, a system designed to improve performance on telecommunications QA by leveraging structural and temporal insights from 3GPP standards. Through explicit clause-level cross-reference isolation and resolution and specification-evolution reasoning, DEEPSPECS addresses the deficiencies of conventional RAG approaches. This architectural approach is validated by our evaluation results on a real-world QA dataset and specific microbenchmarks, which demonstrate consistent gains over baselines.

6.1 Potential Risks

Although our system produces fluent, citation-grounded responses, retrieval gaps may still result in incomplete or potentially misleading outputs. A significant risk lies in automation bias, where users might forgo manual verification against the official specifications, potentially leading to non-compliant implementations. As our training data consists exclusively of public 3GPP standards, concerns regarding privacy and data security are minimal. Nevertheless, we strongly advocate for deployment safeguards, including strict citation enforcement, uncertainty quantification, and role-based access controls.

6.2 Future Work

Building on our work, some future directions to take telecom questions answering are:

- Expanding the TDoc chunking mechanisms to include other kinds of TDocs beyond just CRs

- Testing the effectiveness of coupling other retrieval strategies outside of just semantic retrieval with our metadata search
- Expanding our fundamental strategy to other domains outside of 3GPP like ORAN

APPENDIX A

Prompt Templates

A.1 Pairwise Win Rate

We use the following prompt for the pairwise evaluator:

```
# Task
You are an expert in the 5G domain. You will be provided with a 5G-related question, an expert-written reference answer, and two candidate answers to evaluate. Your task is to determine which candidate answer is better based on the reference answer and the following criteria:
- **Answer Accuracy**: How factually correct is each candidate answer compared to the reference answer? Are there any inaccuracies, omissions,or misleading statements?
- **Explanation Helpfulness**: How well does each candidate answer explain the reasoning behind itself? Does it provide useful context, background information, or insights that enhance understanding of the answer?


# Question
{question}


# Reference Answer
{ground_truth}
```

Candidate Answers

- Answer A: {answer_a}

- Answer B: {answer_b}

Instructions

Provide brief reasoning based on the criteria above, followed by your final decision in the format below:

```
Reasoning: <Your brief reasoning>

Judgment: <"Answer A" or "Answer B">
```

A.2 Rubric-based scoring

We use the following prompt to generate the initial question-specific for each QA pairs:

You are an expert in 5G telecommunications and assessment design. Create a detailed scoring rubric for the following question based on the ground truth answer provided.

Question: {question}

Ground Truth Answer: {ground_truth}

Create a scoring rubric with a total of 5 points. Break down the points based on:

1. Core answer/concept (typically 2-3 points)

2. Key details/explanation (typically 1-2

points)

3. Technical accuracy (typically 0.5-1 point)

4. Completeness (typically 0.5-1 point)

Return ONLY a JSON object with this structure:

{

"total_points": 5,

"criteria": [

{"description": "criterion description",

"points": X},

{"description": "criterion description",

"points": Y}

]

}

Make the rubric specific to this question. Do not include any other text outside the JSON.

After human curation and validation, we use the following prompt to score each answer with the question-specific rubrics:

You are an expert grader for 5G telecommunications questions. Grade the predicted answer

against the ground truth using the provided rubric.

Question: {question}

Ground Truth Answer: {ground_truth}

Predicted Answer: {predicted_answer}

Scoring Rubric (Total: {total_points} points):

{criteria_text}

Carefully evaluate the predicted answer against each criterion in the rubric. Award points based on:

- Factual correctness compared to ground truth
- Completeness of the answer
- Technical accuracy of terminology
- Whether key points from ground truth are covered

Return ONLY a JSON object with this exact structure:

{

"score": X.X,

"grading_reasoning": "Detailed explanation of the score, breaking down points awarded/deducted for each criterion"

}

The score should be a number between 0 and {total_points}, and can include decimals (e.g.,

3.5, 4.0).

Be strict but fair. If the predicted answer contradicts the ground truth or misses key points, deduct points accordingly.

Do not include any text outside the JSON object.

A.3 CR QA Generation Prompt

We use the following prompt to generate question answer pairs from approved Change Request documents. The prompt instructs the language model to create questions in one of two formats depending on the clarity of before-and-after states in the specification text, and to skip CRs that lack sufficient technical detail for meaningful QA generation.

You are analyzing a 3GPP Change Request (CR) document. Based on the information provided, generate ONE question-answer pair.

CR Information:

- Specification: {spec_number}
- Summary of Change: {summary}
- Reason for Change: {reason}
- Consequences if Not Approved: {consequence}
- Clauses Affected: {clauses_affected}

Decision rule BEFORE producing output:

- If AND ONLY IF the Summary of Change, Reason for Change, AND Consequences are all very short one-liners (i.e., each is so brief that you cannot identify a meaningful technical change and rationale), then DO NOT create a Q&A. Instead, return: { "skip": true, "reason": "why the three fields are too simple to form a meaningful Q&A" } Otherwise, create exactly ONE Q&A with the following instructions.

Instructions for Q&A:

1) Decide if you can identify clear "before" and "after" states in the spec text from the provided info.

2) Generate a question in ONE of these formats:

Format A (if before AND after states are clear; do NOT mention the spec number in the question):

- Example templates:
  * "Why was [specific parameter/behavior X] changed to [specific parameter/behavior Y]?"
  * "What is the reason for changing [feature/requirement X] to [feature/requirement Y]?"
  * "I observed that [X] was modified to [Y]. What is the rationale for this change?"

Format B (if before OR after is unclear; explicitly mention the specification number):

- Example templates:

    * "What is the reason for the change to [specific aspect] in specification {spec_number}?"

    * "Why was [feature/behavior] modified in 3GPP TS {spec_number}?"

    * "I observed a change regarding [topic] in specification {spec_number}. What is the reason?"

3) The answer should:

- Explain the reason for the change, - Include the consequences if the change is not accepted,

- Be natural and technical (no copy-paste; use proper terminology).

Output format (strict JSON):

- If skipping: { "skip": true, "reason": "short explanation" }

- If generating Q&A: { "question": "your generated question", "answer": "your generated answer", "format_used": "A" or "B" }

All generated question-answer pairs undergo manual review to verify technical accuracy, ensure natural phrasing, and confirm that answers adequately explain both the rationale for the specification change and the potential consequences if the change were not approved.

A.4 Reference Helpfulness Filter

We use the following prompt to filter the helpful references for the microbenchmark on cross-reference resolution:

Compare the original chunk to the reference chunk.

original chunk: {org_chunk}

reference chunk: {ref_chunk}

Instructions:

1. See if the reference chunk provides information that would be useful to understand what the original chunk is talking about.

2. Return 'helpful' if the reference chunk's content will provide necessary detail to fully understand what the original chunk's content is talking about.

3. Return 'unhelpful' if the reference chunk's content is not necessary to fully understand what the original chunk is talking about. 4. Respond only with 'helpful' or 'unhelpful'

APPENDIX B

Additional Results on Real-World QA

B.1 Dataset Categorization

The 573 questions in our real-world QA dataset are organized into five technical categories that reflect the organizational structure of 5G standardization and engineering practice:

- **Category 1: Air Interface & Physical Layer** (186 questions, 32.5%) – Physical channel structures, modulation schemes, MIMO configurations, beamforming, and radio resource allocation in the NR air interface.
- **Category 2: Protocol Stack & Resource Management** (174 questions, 30.4%) – Layer 2/3 protocol operations (MAC, RLC, PDCP, SDAP), quality-of-service mechanisms, numerology, and resource mapping procedures.
- **Category 3: Core Network & Session Management** (114 questions, 19.9%) – 5G Core architecture, session establishment and mobility procedures, network function interactions, and signaling flows.
- **Category 4: Network Architecture & Deployment** (66 questions, 11.5%) – Network slicing, multi-access architectures, deployment scenarios (NSA/SA), interworking with legacy systems, and infrastructure planning.

- **Category 5: Operations, Optimization & Troubleshooting** (33 questions, 5.8%) – Performance optimization strategies, diagnostic procedures, measurement reporting, and operational best practices.

The distribution reflects the relative emphasis in 5G technical documentation and practitioner discussions, with lower-layer protocol mechanics and physical-layer operations receiving the most attention. Each question is assigned to exactly one category by the annotation team based on its primary technical focus.

B.2 Additional Evaluation

Table 5 presents additional rubrics-scoring results. Table 6 presents the per-category win rates against the GPT-4o baseline across five 5G technical domains, demonstrating the performance of base models, Chat3GPP, and DEEPSPECS for different model families and sizes.

Metric	GPT-4.1	GPT-4.1-mini
Baseline Mean	2.67	2.46
DEEPSPECS Mean	**3.29**	**3.06**
Improvement	+0.62***	+0.60***
Effect Size (dz)	0.41	0.40

Table 5: Rubric-based scoring results comparing DEEPSPECS against base models. Scores range from 0 to 5. All differences significant at $p<0.0001$ (Wilcoxon signed-rank test).

Model	Method	Air Interface	Protocol Stack	Core Network	Network Arch.	Operations & Troubleshooting	Overall
GPT-4o	Base Model	(50)	(50)	(50)	(50)	(50)	(50)
	Chat3GPP	54.8	67.8	64.5	62.1	69.7	62.4
	DEEPSPECS	**60.5**	**72.1**	**72.4**	**66.7**	**77.3**	**68.1**
Qwen3-4B	Base Model	33.6	30.7	38.6	39.4	33.3	34.4
	Chat3GPP	59.9	70.4	71.9	72.0	66.7	67.3
	DEEPSPECS	**62.9**	**76.1**	**74.6**	**75.0**	**72.7**	**71.2**
Qwen3-8B	Base Model	47.6	50.6	50.4	56.1	60.6	50.8
	Chat3GPP	72.0	78.4	71.9	74.2	75.8	74.4
	DEEPSPECS	**73.4**	**83.6**	**80.3**	**86.4**	**80.3**	**79.8**
Qwen3-14B	Base Model	56.5	61.5	65.4	65.2	63.6	61.2
	Chat3GPP	73.9	**85.9**	79.8	75.8	77.3	79.1
	DEEPSPECS	**75.8**	85.6	**82.0**	**87.9**	**81.8**	**81.8**
Qwen3-32B	Base Model	67.7	77.0	71.9	78.0	86.4	73.6
	Chat3GPP	79.6	87.4	84.2	78.0	83.3	82.9
	DEEPSPECS	**82.3**	**92.5**	**86.8**	**92.4**	**92.4**	**88.0**
Claude 3.5 Haiku	Base Model	47.8	50.9	53.1	41.7	39.4	48.6
	Chat3GPP	51.6	62.4	57.5	53.0	59.1	56.6
	DEEPSPECS	**53.0**	**64.4**	**61.8**	**58.3**	**63.6**	**59.4**
GPT-4.1 mini	Base Model	86.3	81.3	81.1	75.0	78.8	82.0
	Chat3GPP	84.4	89.1	**89.9**	88.6	90.9	87.8
	DEEPSPECS	**87.6**	**91.4**	89.5	**92.4**	**93.9**	**90.1**
GPT-4.1	Base Model	89.2	87.6	83.8	84.8	89.4	87.2
	Chat3GPP	92.5	93.4	94.3	**97.7**	92.4	93.7
	DEEPSPECS	**93.8**	**95.4**	**96.1**	**97.7**	**97.0**	**95.4**

Table 6: Per-category performance on the real-world QA dataset (Win Rate vs. GPT-4o, %)

APPENDIX C

Additional Results on CR Focused QA

Table 7 presents additional pairwise win rate results. Table 8 presents additional rubric-based scoring results.

Model	Method	Length (tokens)	Win Rate vs. GPT-4o (%)
Qwen3-4B	Base Model	283	68.6
	Chat3GPP	372	81.4
	DEEPSPECS	341	**88.0**
Qwen3-8B	Base Model	312	73.3
	Chat3GPP	466	85.7
	DEEPSPECS	427	**92.6**
Qwen3-14B	Base Model	295	81.1
	Chat3GPP	481	91.4
	DEEPSPECS	443	**93.6**
Qwen3-32B	Base Model	353	89.9
	Chat3GPP	549	93.7
	DEEPSPECS	506	**95.7**

Table 7: Additional results on the CR-focused QA dataset.

Metric	Score
Base Model Mean	2.26
Chat3GPP Mean	2.48
DEEPSPECS Mean	**3.05**
DEEPSPECS vs. Base	+0.79 (0.68) ***
DEEPSPECS vs. Chat3GPP	+0.58 (0.51) ***
Chat3GPP vs. Base	+0.21 (0.26) ***

Effect sizes (dz) shown in parentheses.

Table 8: Rubric-based scoring results on CR-focused QA using GPT-4o as the base model. Scores range from 0 to 5. All pairwise differences significant at $p<0.0001$ (Wilcoxon signed-rank test).

APPENDIX D

Implementation Details

D.1 Inference Hyperparameters

We adopt default or recommended settings for inference across APIs and locally deployed models:

- OpenAI models (GPT-4o, GPT-4.1): temperature = 1.0, top_p = 1.0
- Claude 3.5 Haiku: temperature = 1.0, top_p = 1.0
- Qwen3 models: temperature = 0.6, top_p = 0.95, top_k = 20, with thinking mode enabled

D.2 Computational Infrastructure

Hardware and API Access. Experiments with GPT and Claude models were conducted via official APIs. For Qwen3 models, inference was performed locally using vLLM on a single NVIDIA A100 80GB GPU.

D.3 Other Experimental Settings

Unless otherwise specified:

- DEEPSPECS cross-reference resolution, the recursion depth is fixed at 2.
- All embeddings are obtained using text-embedding-3-large.
- GPT-4o serves as the default assistant, extractor, generator, and evaluator model.

REFERENCES

IHS Markit. 2020. The 5G Economy in a Post-COVID-19 Era. White Paper. Qualcomm Technologies, Inc.

3rd Generation Partnership Project (3GPP). 2024a. 5g system overview. Accessed: 2025-07-28.

3rd Generation Partnership Project (3GPP). 2024b. Specifications and technologies. Accessed: 2025- 07-28.

Anthropic. 2024. Claude 3.5 haiku. https://www. anthropic.com/claude/haiku. Accessed: 2025-10-06.

Justus Baron and Kirti Gupta. 2018. Unpacking 3gpp standards. *Journal of Economics & Management Strategy*, 27(3):433–461.

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. In *GLOBECOM 2024-2024 IEEE Global Communications Conference*, pages 2359– 2364. IEEE

Andrei-Laurentiu Bornea, Fadhel Ayed, Antonio De Domenico, Nicola Piovesan, Tareq Si Salem, and Ali Maatouk. 2025. Telco-orag: Optimizing retrieval-augmented generation for telecom queries via hybrid retrieval and neural routing. *arXiv preprint arXiv:2505.11856.*

Yi Chen, Di Tang, Yepeng Yao, Mingming Zha, XiaoFeng Wang, Xiaozhong Liu, Haixu Tang, and Dongfang Zhao. 2022. Seeing the forest for the trees: Understanding security hazards in the

{3GPP} ecosystem through intelligent analysis on change requests. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 17–34.

Kaustubh Dhole, Kai Shu, and Eugene Agichtein. 2025. ConQRet: A new benchmark for fine-grained automatic evaluation of retrieval augmented computational argumentation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5687–5713, Albuquerque, New Mexico. Association for Computational Linguistics.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2024. From local to global: A graph rag approach to query-focused summarization. *arXiv preprint arXiv:2404.16130.*

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Long Huang, Ming Zhao, Limin Xiao, Xiujun Zhang, and Jungang Hu. 2025. Chat3gpp: An open-source retrieval-augmented generation framework for 3gpp documents. In *2025 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 492–497.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel

Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS '20)*. Curran Associates Inc., Red Hook, NY, USA, Article 159, 1877–1901.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2024. A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions. *ACM Transactions on Information Systems* 1, 1, Article 1 (January 2024), 58 pages. https://doi.org/10.1145/3703155

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane DwivediYu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: Few-shot learning with retrieval augmented language models. *Journal of Machine Learning Research*, 24(251):1–43.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459– 9474.

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*.

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*.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. *arXiv preprint arXiv:2503.09516,* 2025.

OpenAI. 2024a. Gpt-4o system card. https:// openai.com/index/gpt-4o-system-card/. Accessed: 2025-10-06.

OpenAI. 2024b. Openai text-embedding3-large. https://openai.com/index/new-embedding-models-and-api-updates/.

OpenAI. 2025. Gpt-4.1 (and variants: mini, nano). https://openai.com/index/gpt-4-1/. Accessed: 2025-10-06.

OpenAirInterface. 2022. Walkthrough 3gpp specifications. Accessed: 2025-07-28.

Qualcomm. 2017. Understanding 3gpp: Starting with the basics. Accessed: 2025-07-28.

Felipe A Rodriguez Y. 2025. Technical language processing for telecommunications specifications. *Applied AI Letters*, 6(2):e111.

Jaeku Ryu. Sharetechnote. https://www.sharetechnote.com. Accessed: 2025-10-04.

ShareTechnote. 5g frame structure and related discussions: Background context and rationale. Accessed: 2025-07-28.

telecomHall. telecomhall forum. https://www.telecomhall.net. Accessed: 2025-10-04.

Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037.

Xinyu Wang, Jijun Chi, Zhenghan Tai, Tung Sum Thomas Kwok, Muzhi Li, Zhuhong Li, Hailin He, Yuchen Hua, Peng Lu, Suyuchen Wang, and 1 others. 2025. Finsage: A multi-aspect rag system for financial filings question answering. *arXiv preprint arXiv:2504.14493*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Adian Liusie, Potsawee Manakul, and Mark Gales. 2024. LLM Comparative Assessment: Zero-shot NLG Evaluation through Pairwise Comparisons using Large Language Models. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 139–151, St. Julian's, Malta. Association for Computational Linguistics.

Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulic, Anna Korhonen, and Nigel Collier. 2024a. Aligning with human judgement: The role of pairwise preference in large language model evaluators. *CoRR*, abs/2403.16950.

Wutong Zhang, Hefeng Zhou, Qiang Zhou, Yunshen Li, Yuxin Liu, Jiong Lou, Chentao Wu, and Jie Li. 2025a. Towards comprehensive legal document analysis: A multi-round rag approach. In *Proceedings of the 2025 International Conference on Multimedia Retrieval*, pages 1840–1848.

Ze Yu Zhang, Zitao Li, Yaliang Li, Bolin Ding, and Bryan Kian Hsiang Low. 2025b. Respecting temporal-causal consistency: Entity-event knowledge graphs for retrieval-augmented generation. *arXiv preprint arXiv:2506.05939.*