ClauseRouteRAG: A Clause-Level Retrieval-Augmented Generation Framework for 3GPP Telecommunications Standards

Shan Lu[1,2], Xiayi Wang[3], Zhanying Shao[4],
Jing Bai[3], *Senior Member*, IEEE, Zhu Xiao[5], *Senior Member*, IEEE,
and Yasheng Zhang[1,2,*]
[1]The 54th Research Institute of China Electronics Technology Group Corporation, Shijiazhuang, China
[2]National Key Laboratory of Advanced Communication Networks
[3]School of Artificial Intelligence, Xidian University, Xi'an, China
[4]Hebei Construction Material Vocational and Technical College, Qinhuangdao, China
[5]College of Computer Science and Electronic Engineering, Hunan University, Changsha, China

***Abstract*—The 3rd Generation Partnership Project (3GPP) produces technical specifications with long documents, hierarchical clause structures, and many tables, which makes automated question answering difficult. While Retrieval Augmented Generation (RAG) has improved robustness for knowledge intensive tasks, existing systems can still miss the right clauses in long specifications and may retrieve incomplete evidence for standards questions. This paper presents ClauseRouteRAG, a RAG framework for multiple choice question answering over 3GPP specifications. ClauseRouteRAG first selects a small set of relevant clauses by aggregating evidence from an initial retrieval pass. It then retrieves passages within these clauses using dense retrieval and BM25 with rank fusion. Finally, it reranks the candidate passages using overlap of technical identifiers extracted from the question and the retrieved passages, and it prompts the language model to answer using only the provided evidence. Experiments on TeleQnA Release 17 and Release 18 show that ClauseRouteRAG consistently improves over LLM only baselines and a naive RAG pipeline, reaching an overall accuracy of 0.849 on TeleQnA under the default setting.**

***Index Terms*—Telecommunications, Retrieval Augmented Generation, Large Language Model, 3GPP**

I. Introduction

Large language models (LLMs) have recently shown strong performance in natural language understanding and generation, making them attractive for building interactive assistants that answer technical questions [1]. Deep learning methods have also been widely explored in telecommunications-related tasks, including modulation signal representation learning and few-shot signal modulation classification [2], [3]. In telecommunications, many questions must ultimately be resolved against standards documents, where correctness depends not only on fluent explanations but also on precise evidence traceable to authoritative text. The 3rd Generation Partnership Project (3GPP) specifications are essential references in global telecommunications [4], yet their scale, complex structure, and frequent updates make it difficult to locate the exact clauses needed for practical engineering questions. These characteristics have motivated efforts to combine LLMs with retrieval-augmented generation (RAG), grounding responses in relevant specification clauses rather than relying solely on the model's pretraining [5], [6].

Recent work on RAG for 3GPP question answering mainly builds indices over 3GPP releases and retrieves relevant segments as evidence for generation [7]. TelecomRAG [8] and Chat3GPP [9] follow this direction by chunking specifications, building efficient indexes, and retrieving supporting passages so answers can be checked against the original text. These systems improve grounding and make it easier to trace responses to the specifications, but they still struggle on questions that require evidence from multiple clauses. When the needed conditions are scattered and many passages share similar wording, retrieval may return incomplete or distracting context, which can lead to missed constraints or wrong clause attribution.

To reduce retrieval noise and improve evidence coverage in telecom standards QA, researchers have explored structure-aware retrieval. Sousa et al. [10] organize 3GPP-oriented corpora by clustering chunk embeddings with bisecting k-means and use the resulting clusters to constrain the retrieval space. Yuan et al. [11] propose a KG-RAG framework that integrates a telecom knowledge graph with retrieval-augmented generation to retrieve structured, up-to-date domain evidence for standards such as 3GPP and O-RAN. Even so, these methods add extra work to build and update clusters or graphs as the specifications change, and they can still miss the exact clause conditions needed for questions that span several clauses.

Based on these observations, we propose ClauseRouteRAG, a retrieval augmented generation framework for question answering over 3GPP telecommunications standards.

This work was supported in part by the Future Industry Technology Innovation Special Project of the Key Science and Technology Support Program of Hebei Province under Grant 2050201.

*Corresponding author: Yasheng Zhang.

ClauseRouteRAG adds routing at the clause level, so retrieval first focuses on a small set of candidate clauses and avoids distraction from long and repetitive text. Within the selected clauses, it uses dense retrieval together with BM25, which helps when queries contain abbreviations, different phrasing, or terms that appear in tables. This design improves evidence coverage for questions that involve more than one clause and keeps the evidence easy to trace in the generation stage.

Our main contributions are summarized as follows:

- We introduce ClauseRouteRAG, a RAG framework tailored to 3GPP specifications that performs clause-level routing to identify a compact candidate set and reduce retrieval interference in long, similarly worded standards documents.
- We design a hybrid retrieval module with rank-based fusion, along with an entity/anchor-coherence reranking strategy, to improve evidence completeness and clause-level traceability for complex multi-clause queries.
- We evaluate ClauseRouteRAG on open-source 3GPP QA benchmarks and show consistent accuracy improvements over strong LLM-only and RAG baselines.

II. Related Work

Recent advances in LLMs have enabled their application to interpreting complex telecommunications standards, particularly 3GPP specifications [5]. However, general-purpose LLMs may lack domain-specific proficiency and up-to-date knowledge when answering technical queries, motivating research on telecommunications domain adaptation [7]. Existing approaches are commonly grouped into two complementary paradigms [11], [12]. The first adapts model parameters through domain-specific pre-training or fine-tuning to strengthen internal knowledge, while the second relies on retrieval-augmented generation (RAG) to incorporate external evidence at inference time.

In telecom domain adaptation, recent work puts emphasis on clear evaluation and practical deployment settings. TeleQnA [13] provides a benchmark with 10,000 multiple choice questions from standards and research articles, which supports consistent measurement of telecom knowledge.Using such benchmarks, TelecomGPT [14] and Tele-LLMs [15] study standard pipelines for telecom adaptation, including continual pre training and instruction tuning on curated telecom corpora. In parallel, model efficiency has been explored for practical 3GPP use cases by combining fine-tuning with low-bit quantization under hardware constraints [16], highlighting the need to balance accuracy and deployment cost.

RAG for 3GPP specifications has progressed from basic evidence retrieval to optimization and structure-aware modeling. Early frameworks, including Telco-RAG [17], TelecomRAG [8], and Chat3GPP [9], build 3GPP-grounded assistants by combining specialized indexing with retrieval mechanisms such as hybrid lexical-semantic matching. Subsequent methods refine retrieval quality by incorporating text–table handling [18], leveraging long-context modeling [19], [20], and introducing routing mechanisms to narrow the search space [21]. Beyond direct semantic matching, other lines of work organize the embedding space for retrieval [10] or integrate knowledge graphs to model structured relations [11]. More recent systems, such as TelcoAI [22] and DeepSpecs [23], adopt agentic and structure-aware retrieval to better handle diagrams, cross-references, and evolving specifications.

III. Methodology

In this section, we introduce the methodology of ClauseRouteRAG, as shown in Fig. 1. The overall architecture involves several key phases, including data pre-processing, indexing, retrieval, and generation, ensuring both efficiency and accuracy in processing and retrieving information from the 3GPP technical documents.

A. Preprocessing of Document Texts

We preprocess 3GPP specifications into a passage corpus with explicit structure information to support retrieval aligned with clause boundaries. Our knowledge base is built from the TSpec-LLM collection [15] and covers Release 17 and Release 18 specifications. All inputs are converted into a consistent `.docx` format, and temporary or corrupted files are filtered to improve parsing robustness. To reduce retrieval noise, we remove sections that rarely contribute to standards reasoning, such as the table of contents, reference lists, and clearly non normative materials.

Unlike many telecom RAG pipelines that discard tables during extraction, ClauseRouteRAG preserves tabular content as retrievable evidence. In 3GPP specifications, critical parameters and constraint mappings are often stated in tables, and excluding them can remove decisive evidence for standards queries. For each table, we linearize every data row into a compact passage using the header as field names. Given header cells $\{h_j\}_{j=1}^{m}$ and row values $\{v_j\}_{j=1}^{m}$, we construct

$$t_{\text{row}} = \bigoplus_{j=1}^{m} (h_j = v_j)\,, \tag{1}$$

where $\oplus$ denotes string concatenation with separators. Each t_{row} is stored with its source file, recovered clause identifier, title path, and table index and row index.

We further segment the cleaned narrative text into chunks that follow the document headings, and we prepend the full title path to each chunk so that each passage remains self contained and traceable to its source clause. Clause identifiers are recovered from local headings and textual patterns, and when explicit clause markers are missing, we assign the passage to the nearest preceding heading and inherit the closest available ancestor clause identifier. We set the chunk length to 250 tokens, which has shown strong performance for 3GPP question answering in prior telecom RAG studies. [17], [21] All passages, including narrative chunks and table row passages, are stored with metadata fields that include the source file, clause identifier, title path, and a stable passage index for indexing and retrieval.

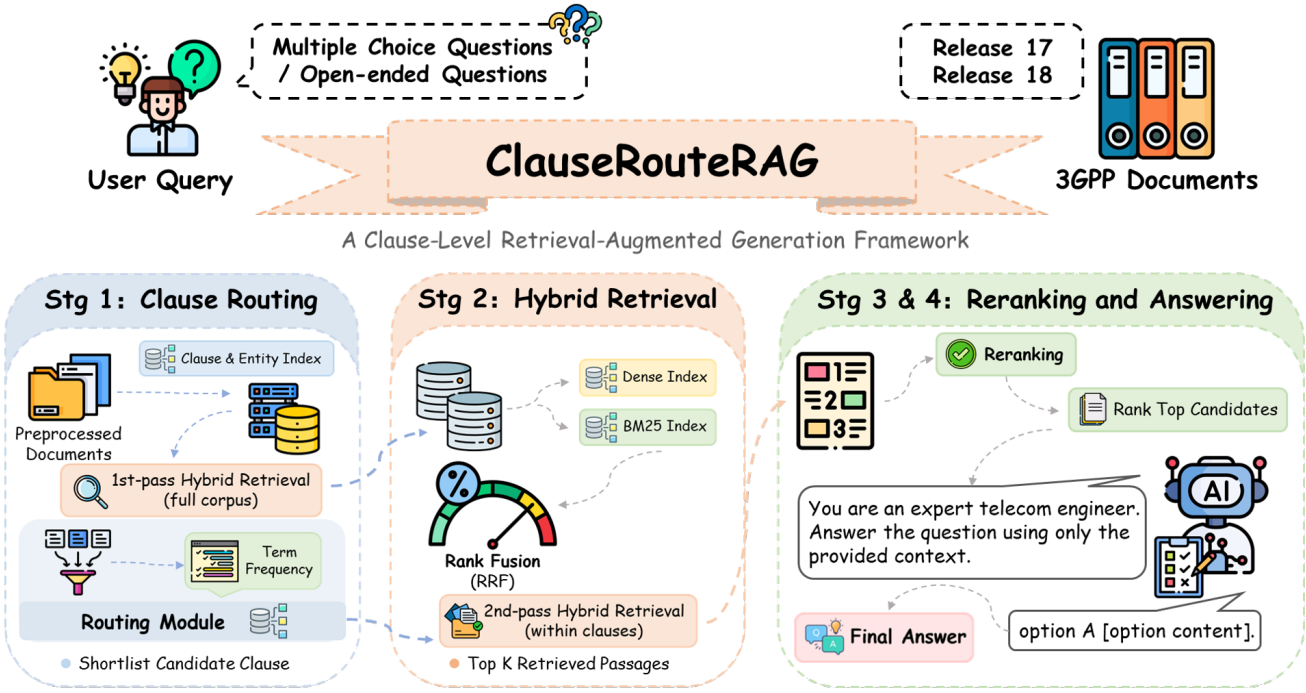

Fig. 1. Overview of ClauseRouteRAG for 3GPP specifications. ClauseRouteRAG builds dense and BM25 indexes over the same passage corpus, along with a clause mapping and an identifier cache. At inference time, it runs a broad hybrid retrieval, aggregates evidence to select candidate clauses, then retrieves and reranks passages within these clauses to form the final evidence set for answer generation.

B. Index Construction and Retrieval Caches

ClauseRouteRAG builds dense and sparse indices over the same passage corpus, using embeddings for semantic similarity and BM25 for lexical matching. We also precompute a clause-to-passage mapping and cache technical identifiers from each passage. These lightweight caches support clause routing and reranking without scanning the full corpus for each query.

For dense retrieval, each passage p_i with text x_i is encoded into an ℓ_2 normalized embedding vector

$$\mathbf{e}_i = \frac{f_{\text{enc}}(x_i)}{\|f_{\text{enc}}(x_i)\|_2}. \tag{2}$$

For a query q, we compute $\mathbf{e}_q$ in the same way and define a dense relevance score by inner product

$$s_{\text{dense}}(q, i) = \mathbf{e}_q^\top \mathbf{e}_i. \tag{3}$$

For sparse retrieval, we rank passages with BM25 [24]. Dense and sparse scores are not calibrated on the same scale, so we merge their rankings using reciprocal rank fusion [25]. Let $r_{\text{dense}}(i)$ and $r_{\text{sparse}}(i)$ be the ranks of passage p_i under dense retrieval and sparse retrieval. The fused score is

$$s_{\text{hyb}}(q, i) = \frac{1}{k_{\text{rrf}} + r_{\text{dense}}(i)} + \frac{1}{k_{\text{rrf}} + r_{\text{sparse}}(i)}, \tag{4}$$

where k_{rrf} is a smoothing constant.

Each passage is associated with a clause identifier c_i recovered during preprocessing. We build a clause mapping

$$\mathcal{M}(c) = \{\, i \mid c_i = c \,\}, \tag{5}$$

and store the parent chain for dotted clause identifiers, for example $5.2.1 \to 5.2 \to 5$.

To support reranking for technical queries, we cache a set of surface form identifiers for each passage. For passage p_i, we extract an identifier set E_i, such as specification numbers, abbreviations, and formatted tokens that often serve as anchors in 3GPP reasoning. These identifiers are extracted using lightweight pattern-based rules together with named-entity recognition, and are retained in their surface forms for matching and reranking. From $\{E_i\}$ we build postings lists

$$\mathcal{P}(e) = \{\, i \mid e \in E_i \,\}, \tag{6}$$

and compute an inverse document frequency weight

$$\text{IDF}(e) = \log \frac{N + 1}{|\mathcal{P}(e)| + 1} + 1, \tag{7}$$

where N is the number of passages.

C. Retrieval with Clause Routing

ClauseRouteRAG retrieves a compact evidence set that is relevant to the query and traceable to clause-level sources. Given a query q, it extracts query signals, routes the search to a small set of candidate clauses, then retrieves and reranks passages within the routed space. Compared with prior routing approaches, our method uses a non-parametric clause selection step grounded in the explicit hierarchical structure of 3GPP documents and aggregates first-pass retrieval evidence at the clause level before restricting the second retrieval pass.

We extract lightweight signals from q that include explicit clause references, specification identifiers, and a set of technical identifiers E_q such as abbreviations, formatted tokens, and specification numbers. When the query contains an explicit clause reference, we also include its parent clauses using the dotted identifier hierarchy.

Clause routing identifies candidate clauses before the second retrieval pass. We first run hybrid retrieval over the full index and obtain the top L passages, denoted by $\mathcal{U}_L$. Each retrieved passage p_i is associated with a clause identifier c_i. For each clause c, we compute a clause score by averaging the scores of the top K_{cl} passages within c

$$S(c) = \frac{1}{K_{\mathrm{cl}}} \sum_{i \in \mathrm{Top}_{K_{\mathrm{cl}}}(\mathcal{U}_L(c))} s_{\mathrm{hyb}}(q, i), \tag{8}$$

where $\mathcal{U}_L(c) = \{i \in \mathcal{U}_L \mid c_i = c\}$. We then select the top N clauses ranked by $S(c)$ and union them with any explicitly referenced clauses and their parent clauses, forming the candidate clause set $\widetilde{\mathcal{C}}$.

Given $\widetilde{\mathcal{C}}$, we construct a candidate pool of passages

$$\mathcal{V} = \bigcup_{c \in \widetilde{\mathcal{C}}} \mathcal{M}(c). \tag{9}$$

If the query contains a specification identifier, we filter $\mathcal{V}$ by the corresponding source document and fall back to the unfiltered pool when the filter yields an empty set. We then run hybrid retrieval within $\mathcal{V}$ and keep the top M passages as reranking candidates.

We rerank these candidates using identifier overlap. For each candidate passage p_i, we compute

$$s_{\mathrm{id}}(q, i) = \sum_{e \in E_q \cap E_i} \mathrm{IDF}(e), \tag{10}$$

normalize $s_{\mathrm{id}}(q, i)$ within the candidate set, and combine it with the hybrid retrieval score

$$s_{\mathrm{final}}(q, i) = (1 - \beta) s_{\mathrm{hyb}}(q, i) + \beta \widehat{s}_{\mathrm{id}}(q, i). \tag{11}$$

Finally, we return the top K_{ev} passages ranked by $s_{\mathrm{final}}(q, i)$ as evidence for answer generation.

We set retrieval and fusion hyperparameters on a held out validation split, and we report the default values in Section IV-A. We use $k_{\mathrm{rrf}} = 60$ following the near optimal setting reported for reciprocal rank fusion. [25]

D. Generation Phase

Given the top K_{ev} evidence passages returned by the retrieval stage, ClauseRouteRAG builds two task-specific prompts, as shown in Fig. 2 and Fig. 3. For multiple choice questions, the prompt concatenates the retrieved passages with their structural metadata, appends the question and options, and constrains the model to output a single selected option in a fixed format. For open-ended questions, the prompt provides the same retrieved context and asks the model to produce a short direct answer, which matches the LLM-Eval style used in prior telecom QA evaluation.

PROMPT FOR MULTIPLE-CHOICE QUESTIONS

System: You are an expert telecom engineer. Use only the provided context to answer the multiple-choice question.If the context is insufficient, output "Insufficient context". Output format: "answer option X [option content]".

Context: {*Retrieved passages* $p_1, \ldots, p_{K_{\mathrm{ev}}}$ *with clause identifiers and title paths*}

Question: {*Question*}

Options: {*Options* $O = \{o_1, \ldots, o_n\}$}

Answer: *[Model output]*

Fig. 2. Prompt used for multiple-choice question answering.

PROMPT FOR OPEN-ENDED QUESTIONS

System: You are an expert telecom engineer. Use only the provided context to answer the question. Give a short direct answer grounded in the context. If the context is not sufficient, output "Insufficient context".

Context: {*Retrieved passages* $p_1, \ldots, p_{K_{\mathrm{ev}}}$ *with sources and clause identifiers*}

Question: {*Question*}

Answer: *[Model output]*

Fig. 3. Prompt used for open-ended question answering.

IV. EXPERIMENTS

A. Overall

In this section, we use the default settings for ClauseRouteRAG as shown in Table I. Based on these settings, we conduct experiments on 3GPP benchmarks to evaluate the effectiveness of ClauseRouteRAG.

TABLE I
DEFAULT SETTINGS USED IN CLAUSEROUTERAG EXPERIMENTS.

Parameter	Setting
Backbone LLM	Llama3-8B-Instruct
Dense embedding model	BGE-M3
Sparse retrieval	BM25
Rank fusion	RRF with $k_{\mathrm{rrf}} = 60$
Chunk length	250 tokens
First pass retrieval size	$L = 200$
Clauses selected by routing	$N = 20$
Passages per clause for routing score	$K_{\mathrm{cl}} = 3$
Reranking candidate size	$M = 50$
Reranking weight	$\beta = 0.15$
Evidence passages to LLM	$K_{\mathrm{ev}} = 5$

B. Datasets, Models and Baselines

We evaluate ClauseRouteRAG on multiple choice question answering over 3GPP specifications, and the retrieval knowledge base in all experiments is constructed from the TSpec-LLM.

a) Datasets: There are two open-source evaluation datasets available for 3GPP multiple choice question answering in our experiments, namely TeleQnA, the TSpec-LLM benchmark dataset and Tele-Eval.

- **TeleQnA.** We use TeleQnA [13] and report results on the Release 17 split with 734 questions and the Release 18 split with 780 questions. We report accuracy for each split and their macro average.
- **TSpec-LLM benchmark dataset.** We also evaluate on the benchmark dataset released with TSpec-LLM [15]. It contains 100 multiple choice questions derived mainly from 3GPP series 36 and 38. This dataset complements TeleQnA by focusing on questions that require careful reading of the specification text.
- **Tele-Eval.** We report results on Tele-Eval [26], an open-ended telecommunications question-answer dataset introduced to complement multiple choice evaluation. Tele-Eval contains 26,225 questions for Release 17 and 32,402 questions for Release 18. To manage evaluation costs, we sample 500 questions each from two Release, following Chat3GPP's preprocessing procedure. We evaluate Tele-Eval with LLM-Eval, using Mixtral-8x7B-Instruct [27] as the judge model. Figure 4 shows the judging prompt used for LLM-Eval.

LLM-EVAL JUDGE PROMPT FOR TELE-EVAL

System: You are a strict evaluator for telecommunications question answering. Decide whether the model answer matches the reference answer for the given question. Return only "**Yes** or **No**".

Question: {*Tele-Eval question*}

Reference Answer: {*Reference answer*}

Model Answer: {*Model answer*}

Fig. 4. Judging prompt used for LLM-Eval on Tele-Eval.

b) Models: To test how ClauseRouteRAG behaves with different backbone LLMs, we run the same retrieval and prompting pipeline with three instruction tuned models: Llama3-8B-Instruct [28], Qwen2.5-7B-Instruct [29], and Phi-4-mini-Instruct [30].

c) Baselines: We compare ClauseRouteRAG with widely used LLM and RAG baselines for question answering over telecom standards. For the LLM baseline, we use Llama-3-8B-Tele-it, the best-performing model reported in the Tele-LLMs series and instruction tuned for telecommunications [26].

For RAG baselines, we include Chat3GPP [9], Telco-RAG [17], and Telco-oRAG [21], which are open-source systems built for 3GPP-style documents.We also implement a Naive RAG baseline, which implements the standard retrieve-then-generate pattern without any domain-specific optimizations.

C. Results

We evaluate ClauseRouteRAG on the TSpec-LLM benchmark dataset. For TSpec-LLM, we report accuracy on the Easy, Intermediate, and Hard subsets, as well as the overall score, to measure performance under different levels of question difficulty.

TABLE II
ACCURACY ON THE TSPEC-LLM BENCHMARK DATASET.

Method	Easy	Inter.	Hard	Overall
Base model	0.500	0.510	0.316	0.470
Naive RAG	0.706	0.790	0.700	0.720
ClauseRouteRAG	**0.900**	**0.842**	**0.902**	**0.890**

Table II shows that ClauseRouteRAG achieves the best overall accuracy at 89%. It also remains strong across difficulty levels, and it brings the largest gain on the Hard subset compared with the base model.

We next evaluate ClauseRouteRAG on TeleQnA. Following common practice in telecom question answering, we exclude answer options in retrieval and add the options only in the final generation prompt. For Chat3GPP, Telco-RAG, and Telco-oRAG, we report the best accuracy values stated in their papers. We report accuracy for the Release 17 subset and the Release 18 subset using the corresponding release knowledge base. We also report an Overall score computed by evaluating the full TeleQnA set under a single knowledge base that contains documents from both Release 17 and Release 18.

TABLE III
ACCURACY ON TELEQNA. "–" INDICATES THAT THE RESULT IS NOT REPORTED."FINE-TUNING" MEANS THE BACKBONE MODEL USED IN THE REFERENCED RESULT IS TELECOM TUNED.

Method	Rel17	Rel18	Overall	Fine-tuning
Llama3-8B-Tele-it	0.531	0.571	0.552	Yes
Telco-RAG	0.725	0.784	–	Yes
Chat3GPP	0.783	0.791	0.787	No
Telco-oRAG	–	0.809	–	Yes
ClauseRouteRAG	**0.837**	**0.856**	**0.849**	No

Table III shows that ClauseRouteRAG achieves the best overall accuracy among the methods listed in this comparison. ClauseRouteRAG improves over Chat3GPP on both Release 17 and Release 18 without using a finetuned backbone model. Compared with Telco-RAG and Telco-oRAG, ClauseRouteRAG reaches higher accuracy without training an extra router component, which suggests that the gains come from evidence selection in retrieval.

We further report results on Tele-Eval using LLM-Eval. We compare ClauseRouteRAG with the same set of baselines. We report scores for the Release 17 subset and the Release 18 subset, and we report an Overall score under the same setting as TeleQnA.

Table IV shows that retrieval helps on open ended questions, and ClauseRouteRAG gives the strongest results in this setting. The gains indicate that clause routing and reranking still improve evidence quality when answers are not restricted to multiple choice options. The results also show that a general

TABLE IV
ACCURACY ON TELE-EVAL USING LLM-EVAL.

Method	Rel17	Rel18	Overall
Base model	0.241	0.232	0.236
Naive RAG	0.356	0.368	0.362
Llama3-8B-Tele-it	0.287	0.271	0.279
Chat3GPP	0.487	0.542	0.515
ClauseRouteRAG	**0.553**	**0.592**	**0.573**

backbone with ClauseRouteRAG can match or exceed telecom finetuned baselines under the same judge protocol.

We study how ClauseRouteRAG performs with different backbone LLMs on the TeleQnA Release 17. Figure 5 compares the base model, naive RAG, and ClauseRouteRAG using three instruction tuned backbones.

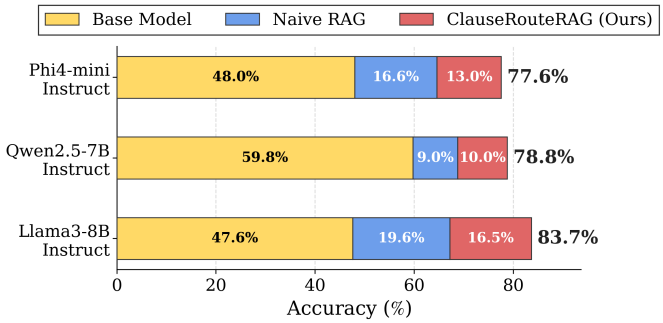

Fig. 5. Accuracy on TeleQnA Release 17 with different backbone LLMs.

Figure 5 shows that retrieval improves accuracy across all evaluated backbones, and ClauseRouteRAG provides an additional improvement over naive RAG. After applying ClauseRouteRAG, the smaller Phi-4-mini-Instruct reaches a similar accuracy level to Qwen2.5-7B-Instruct, which suggests that the method can help lightweight models remain competitive on standards question answering.

D. Ablation Study

We quantify the contribution of key components in ClauseRouteRAG. We start from a Naive RAG baseline and add one component at a time. All runs use the same backbone model and the same default settings as in Section IV-A.

TABLE V
INCREMENTAL ABLATION RESULTS ON TELEQNA RELEASE 17. EACH ROW ADDS ONE COMPONENT TO THE PREVIOUS SETTING.

Stage	Method	Rel17
Baseline	Naive RAG (dense only)	0.640
Retrieval	+ BM25 in hybrid retrieval (RRF)	0.672
Indexing	+ Table row passages retained	0.702
Routing	+ Clause routing	0.812
Rerank	+ Identifier overlap reranking	**0.837**

Table V shows that clause routing brings the largest gain. This step restricts retrieval to a small set of relevant clauses, which reduces interference from similar wording across long specifications. BM25 and table row passages help when questions rely on exact tokens, abbreviations, or parameter values. Identifier overlap reranking provides a smaller but consistent gain by preferring evidence that matches the key identifiers in the query.

V. CONCLUSIONS

In this paper, we introduce ClauseRouteRAG, a clause-level RAG framework for 3GPP telecommunications standards. ClauseRouteRAG selects a small set of relevant clauses, retrieves evidence within these clauses using dense retrieval and BM25, and reranks candidates with technical identifier overlap. This design reduces noise from long specifications and keeps the final evidence traceable to clause locations. Experiments show consistent gains over LLM-only and RAG baselines.

However, several limitations remain. Identifier extraction uses lightweight rules and NER, which can miss implicit anchors or inconsistent aliases. The current pipeline treats tables as linear text and does not model table structure or figures.In future work, we plan to explore further optimizations to better support a wider range of tasks within the telecommunications domain.

REFERENCES

[1] L. Bariah, H. Zou, Q. Zhao, B. Mouhouche, F. Bader, and M. Debbah, "Understanding telecom language through large language models," in *GLOBECOM 2023-2023 IEEE Global Communications Conference*. IEEE, 2023, pp. 6542–6547.

[2] J. Bai, X. Wang, Z. Xiao, H. Zhou, T. A. A. Ali, Y. Li, and L. Jiao, "Achieving efficient feature representation for modulation signal: A cooperative contrast learning approach," *IEEE Internet of Things Journal*, vol. 11, no. 9, pp. 16 196–16 211, 2024.

[3] Y. Wang, J. Bai, Z. Xiao, H. Zhou, and L. Jiao, "Msmcnet: A modular few-shot learning framework for signal modulation classification," *IEEE Transactions on Signal Processing*, vol. 70, pp. 3789–3801, 2022.

[4] A. Damnjanovic, J. Montojo, Y. Wei, T. Ji, T. Luo, M. Vajapeyam, T. Yoo, O. Song, and D. Malladi, "A survey on 3gpp heterogeneous networks," *IEEE Wireless communications*, vol. 18, no. 3, pp. 10–21, 2011.

[5] A. Maatouk, N. Piovesan, F. Ayed, A. De Domenico, and M. Debbah, "Large language models for telecom: Forthcoming impact on the industry," *IEEE communications magazine*, 2024.

[6] X. Qian, Z. He, W. Wang, W. Yue, X. Yao, and G. Cheng, "Introducing wsod-sam proposals and heuristic pseudo-fully supervised training strategy for weakly supervised object detection in remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 64, pp. 1–12, 2026.

[7] H. Zhou, C. Hu, Y. Yuan, Y. Cui, Y. Jin, C. Chen, H. Wu, D. Yuan, L. Jiang, D. Wu *et al.*, "Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities," *IEEE Communications Surveys & Tutorials*, vol. 27, no. 3, pp. 1955–2005, 2024.

[8] G. M. Yilma, J. A. Ayala-Romero, A. Garcia-Saavedra, and X. Costa-Perez, "Telecomrag: Taming telecom standards with retrieval augmented generation and llms," *ACM SIGCOMM Computer Communication Review*, vol. 54, no. 3, pp. 18–23, 2025.

[9] L. Huang, M. Zhao, L. Xiao, X. Zhang, and J. Hu, "Chat3gpp: An open-source retrieval-augmented generation framework for 3gpp documents," *arXiv preprint arXiv:2501.13954*, 2025.

[10] P. Sousa, C. K. Mello, F. B. Morte, and L. F. S. Navarro, "Bisecting k-means in rag for enhancing question-answering tasks performance in telecommunications," in *2024 IEEE Globecom Workshops (GC Wkshps)*. IEEE, 2024, pp. 1–6.

[11] D. Yuan, H. Zhou, D. Wu, X. Liu, H. Chen, Y. Xin, and J. C. Zhang, "Enhancing large language models (llms) for telecommunications using knowledge graphs and retrieval-augmented generation," in *2025 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2025, pp. 486–491.

[12] J. Bai, L. Si, Z. Xiao, X. Jiang, T. Li, Y. Zhang, and L. Jiao, "Jamming signal power detection based on masked convolutional encoder-decoder network," *IEEE Transactions on Cognitive Communications and Networking*, 2025.

[13] A. Maatouk, F. Ayed, N. Piovesan, A. De Domenico, M. Debbah, and Z.-Q. Luo, "Teleqna: A benchmark dataset to assess large language models telecommunications knowledge," *IEEE Network*, 2025.

[14] H. Zou, Q. Zhao, Y. Tian, L. Bariah, F. Bader, T. Lestable, and M. Debbah, "Telecomgpt: A framework to build telecom-specific large language models," *IEEE Transactions on Machine Learning in Communications and Networking*, 2025.

[15] R. Nikbakht, M. Benzaghta, and G. Geraci, "Tspec-llm: An open-source dataset for llm understanding of 3gpp specifications," *arXiv preprint arXiv:2406.01768*, 2024.

[16] J. D. A. P. L. Júnior, J. Pereira, D. S. Do Carmo, R. Lotufo, and C. E. Rothenberg, "Lightweight llms for 3gpp specifications: Fine-tuning, retrieval-augmented generation and quantization," in *2025 IEEE 11th International Conference on Network Softwarization (NetSoft)*. IEEE, 2025, pp. 37–42.

[17] A.-L. Bornea, F. Ayed, A. De Domenico, N. Piovesan, and A. Maatouk, "Telco-rag: Navigating the challenges of retrieval augmented language models for telecommunications," in *GLOBECOM 2024-2024 IEEE Global Communications Conference*. IEEE, 2024, pp. 2359–2364.

[18] T. Saraiva, M. Sousa, P. Vieira, and A. Rodrigues, "Telco-dpr: A hybrid dataset for evaluating retrieval models of 3gpp technical specifications," in *2025 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2025, pp. 01–06.

[19] O. Erak, N. Alabbasi, O. Alhussein, I. Lotfi, A. Hussein, S. Muhaidat, and M. Debbah, "Leveraging fine-tuned retrieval-augmented generation with long-context support: For 3gpp standards," *arXiv preprint arXiv:2408.11775*, 2024.

[20] N. Alabbasi, O. Erak, O. Alhussein, I. Lotfi, S. Muhaidat, and M. Debbah, "Teleoracle: Fine-tuned retrieval-augmented generation with long-context support for networks," *IEEE Internet of Things Journal*, 2025.

[21] A.-L. Bornea, F. Ayed, A. De Domenico, N. Piovesan, T. S. Salem, and A. Maatouk, "Telco-orag: Optimizing retrieval-augmented generation for telecom queries via hybrid retrieval and neural routing," *arXiv preprint arXiv:2505.11856*, 2025.

[22] R. Ghosh, C.-H. Liu, G. Rele, and V. S. Ravipati, "Telcoai: Advancing 3gpp technical specification search through agentic multi-modal retrieval-augmented generation," 2025. [Online]. Available: https://www.amazon.science/publications/telcoai-advancing-3gpp-technical-specification-search-through-agentic-multi-modal-retrieval-augmented-generation

[23] A. G. Manvattira, "Deepspecs expert-level questions answering in 5g," Master's thesis, University of California, Los Angeles, 2025.

[24] S. Robertson and H. Zaragoza, *The probabilistic relevance framework: BM25 and beyond*. Now Publishers Inc, 2009, vol. 4.

[25] G. V. Cormack, C. L. Clarke, and S. Buettcher, "Reciprocal rank fusion outperforms condorcet and individual rank learning methods," in *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 2009, pp. 758–759.

[26] A. Maatouk, K. C. Ampudia, R. Ying, and L. Tassiulas, "Tele-llms: A series of specialized large language models for telecommunications," *arXiv preprint arXiv:2409.05314*, 2024.

[27] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand *et al.*, "Mixtral of experts," *arXiv preprint arXiv:2401.04088*, 2024.

[28] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[29] Q. Team *et al.*, "Qwen2 technical report," *arXiv preprint arXiv:2407.10671*, vol. 2, no. 3, 2024.

[30] A. Abouelenin, A. Ashfaq, A. Atkinson, H. Awadalla, N. Bach, J. Bao, A. Benhaim, M. Cai, V. Chaudhary, C. Chen *et al.*, "Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras," *arXiv preprint arXiv:2503.01743*, 2025.