Lightweight LLMs for 3GPP Specifications: Fine-Tuning, Retrieval-Augmented Generation and Quantization

José de Arimatéa Passos Lopes Júnior
Unicamp
Campinas, Brazil
j271467@g.unicamp.br

Jayr Pereira
Universidade Federal do Cariri
Juazeiro do Norte, Brazil
jayr.pereira@ufca.edu.br

Diedre Santos do Carmo
Unicamp
Campinas, Brazil
diedre@unicamp.br

Roberto Lotufo
Unicamp
Campinas, Brazil
lotufo@unicamp.br

Christian Esteve Rothenberg
Unicamp
Campinas, Brazil
chesteve@unicamp.br

***Abstract*—Interpreting complex 3GPP telecommunications standards for question and answering (QA) poses a challenge for general-purpose LLMs due to their specialized terminology and high computational demands, limiting their use in resource-constrained environments. This work explores an efficient, open-source approach using the TeleQnA dataset of 10,000 telecom questions and the TSpec-LLM repository of processed 3GPP documents. We enhance a lightweight Llama 3.2 (3B parameters) model, quantized from 16-bit precision to 4 bits, through fine-tuning and RAG to improve accuracy without heavy resource reliance. Unlike prior resource-intensive or proprietary solutions, our method reduces memory demands, enabling deployment on modest hardware like edge devices or softwarized networks. Shared via GitHub repositories [1], this approach advances cost-effective, reproducible AI for telecommunications QA, supporting contexts where budgets, computation, or public internet access are limited.**

***Index Terms*—LLM, Fine-Tuning, RAG, 4Bit-Quantization, 3GPP**

I. Introduction

The rapid rise of Large Language Models (LLMs) is transforming telecommunications, particularly in interpreting complex technical standards like those from the 3rd Generation Partnership Project (3GPP). These standards, foundational to modern network infrastructures, feature specialized terminology and evolve continuously, posing challenges for general-purpose LLMs [2], [3]. While public commercial models offer impressive capabilities, their high computational requirements and costs often render them impractical for resource-constrained environments such as edge devices or softwarized and/or private networks.

Existing research highlights a gap in addressing these practical constraints. Studies like [4] emphasize the need for telecom-specific LLMs, yet often assume resource-intensive setups, while [5] provides processed 3GPP datasets without tackling deployment limitations. Similarly, [6] showcases fine-tuning benefits but relies on proprietary data, hindering reproducibility. Meanwhile, [7] applies LLMs to Intent-Based Networking (IBN), illustrating their potential in network management, though it overlooks efficiency in resource-limited settings. However, effectively leveraging LLMs in IBN requires models that not only process telecom-specific terminology but also comprehend and translate high-level intent into precise network configurations and documentation. Network automation demand solutions that minimize operational costs and computational overhead while maintaining accuracy and adaptability. Addressing this gap requires specialized, lightweight models capable of efficiently interpreting user requests, generating optimized configurations, and enhancing network management with a balance between performance and resource efficiency.

In this work, we present an open-source approach to enhance LLMs for 3GPP comprehension, leveraging fine-tuning and Retrieval-Augmented Generation (RAG). We adapt Llama 3.2 (3B parameters), quantized from 16-bit precision to 4 bits to substantially reduce memory demands, using the TeleQnA dataset [8] of 10,000 telecom questions, and integrate RAG with the TSpec-LLM repository [5]. This combination refines the model's telecom expertise, retrieves relevant context, and enables deployment on modest hardware, offering a cost-effective alternative to resource-heavy or proprietary models.

Experiments demonstrate that our quantized model achieves a performance comparable to GPT-4o-mini (76.3% vs. 74.0% average accuracy with RAG), despite requiring significantly less memory(2.768 GB and 3.5 GB peak). The 4-bit quantization enables inference on consumer-grade GPUs while maintaining robust accuracy in technical question answering. These findings reinforce the feasibility of deploying specialized LLMs in constrained environments, broadening accessibility to AI-driven telecom solutions.

To facilitate reproduction of our results, we provide two complementary repositories: our comprehensive codebase [1] containing all experimental results and testing configurations, and a streamlined implementation [9] that includes only the essential files required to reproduce our key findings. This lightweight repository reduces barriers to verification by focusing solely on the core components necessary to replicate our approach.

II. Related Work

Recent advancements in LLMs have driven efforts to interpret technical telecommunications standards, such as those from 3GPP. Piovesan et al. (2023) [8] introduced TeleQnA, a benchmark dataset of 10,000 questions spanning general and technical telecom knowledge, enabling evaluation of LLMs in this domain. Complementing this, Benzaghta et al. (2024) [5] developed TSpec-LLM, a repository of processed 3GPP documents (Releases 8–19), facilitating retrieval-based approaches like RAG to enhance response accuracy with structured content.

Other studies have explored domain-specific enhancements. Bornea et al. (2024) [10] proposed Telco-RAG, a framework integrating retrieval mechanisms with LLMs to improve performance on telecom queries, emphasizing tailored processing of technical documents. Similarly, Nikou et al. (2024) [6] presented TeleRoBERTa, a RoBERTa-based model fine-tuned on a proprietary telecom dataset, achieving notable gains in 3GPP standards comprehension. Additional research, such as [4], underscores the need for telecom-specialized LLMs, though often relying on resource-intensive setups and public internet-based APIs.

Despite these advances, limitations persist in accessibility and efficiency. Works like [4] and [5] tend to assume computationally heavy models, while [6] restricts reproducibility with closed data. Our approach addresses these gaps by combining domain-specific fine-tuning and RAG on a lightweight Llama 3.2 (3B parameters) model, quantized to 4 bits, using TeleQnA and TSpec-LLM. This reduces memory demands, supports deployment on modest hardware, and ensures transparency through fully open datasets and code [1], contrasting with prior resource-heavy or proprietary solutions.

III. Methodology

This study evaluates LLMs' ability to answer 3GPP-related questions from Releases 17, 18, and earlier releases, using the TeleQnA dataset [8] of 10,000 telecom questions and multiple choice answers as the benchmark. Experiments assess performance on 600 multiple-choice questions, evenly distributed (200 each) across Releases 17, 18, and others, chosen to reflect diverse telecom standards while maintaining manageable computation on consumer hardware. The overall workflow, including data processing, model fine-tuning, RAG integration, and evaluation, is illustrated in Figure 1.

Two test conditions were applied:

1) **No context**: Models received only the question and options, testing baseline telecom knowledge.
2) **RAG-enhanced**: RAG retrieved relevant context from TSpec-LLM [5] documents, aiming to enhance accuracy with minimal resource overhead.

Models selected answers from provided options, with accuracy calculated as the proportion of correct responses. To standardize evaluation and improve response quality, prompt engineering combined Chain of Thought [11] and Few-Shot Learning [12] . Chain of Thought prompted models to reason step-by-step before outputting answers in the format "correct option: X", where X is the chosen letter. Few-Shot Learning included five RAG-retrieved contexts as examples in the prompt, enhancing performance on technical queries while keeping resource demands low. These strategies ensured consistent evaluation across models and conditions, supporting efficient and accessible telecom AI deployable on modest hardware.

A. Models

Three models were tested:

- **GPT-4o-mini**: A public OpenAI proprietary model accessed via API, serving as a baseline.
- **Llama 3.2 3B**: A 3-billion-parameter model from Meta, quantized to 4 bits via Unsloth [13], selected for its recent release and compact size, enabling local execution, including RAG, on a personal NVIDIA RTX 3050 Ti GPU (3.712 GB max memory).
- **Llama-4bit-Tuned**: The same Llama 3.2 3B, fine-tuned on 4,000 TeleQnA questions (500 from Release 17, 3,500 from prior releases) and quantized to 4 bits.

Llama 3.2 3B was chosen over alternatives (e.g., Gemma, Mistral) as it offered an effective balance between its manageable 3-billion-parameter design and performance, paired with Unsloth's framework [14], which provided one of the most efficient open-source models for quantization and fine-tuning. This combination reduced memory usage to 2.768 GB, fitting within the 3.712 GB capacity of a single consumer GPU and enabling full local deployment, aligning with the study's focus on lightweight, reproducible telecom solutions.

B. Quantization and Fine-tuning Framework: Unsloth

Unsloth [13] is an open-source library that optimizes LLM training and inference across diverse hardware. Its selection was driven by key aspects that enabled our model's efficiency both in training and deployment:

- **Simplified fine-tuning**: It uses LoRA for parameter-efficient training, targeting 24.3M parameters.
- **Quantization support**: It reduces memory usage from 6.688 GB to 2.768 GB at 4 bits, enabling efficient inference on consumer GPUs.
- **Training speed**: It accelerates fine-tuning, completing 4,000 TeleQnA questions in approximately 42 minutes on Google Colab, compared to longer times observed with standard frameworks, ensuring rapid iteration without sacrificing performance.

Training Dataset. A subset of 4,000 questions from TeleQnA [15] was selected, with 500 from Release 17 and 3,500 from prior releases (excluding Releases 17 and 18). Older releases were prioritized for training to reserve newer releases (17 and 18) for testing, though 500 Release 17 questions were included to incorporate recent standards. Two context-response formats were used and shuffled to refine answer precision and avoid bias toward option-based responses, enabling future evaluation with open-ended questions:

- For the first 2,000 questions:
  - Context: question and options
  - Expected Response: the correct option and its text (e.g., ”option 2: It uses stored info”)
- For the last 2,000 questions:
  - Context: only the question
  - Expected Response: only the text of the correct option

Training Parameters. Training was conducted on Google Colab using the unquantized model (6.688 GB in 16-bit precision), enabling a 42-minute process for 4,000 questions. Post-training, the model, named **Llama-4bit-Tuned**, was quantized to 4 bits (2.768 GB) for local inference on an RTX 3050 Ti GPU. Key parameters were:

- Number of trainable parameters: 24,313,856
- Batch size: 32 (2 per device, 16 gradient accumulation steps)
- Total steps: 400 (3.2 epochs)
- Learning rate: 0.0002
- Optimizer: AdamW (8-bit precision)
- Initial loss: 3.207
- Final loss: 0.333

Code is available at [1] under ‘`/Source/Fine_tuning/`’.

C. *Retrieval-Augmented Generation (RAG)*

RAG enhances LLMs by retrieving relevant context from TSpec-LLM [16] documents. Chunks were generated using `MarkdownHeaderTextSplitter` with a 2,000-character size and 100-character overlap, yielding 780,651 chunks. This size was chosen after local tests, balancing sufficient context for the 3B-parameter model against input size, with 5 chunks retrieved per query to optimize accuracy without excessive computation. Embeddings were created with `SentenceTransformer(’all-mpnet-base-v2’)` over 4 hours on an RTX 3050 Ti GPU, while Faiss indexed vectors for cosine-similarity searches.

RAG inference peaked at 3.5 GB of GPU memory, nearing the 3.712 GB limit, proving feasibility on modest hardware. These parameters minimized retrieval overhead while supporting effective telecom question-answering, aligning with the study’s efficiency goals.

IV. Evaluation

GPT-4o-mini, Llama 3.2 3B, and Llama-4bit-Tuned were evaluated on 600 multiple-choice questions from the TeleQnA dataset [8], comprising 200 questions each from Releases 17 and 18, and 200 from other releases. This subset was selected to evenly represent recent and historical 3GPP standards while fitting computational constraints. Using a Chain of Thought [11] prompting approach, the models were instructed to reason step-by-step before selecting an answer. Accuracy served as the evaluation metric, allowing for performance comparison across the question sets.

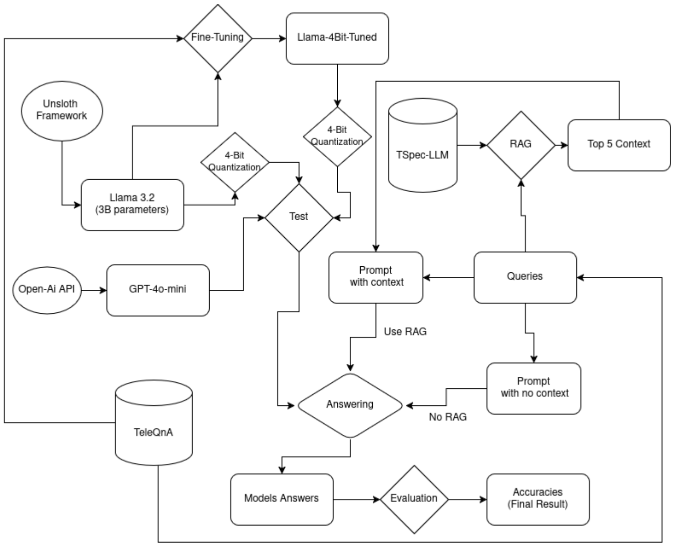

Fig. 1: Workflow of the proposed methodology, illustrating the process from data preparation (TeleQnA, TSpec-LLM) to model fine-tuning (Llama 3.2 3B via Unsloth), RAG integration (top-5 contexts), prompting (with or without RAG), and evaluation (accuracy).

A. *Prompts*

Evaluations were conducted with and without Retrieval-Augmented Generation (RAG). RAG provided additional context retrieved from the TSpec-LLM dataset. Without RAG, models relied solely on the provided question and answer options. The specific prompts used for questions with options were:

- Without RAG:

```
f"Question: {question}\n"
f"Options:\n" + "\n".join(options) + "\n"
"Think step by step before answering and
    respond with the correct option in the
    format ’correct option: <X>’."
```

- With RAG:

```
f"Relevant Information:\n{rag_results}\n"
f"Question: {question}\n"
f"Options:\n" + "\n".join(options) + "\n"
"Think step by step and choose the correct
    option.\n"
"You must respond in the format ’correct
    option: <X>’, where <X> is the correct
    letter for the option."
```

B. *Results*

Accuracy. Table I shows the accuracy results across different TeleQnA releases datasets of the three models GPT-4o-mini, Llama 3.2 3B, and Llama-4bit-Tuned on 600 multiple-choice questions with and without RAG.

Overall, RAG-enabled models outperformed their non-RAG counterparts, with the exception of GPT-4o-mini on the “other releases” subset. While RAG-enabled GPT-4o-mini achieved the highest accuracy on Release 17 and the “other releases” subset (closely followed by Llama-4bit-Tuned), Llama-4bit-Tuned significantly outperformed all other models on Release 18, resulting in the highest average accuracy across all results.

Furthermore, the substantial performance difference between the fine-tuned Llama model and its non-tuned counterpart underscores the effectiveness of the fine-tuning process. These results suggest that fine-tuning, especially in conjunction with RAG, significantly improves the model accuracy of Llama-based architectures for the understanding of technical standards such as 3GPP.

Confidence Interval. To provide a comprehensive view of the models’ performance across all 600 multiple-choice questions, Figure 2 presents a consolidated accuracy comparison for the GPT-4o-mini, Llama 3.2 3B, and Llama-4bit-Tuned models. The results include the calculated confidence intervals for each model’s accuracy for RAG and non-RAG conditions. The confidence intervals (CIs) were calculated using the Wilson score interval, a method commonly used for proportions such as accuracy. For each model, the CI was computed as $\hat{p} \pm z\sqrt{\frac{\hat{p}(1-\hat{p})}{n}}$, where $\hat{p}$ is the observed accuracy, $n = 600$ is the total number of questions, and $z = 1.96$ is the critical value for a 95% confidence level. This ensures that the reported intervals represent the range within which the true accuracy is expected to fall with 95% confidence.

C. Discussion

The evaluation across 600 multiple-choice questions from the TeleQnA dataset [8] revealed distinct accuracy patterns for GPT-4o-mini, Llama 3.2 3B, and Llama-4bit-Tuned, both with and without Retrieval-Augmented Generation (RAG). With RAG, models achieved higher accuracy by leveraging contextual information from the TSpec-LLM dataset [5]. Llama-4bit-Tuned, despite being a quantized model, obtained the highest accuracy (76.3%), followed by GPT-4o-mini (74.0%) and Llama 3.2 3B (67.8%). Without RAG, the accuracy dropped to 64.3% for GPT-4o-mini, 61.0% for Llama-4bit-Tuned, and 54.7% for Llama 3.2 3B. Table I illustrates that RAG consistently improved accuracy across different releases, except for GPT-4o-mini in the ”other releases” category, where performance remained stable at 76.0%.

The accuracy gap between Llama-4bit-Tuned and Llama 3.2 3B was consistent across releases: 74.0% vs. 67.5% on Release 17, 81.5% vs. 69.0% on Release 18, and 73.5% vs. 67.0% on other releases. This improvement reflects the impact of fine-tuning on 4,000 TeleQnA questions, leveraging LoRA [14] to adjust 24.3M parameters for optimal performance with minimal computational overhead. An intermediate fine-tuning step with 2,000 questions showed relevant improvements, but increasing to 4,000 further refined performance, particularly by prioritizing older releases in training while reserving Releases 17 and 18 for evaluation. This strategy yielded a marginal yet meaningful accuracy boost, validating the effectiveness of task-specific fine-tuning in constrained environments.

Model	RAG	Rel. 17	Rel. 18	Other releases	Average Accuracy
GPT-4o-mini	yes	**75.0%**	71.0%	**76.0%**	74.0%
Llama-4Bit-Tuned	yes	74.0%	**81.5%**	73.5%	**76.3%**
Llama 3.2	yes	67.5%	69.0%	67.0%	67.8%
GPT-4o-mini	no	61.5%	55.5%	**76.0%**	64.3%
Llama-4Bit-Tuned	no	55.0%	60.0%	68.0%	61.0%
Llama 3.2	no	49.5%	52.0%	62.5%	54.7%

TABLE I: Accuracy results (in bold the best one) for different models across multiple releases.

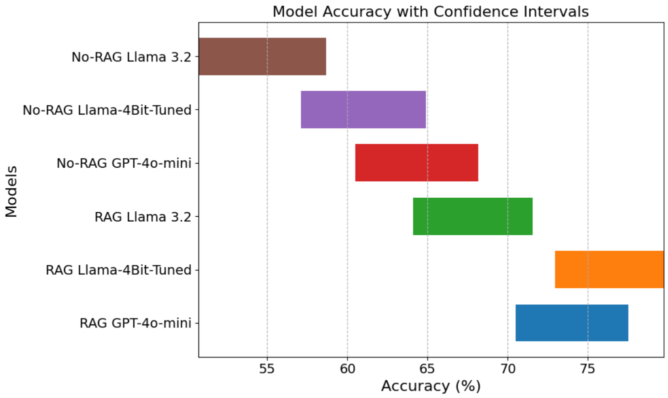

Fig. 2: Consolidated accuracy results with 95% confidence intervals for all 600 questions across the three models (GPT-4o-mini, Llama 3.2 3B, and Llama-4bit-Tuned) under RAG and non-RAG conditions.

Llama-4bit-Tuned’s accuracy with RAG (76.3%) slightly exceeded that of GPT-4o-mini (74.0%), despite the latter being a larger proprietary model not specifically adapted to this task. Their confidence intervals overlap, suggesting comparable statistical performance within this test set. Crucially, this result demonstrates that even a quantized model, which inherently sacrifices some representational power for efficiency, can achieve competitive accuracy in domain-specific tasks when combined with RAG. This approach provides a practical alternative for applications requiring high accuracy while operating on hardware with limited computational capacity, such as edge devices or consumer-grade GPUs.

RAG played a crucial role in mitigating limitations associated with smaller models, particularly for Llama 3.2 3B, where accuracy increased from 54.7% to 67.8%—a 13.1% gain—compared to a 9.7% improvement for GPT-4o-mini (64.3% to 74.0%). This suggests that retrieval-based augmentation is particularly effective in compensating for the degraded internal knowledge of quantized and parameter-constrained models. The evaluation, structured around 600 multiple-choice questions (200 per release), was designed to balance recent and historical standards, though real-world telecom applications may involve more complex, open-ended queries.

The proposed solution—combining 4-bit quantization, fine-tuning, and RAG—demonstrates that high-performance LLMs can be deployed efficiently on hardware with limited resources without significant loss in accuracy. The 4-bit quantization enabled local inference on a consumer GPU, validating its viability for resource-constrained environments. These results confirm that it is possible to bridge the performance gap between commercial models and open-source alternatives through targeted adaptations, ensuring accessibility to high-quality NLP models in scenarios where computational resources are limited.

V. Concluding Remarks

This work demonstrated that fine-tuning a relatively small Llama model, combined with retrieval-augmented generation (RAG) and 4-bit quantization, enables efficient adaptation to specialized domains while maintaining strong performance. Despite its reduced size, the quantized model performs comparably to commercial alternatives, making it a practical solution for deployment on resource-constrained devices. These findings highlight the potential of optimizing smaller models to balance computational efficiency and domain-specific expertise.

While fine-tuning enhances domain adaptation, small models inherently retain limitations from their pre-training phase. However, the integration of RAG mitigates these constraints by dynamically retrieving external knowledge, reducing reliance on extensive internal representations. This demonstrates that a carefully optimized small model—leveraging fine-tuning, retrieval, and quantization—can effectively process specialized information with minimal computational overhead.

Our results confirm that 4-bit quantization significantly reduces memory consumption while preserving the model's interpretative capabilities, even for complex technical standards like 3GPP. By enabling execution on low-resource hardware, this approach expands AI accessibility, making specialized models viable for local deployment. This suggests that quantized LLMs can serve as effective alternatives to larger, resource-intensive models in real-world applications.

Limitations & Future Research. Despite these promising results, some aspects warrant further exploration:

1) The evaluation was conducted using multiple-choice questions. Future work should explore open-ended question-answering scenarios, where models generate responses instead of selecting predefined options. Metrics such as RAGAS [17] could provide valuable insights in these cases.
2) Investigating the impact of significantly larger training datasets—beyond the scale of TeleQnA—could reveal whether additional data further improves model generalization and performance in specialized tasks.
3) Due to hardware constraints, the study was limited to smaller models. Future research could evaluate whether larger quantized models, such as Llama 8B, offer superior performance while maintaining computational efficiency.
4) The comparison was restricted to GPT-4o-mini for cost reasons. Benchmarking against larger commercial models like GPT-4, GPT-4o, or DeepSeek-V3 could provide a more comprehensive assessment of the proposed approach.
5) The use of Agents or Multi-Agent systems could be explored to refine retrieval and response generation. Allowing iterative interactions may improve answer quality, particularly for complex queries where initial retrievals are insufficient.

References

[1] J. de Arimatéa Passos Lopes Júnior, "3GPP LLM Evaluation," gitHub repository. [Online]. Available: https://github.com/josearimatea/3gpp_llm_evaluation, 2024, accessed: Jan. 2025.

[2] H. Ghasemirahni, A. Farshin, M. Scazzariello, M. Chiesa, and D. Kostić, "Deploying stateful network functions efficiently using large language models," in *Proceedings of the 4th Workshop on Machine Learning and Systems*, ser. EuroMLSys '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 28–38. [Online]. Available: https://doi.org/10.1145/3642970.3655836

[3] C. Wang, M. Scazzariello, A. Farshin, S. Ferlin, D. Kostić, and M. Chiesa, "Netconfeval: Can llms facilitate network configuration?" *Proc. ACM Netw.*, vol. 2, no. CoNEXT2, Jun. 2024. [Online]. Available: https://doi.org/10.1145/3656296

[4] A. Maatouk, K. C. Ampudia, R. Ying, and L. Tassiulas, "Tele-LLMs: A Series of Specialized Large Language Models for Telecommunications," 2024. [Online]. Available: https://arxiv.org/abs/2409.05314

[5] R. Nikbakht, M. Benzaghta, and G. Geraci, "TSpec-LLM: An Open-source Dataset for LLM Understanding of 3GPP Specifications," 2024. [Online]. Available: https://arxiv.org/abs/2406.01768

[6] A. Karapantelakis, M. Thakur, A. Nikou, F. Moradi, C. Orlog, F. Gaim, H. Holm, D. D. Nimara, and V. Huang, "Using Large Language Models to Understand Telecom Standards," 2024. [Online]. Available: https://arxiv.org/abs/2404.02929

[7] A. Mekrache and A. Ksentini, "LLM-enabled Intent-driven Service Configuration for Next Generation Networks," in *2024 IEEE 10th International Conference on Network Softwarization (NetSoft)*, 2024, pp. 253–257.

[8] A. Maatouk, F. Ayed, N. Piovesan, A. D. Domenico, M. Debbah, and Z.-Q. Luo, "TeleQnA: A Benchmark Dataset to Assess Large Language Models Telecommunications Knowledge," 2023. [Online]. Available: https://arxiv.org/abs/2310.15051

[9] J. de Arimatéa Passos Lopes Júnior, "3GPP LLM Evaluation," gitHub repository. [Online]. Available: https://github.com/josearimatea/3gpp_llm_eval_light, 2024, accessed: Jan. 2025.

[10] A.-L. Bornea, F. Ayed, A. D. Domenico, N. Piovesan, and A. Maatouk, "Telco-RAG: Navigating the Challenges of Retrieval-Augmented Language Models for Telecommunications," 2024. [Online]. Available: https://arxiv.org/abs/2404.15939

[11] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-Consistency Improves Chain of Thought Reasoning in Language Models," 2023. [Online]. Available: https://arxiv.org/abs/2203.11171

[12] Y. Song, T. Wang, S. K. Mondal, and J. P. Sahoo, "A comprehensive survey of few-shot learning: Evolution, applications, challenges, and opportunities," 2022. [Online]. Available: https://arxiv.org/abs/2205.06743

[13] Unsloth, "Unsloth Repository," GitHub repository. [Online]. Available: https://github.com/unslothai/unsloth?tab=readme-ov-file, accessed: Nov. 2024.

[14] ——, "Unsloth site," Unsloth. [Online]. Available: https://unsloth.ai/introducing, accessed: Jan. 2024.

[15] N. Team, "Dataset TeleQnA," Hugging Face repository. [Online]. Available: https://huggingface.co/datasets/netop/TeleQnA, accessed: Nov. 2024.

[16] R. Nikbakht, "TSpec-LLM repository," Hugging Face repository. [Online]. Available: https://huggingface.co/datasets/rasoul-nikbakht/TSpec-LLM, accessed: Nov. 2024.

[17] R. Documentation, "RAGAS Documentation," Online. Available: https://docs.ragas.io/en/stable/, accessed: Jan. 2025.