# Efficient LLMs for 3GPP Specifications: Fine-Tuning and Retrieval-Augmented Generation

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

Jayr Alencar Pereira
*Unicamp*
Campinas, Brazil
jayr@unicamp.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*—This work evaluates the performance of Large Language Models (LLMs) in answering telecommunications-related questions by leveraging two key datasets: TeleQnA, a domain-specific question-answering dataset, and TSpec-LLM, a comprehensive repository of technical documents from multiple 3GPP (3rd Generation Partnership Project) releases. To enhance response accuracy, we implement a Retrieval-Augmented Generation (RAG) system that integrates relevant context from TSpec-LLM into TeleQnA queries. Additionally, we investigate whether fine-tuning an open-source model on telecommunications-specific text further improves performance. Our evaluation demonstrates that open-source LLMs, when enhanced with RAG and fine-tuning, can achieve performance comparable to proprietary models while requiring significantly less computation. This makes them well-suited for deployment in resource-constrained environments, particularly appealing for softwarized edge infrastructures or even end-user devices.**

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

## I. Introduction

The increasing adoption of LLMs is transforming various fields. LLMs can be employed as agents capable of text-based assistance regarding technical standards, including those found in telecommunications. However, the complexity and continuous evolution of technical standards require models capable of processing specific, extensive, and detailed documents. Evaluating and enhancing LLMs for interpreting technical standards and answering domain-specific questions is a crucial step for ensuring a precise understanding of 3GPP documents by natural language processing agents.

Despite recent advancements [1], [2], LLMs in telecommunications remain largely generalist, lacking specialized knowledge of 3GPP norms, which limits their effectiveness in such a deep technical domain. Additionally, state-of-the-art models are often costly, computationally intensive, and not optimized for domain-specific applications. Studies such as [3] highlight the need for models tailored to telecom-specific terminology and mathematical representations, while [4] emphasizes the importance of dedicated datasets for fine-tuning. Recent research [5] has explored LLM applications for an intent translation system for Intent-Based Networking (IBN), converting natural language intents into structured network service descriptors, streamlining the management of softwarized networks.

Our work aims to contribute to the emerging trend of applying LLM to networking by developing a lightweight, efficient, and specialized LLM for telecommunications. The goal is to improve the comprehension of 3GPP standards while ensuring accessibility for local deployment. To achieve this, we integrate Retrieval-Augmented Generation (RAG) and fine-tuning to enhance LLM performance. RAG improves response accuracy by retrieving relevant external documents, enabling more precise and context-aware answers. Fine-tuning further refines the model by training it on domain-specific data, optimizing its ability to interpret and generate content aligned with telecommunications standards.

The study presented in this paper evaluates multiple LLMs on TeleQnA, a question-answering dataset specifically designed for telecommunications. Additionally, we utilize TSpec-LLM, an extensive collection of 3GPP technical documents covering multiple releases, to implement a RAG system that enriches the context of TeleQnA responses. A comparative analysis is conducted across three distinct LLMs, along with an assessment of whether fine-tuning a smaller model significantly improves performance, as explored in [6].

Ultimately, this work contributes to telecommunications AI research by demonstrating that a smaller, fine-tuned model, when combined with RAG, can achieve high accuracy while significantly reducing computational costs and memory usage. Our results show that domain-specific fine-tuning and retrieval-based augmentation allow lightweight models to outperform larger, general-purpose models in specialized tasks. To promote further research and reproducibility, we openly release all software, datasets, training methodologies, and the fine-tuned model in our GitHub repository [7], providing a valuable resource for future developments in the field.

## II. Related Work

The rapid advancements in LLMs have spurred interest on assessing LLMs' ability to interpret technical documents, particularly 3GPP standards.

Piovesan et al. (2023) introduced TeleQnA [8], a benchmark dataset of 10000 questions designed to evaluate LLMs' knowledge in telecommunications. Covering both general and technical knowledge, TeleQnA has proven to be a valuable tool for assessing LLMs in standards interpretation.

Building on this, Benzaghta et al. (2024) proposed TSpec-LLM [4], a repository of 3GPP documents (Releases 8 to 19) to aid LLM training. This extensive dataset supports retrieval-based methods, such as Retrieval-Augmented Generation (RAG), which enhance response accuracy by providing structured technical content. Furthering this approach, Bornea et al. (2024) introduced Telco-RAG [9], a framework that integrates retrieval mechanisms with language models to improve performance on domain-specific queries, highlighting the importance of tailored methods for processing technical documents.

Finally, Nikou et al. (2024) introduced TeleRoBERTa [6], a Roberta-based model fine-tuned for telecommunications. Their proprietary dataset significantly enhanced model performance on complex 3GPP standards, demonstrating how domain-specific datasets and fine-tuning strategies can improve LLM capabilities in this specialized field.

Our work builds on existing research by addressing key limitations in current LLMs for telecommunications. While studies such as [3] and [4] focus on large, computationally intensive models, and [6] relies on a proprietary dataset, we demonstrate that domain-specific fine-tuning combined with a lightweight RAG system and a relatively small LLM can achieve high performance with significantly lower computational costs. By leveraging publicly available datasets, including TeleQnA and TSpec-LLM, our approach ensures full transparency and reproducibility.

## III. Methodology

The experiments were designed to evaluate the models' ability to select correct answers to questions derived from 3GPP Releases 17 and 18, as well as other undefined releases. The TeleQnA dataset, which contains 200 questions per release, was used as the primary benchmark.

Two types of tests were conducted:

1) **Without contextual information** – The model received only the question without any additional context.
2) **With Retrieval-Augmented Generation (RAG)** – The RAG system supplied relevant information from TSpec-LLM documents to enhance response accuracy.

In both scenarios, models were required to select the correct answer from multiple alternatives. Performance was assessed by calculating accuracy, based on the number of correctly answered questions.

### *A. Models*

Three different models were tested for comparison using the TeleQnA dataset: (*i*) **GPT-4o-mini** model from OpenAI; (*ii*) **Llama 3.2** model with 3 billion parameters from Meta, utilized through the Unsloth library [10], and (*iii*) the same Llama 3.2 model with 3 billion parameters but fine-tuned, referred to as **Llama-4bit-Tuned,** and trained using the TeleQnA dataset [11].

### *B. Models' framework: Unsloth*

Unsloth [12] is an open-source library that optimizes the training, fine-tuning, and inference of LLMs, enabling efficient model execution across diverse hardware configurations. Several key advantages led to its selection for this study:

- Unsloth simplifies the fine-tuning process and optimizes inference, making it particularly suitable for models like Llama.
- It supports 4-bit quantization, which significantly reduces memory consumption while maintaining performance.
- Unsloth offers seamless integration with Python's CUDA libraries for GPU acceleration, facilitating easier and more efficient loading and running of models.
- It is compatible with both local environments and cloud platforms like Google Colab, enhancing its flexibility.
- Unsloth streamlines the downloading and loading of models into GPU memory, leading to faster execution and smoother integration with deep learning frameworks.
- It also integrates with Hugging Face's Transformers and TRL libraries, supporting features like gradient checkpointing for handling long contexts and mixed precision training (FP16 or BF16). These optimizations contribute to faster training times and lower memory consumption.

In this study, Unsloth was used to fine-tune the ‘`unsloth/Llama-3.2-3B-Instruct`‘ model by implementing LoRA layers for parameter-efficient training on the TeleQnA dataset. The library also enabled the efficient loading of the quantized model onto the GPU, optimizing inference with reduced latency. These capabilities allowed the model to process and interpret technical questions from 3GPP specifications more accurately, all while operating within limited computational resources. The fine-tuning process used 4,000 questions from the TeleQnA dataset, with different context and response pairs for the first and last 2,000 questions, which helped refine the model's ability to interpret and generate content aligned with telecommunications standards. The model, referred to as Llama-4bit-Tuned, was trained in 16-bit precision on Google Colab due to hardware limitations and then loaded onto a personal computer with 4-bit quantization for inference testing.

### *C. Fine Tuning*

The fine-tuning of the Llama 3.2 model with 3 billion parameters followed the structure outlined in [10].

**Training Dataset**. The data used for training came from the TeleQnA dataset [11], in which 4000 questions were utilized. Of these 4000 questions, 500 were related to Release 17 (leaving 233 for testing), while the remaining 3500 questions were about other Releases, excluding Releases 17 and 18.

For the training, two different pairs of context and expected response were provided:

In the first 2,000 questions, the pairs were:

- Context: question and options
- Expected Response: the correct option and the text of the correct option

In the last 2,000 questions, the pairs were:

- Context: only the question
- Expected Response: only the text of the correct option

Finally, the questions were shuffled.

**Parameters and Training.** The training code can be found in [7], under the path '`/Source/Fine_tuning/`'. Due to personal hardware limitations (the local GPU is an NVIDIA GeForce RTX 3050 Ti with a maximum memory of 3.712 GB), the model training was conducted on Google Colab. This setup aimed to load the model without 4-bit quantization and train it with more questions in less time.

The model quantized to 4 bits occupies 2.768 GB of GPU memory, whereas the version loaded from Unsloth without quantization requires 6.688 GB, reaching a peak memory usage of 8.658 GB during training with 4000 questions in 16-bit precision. After a 42-minute training process, the resulting model, named **Llama-4bit-Tuned**, was evaluated using the TeleQnA dataset. The name reflects both the 4-bit quantization applied for compatibility with local hardware and the fine-tuning process performed to enhance its performance.

The training parameters were:

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

### *D. Retrieval Augmented Generation (RAG)*

RAG is a retrieval-based enchantment of LLMs that searches for relevant excerpts in a specific dataset or document collection, providing the language model with additional context to enhance the accuracy and relevance of the generated response.

The documents used for our information retrieval were from the TSpec-LLM Dataset [13]. First, the documents were divided using the `MarkdownHeaderTextSplitter`, creating 780,651 chunks, each with a size of 2,000 characters and a 100-character overlap between chunks. The Embeddings model used was a `SentenceTransformer` model: `all-mpnet-base-v2`. Subsequently, the Embeddings were indexed using Faiss, which was later used to perform searches based on cosine similarity. The input for RAG searches consisted of questions with options for accuracy tests.

## IV. Evaluation

GPT-4o-mini, Llama 3.2 3B, and Llama-4bit-Tuned were evaluated on 600 multiple-choice questions from the TeleQnA dataset, comprising 200 questions each from Releases 17 and 18, and 200 from other releases. Using a Chain of Thought [14] 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.

### *A. Prompts*

Evaluations were conducted with and without Retrieval-Augmented Generation (RAG). The RAG condition provided additional context retrieved from the TSpec-LLM dataset. In the non-RAG condition, 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.

Furthermore, the substantial performance difference between the fine-tuned Llama model and its non-tuned counterpart underscores the effectiveness of the fine-tuning process.

| **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.

These results suggest that fine-tuning, especially in conjunction with RAG, significantly improves model accuracy, particularly for 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 1 presents a consolidated accuracy comparison for the GPT-4o-mini, Llama 3.2 3B, and Llama-4bit-Tuned models. The results includes the calculated confidence intervals for each model's accuracy under both 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.

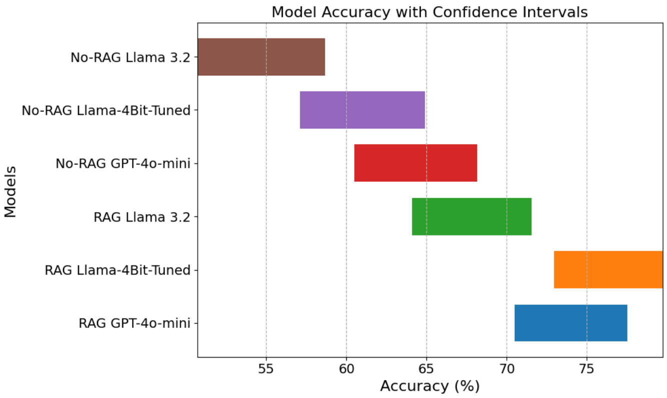

Fig. 1: Consolidated accuracy results with 95% confidence intervals for all 600 questions across the three models (GPT-4o-mini, Llama 3.2 3B, and Llama-4bit-Tuned) under RAG and non-RAG conditions.

### C. Discussion

The evaluation of the models demonstrated significant performance differences, both with and without RAG, across all 600 questions. With RAG, the models achieved higher accuracy by leveraging additional contextual information from the TSpec-LLM dataset. Among the models tested, Llama-4bit-Tuned showed the best overall performance, achieving an accuracy of 76.33% ([72.93%, 79.73%]), surpassing GPT-4o-mini, which attained an accuracy of 74.00% ([70.49%, 77.51%]). The base Llama 3.2 3B model achieved an accuracy of 67.83% ([64.10%, 71.57%]) under the same conditions.

Without RAG, the performance of all models dropped significantly. In this condition, GPT-4o-mini achieved an accuracy of 64.33% (60.50%, 68.17%]), while Llama-4bit-Tuned obtained 61.00% ([57.10%, 64.90%]) and Llama 3.2 3B reached 54.67% ([50.68%, 58.65%]). These results highlight an important finding: RAG serves as a key differentiator by retrieving external information relevant to the query, enabling the model to generate more coherent and accurate responses. The absence of RAG leads to a substantial drop in performance, especially for smaller models, which have inherently limited knowledge due to fewer parameters, reduced pretraining capacity, and a lower ability to generalize across diverse topics.

The analysis of confidence intervals reinforces the statistical significance of these results. With 95% confidence, the accuracy range for Llama-4bit-Tuned (72.93%–79.73%) remains entirely above that of Llama 3.2 3B (64.10%–71.57%) when both use RAG. This confirms that the fine-tuning process effectively improved the Llama model's performance, as there is no overlap in their confidence intervals, demonstrating a statistically significant enhancement.

Furthermore, the accuracy intervals of Llama-4bit-Tuned and GPT-4o-mini are very close, Figure 1 suggests that, statistically, the two models perform similarly. This finding highlights the effectiveness of fine-tuning in reducing the performance gap between an optimized smaller model and a larger, general-purpose model.

Fine-tuning proved particularly effective for small models like Llama-4bit-Tuned, demonstrating that domain-specific adaptation significantly enhances performance. Despite being a smaller, quantized model, Llama-4bit-Tuned outperformed GPT-4o-mini in accuracy when RAG was employed. This finding underscores the potential of combining fine-tuning with retrieval-based methods to optimize performance in domain-specific applications.

These results also emphasize that applications requiring technical or specialized knowledge do not necessarily depend on large and computationally expensive models. Smaller, fine-tuned models can achieve similar or even superior performance, particularly when coupled with retrieval mechanisms like RAG. Additionally, 4-bit quantization in smaller models enabled efficient inference without significant accuracy degradation, reinforcing its viability for deployment in resource-constrained environments.

Finally, it is noteworthy that the inference time of Llama-4bit-Tuned was comparable to that of GPT-4o-mini, demonstrating that smaller models with fine-tuning can be competitive not only in accuracy but also in computational efficiency. This suggests a promising direction for further research into optimizing small, specialized models for practical deployment.

## V. Concluding Remarks

This work demonstrated that training on a specific domain enhances a model's ability to interpret and understand specialized topics, which, when combined with external retrieval, further improves performance. This synergy allows smaller models to effectively extract and process external information, compensating for their inherent limitations. Consequently, the combination of RAG with fine-tuning for small models may represent a new paradigm of efficiency for specialized tasks.

Our results confirm that 4-bit quantization maintains high performance while enabling broader accessibility, even when dealing with complicated and specific technical standards. The ability to deploy models on lower-capacity hardware facilitates the widespread adoption of AI, making it more feasible for local execution on cost-effective devices. This proves that AI models can be effectively utilized in applications with low computational resources, extending their applicability to a broader range of users and scenarios.

At the same time, this work showed that fine-tuning improves the performance of smaller models, yet their intrinsic limitations from pre-training still prevent them from reaching the performance levels of larger models, which undergo more extensive and effective training. The primary performance gain with RAG stems from its ability to provide external information, reducing the model's reliance on extensive internal knowledge and instead shifting the focus to interpreting and extracting relevant information. This demonstrates that fine-tuned small models with strong interpretative capabilities benefit significantly from the combination of external information retrieval and domain-specific training for understanding the 3GPP technical specifications.

**Limitations & Future Research.** Despite the valuable insights, several limitations could be addressed in future work:

1) The evaluation was conducted on questions with options, and future studies should explore model performance when no options are provided. It would be important to develop relevant metrics to compare the answers generated by the models in such scenarios. The use of RAGAS metrics [15] could be particularly useful for assessing the quality of responses in such cases.
2) Another limitation is the relatively small dataset used for fine tuning, consisting of only 4,000 questions. Future work should investigate whether larger datasets would lead to improved model performance, as additional data could help the model generalize better to diverse queries.
3) Hardware limitations also prevented testing larger models, such as the Llama 8B, to evaluate whether they would provide better performance. This represents a future research avenue for those with more powerful hardware, which could lead to improved performance by leveraging larger models.
4) This work was constrained by cost, as only a few models were tested, and the comparison was limited to GPT-4o-mini. Comparing the fine-tuned Llama models to larger models like GPT-4 could provide further insights, but the associated costs would be significant. Nonetheless, demonstrating that a larger model can still be competitive with a fine-tuned smaller model would be a valuable direction for future research.
5) Future studies could explore the use of Agents or Multi-Agent systems to enhance model interactions. For instance, allowing a model to interact more frequently with the same query and retrieval tool might improve responses, particularly for questions that the models answered incorrectly or those where the RAG search did not yield satisfactory results. A more comprehensive search process could help in providing better answers to such questions.

## References

[1] 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

[2] 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

[3] 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

[4] 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

[5] 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.

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

[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] 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

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

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

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

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

[14] 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

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