Using Large Language Models to Understand Telecom Standards

Athanasios Karapantelakis, Mukesh Thakur, Alexandros Nikou, Farnaz Moradi,
Christian Olrog, Fitsum Gaim, Henrik Holm, Doumitrou Daniil Nimara, Vincent Huang
Ericsson AB, Torshamnsgatan 21, 16483, Stockholm, Sweden
{athanasios.karapantelakis, mukesh.thakur, alexandros.nikou, farnaz.moradi, christian.olrog,
fitsum.gaim.gebre, henrik.holm, doumitrou.nimara, vincent.a.huang}@ericsson.com

***Abstract*—The Third Generation Partnership Project (3GPP) has successfully introduced standards for global mobility. However, the volume and complexity of these standards has increased over time, thus complicating access to relevant information for vendors and service providers. Use of Generative Artificial Intelligence (AI) and in particular Large Language Models (LLMs), may provide faster access to relevant information. In this paper, we evaluate the capability of state-of-art LLMs to be used as Question Answering (QA) assistants for 3GPP document reference. Our contribution is threefold. First, we provide a benchmark and measuring methods for evaluating performance of LLMs. Second, we do data preprocessing and fine-tuning for one of these LLMs and provide guidelines to increase accuracy of the responses that apply to all LLMs. Third, we provide a model of our own, TeleRoBERTa, that performs on-par with foundation LLMs but with an order of magnitude less number of parameters. Results show that LLMs can be used as a credible reference tool on telecom technical documents, and thus have potential for a number of different applications from troubleshooting and maintenance, to network operations and software product development.**

***Index Terms*—Large language models, telecom, 3GPP**

I. Introduction

The Transformer model architecture, proposed in 2017, enabled faster training of Machine Learning (ML) sequence models than previous architectures, such as Recurrent Neural Networks (RNNs) and in particular Long-Short Term Memory networks (LSTMs) [1]. In the field of Natural Language Processing (NLP), Transformer-based Large Language Models (LLMs) were shown to perform well in text-based applications such as language generation, knowledge utilization, complex reasoning, structured data generation and information retrieval [2]. Applications that use LLMs, are based in so-called "foundation models", such as OpenAI's Generative Pretrained Transformer (GPT) [3] and Meta's Llama [4] family of models. Foundation models are designed to generate a wide and general variety of outputs and can be used for standard tasks such as summarization, completion, Question Answering (QA) and code generation. They have billions of parameters and are trained on corpora such as Common Crawl [5].

For foundation models to adapt to application domains, such as telecommunications, prompt engineering and fine-tuning approaches are used. Prompt engineering partially relies on "in-context learning", where models learn by using examples as context [6], and partially on guidelines such as patterns and templates for prompt authoring that more accurately describe the task a model should generate outputs for [7]. Prompts are provided to models as "embeddings", i.e., numerical representations organized in vectors that indicate commonality between words in prompts and provide a format required for further processing at the LLM itself.

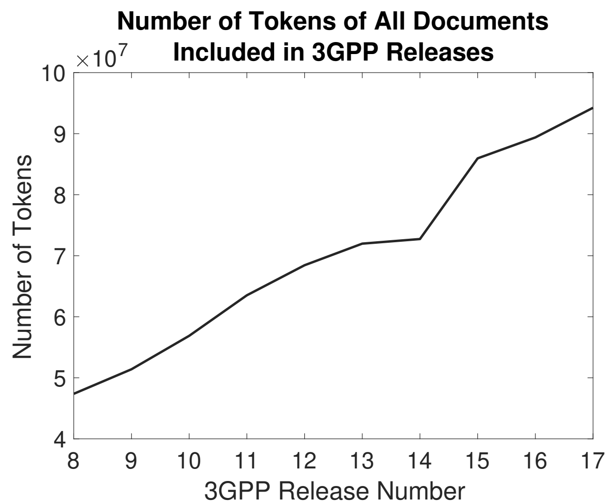

Fig. 1. Increase of number of tokens (words) for 3GPP releases, between Release 8 (2006-01-23) and Release 17 (2018-06-15). At the moment of writing of this paper, Releases 18 and 19 still have an "Open" status, meaning that material is being added to the documents comprising those Releases.

While prompt engineering controls the output of a model, fine-tuning can be used to add knowledge in a specific area that may not be covered sufficiently by initial training data. Fine-tuning techniques include offline learning approaches, such as Supervised Fine-Tuning (SFT), which uses an already available labeled dataset, and online learning approaches such as Reinforcement Learning with Human Feedback (RLHF), which uses human preferences as a reward function [8].

In this paper, we investigate whether LLMs can be used as digital assistants for Third Generation Partnership Project (3GPP) standards. By default, these standards are created to be human-readable and have been increasing in size as newer generations of mobile networks have additional functionality when compared to their predecessors (see Figure 1). This means that it is progressively more difficult for human readers, but also machines, to find relevant information in these standards.

This paper contributes to the state of the art by determining the capabilities and limitations of the current generation of LLM when it comes to referencing 3GPP standards. Given that all telecom standards follow the same natural language document structure, we envision the applicability of this paper to go beyond 3GPP family of standards and also generally apply to other standardization bodies, such as TeleManagement Forum (TMForum), European Telecommunication Standards (ETSI), Open Radio Acess Network (O-RAN) and International Telecommunication Union (ITU).

The contribution of this paper to state of the art (SoA) is threefold:

- Evaluation of performance of SoA foundation LLMs for QA of 3GPP documents using both statistical and reference metrics.
- Introduction of TeleRoBERTa, an extractive QA model, that performs as well as the highest-performing foundation models, while having fewer parameters.
- Data preparation methods for 3GPP documents to reduce hallucinations and increase model perfrormance. Given that telecom standards follow similar structure, methods proposed can be applicable to other families of standards including those published by TeleManagement Forum (TMForum), European Telecommunication Standards Institute (ETSI), Open Radio Access Network (O-RAN) and International Telecommunication Union (ITU).

This paper is structured as follows: In section II, we describe the background, including related work and determine the gap in SoA followed by an overview of studies LLMs including TeleRoberta. In section III, we describe the architecture of the system we built to evaluate the models and describe our methodological approach. In section IV, we describe the experiments we carried out to evaluate model quality. The section includes a description of insfrastructure, benchmark dataset and key performance indicators (KPI) used in the assessment, and ends with identifying capabilities and limitations of models. Finally, in section V, we conclude by recapitulating our key findings and highlighting areas of future research.

II. Background

A. Related Work

Generative AI (GAI) and LLMs have been successfully utilized specialized domains including telecom. A comprehensive survey on GAI including LLMs for mobile networks is presented in [9], identifying several use cases, including customer incident management, configuration management, billing plan generation and exposure to third parties. It is envisioned that LLMs will open up a new era for wireless networks and LLM-based intelligence will become ubiquitous in the future networks. Several recent studies have looked into the opportunities, application, architectural impacts, and challenges that for generative AI and LLMs in telecommunication networks [10]–[13].

The applicability and effectiveness of existing LLMs has also been studied in the literature. In [14], the applicability and effectiveness of LLMs for translating high-level policies and requirements specified in natural language into low-level network Application Program Interfaces (APIs) is been explored. The authors in [15] studied the adaptability of LLMs to telecom domain by fine-tuning a number of LLMs models (BERT, distilled BERT, RoBERTa, and GPT-2) using 3GPP technical specification data from different working groups. The different models were evaluated on a text classification task to determine the 3GPP specification categories with the corresponding working group. The results indicate the applicability of fine-tuned LLMs to the telecom domain. While promising, the paper has focused on a single classification task and the generative aspects of the models are not studied for understanding telecom standards. Using LLMs for conversational assistance and question-answering within the telecom domain has also been explored. In [16], the capabilities, as well as the limitations of LLMs for conversational assistance related to enterprise wireless products were analyzed. The authors performed experiments using GPT-4, GPT-3.5, Bard (based on LaMDA) and LlaMA models to address different research questions related to domain QA, product QA, context continuity, and input perturbations caused by spelling errors. The responses generated by LLMs were evaluated using subjective metrics including Mean Opinion Score (MOS) and inter-rater agreement. Based on the observations, the authors conclude that publicly available LLMs without fine-tuning are not usable for enterprise use cases. However, the paper does not evaluate any fine-tuned models.

In [17], the authors studied the adaptability of a BERT-like model to the telecom domain. In this study, a small model was selected due to its resource and training time efficiency and was pre-trained using general data and then adapted to the telecom domain using 3GPP technology specification files. The authors also developed a benchmark dataset for telecom-specific QA (TeleQuAD) to fine-tune and evaluate the models.

In [18], the authors studied multi-hop neural question answering in the telecom domain by adapting neural open-domain question answering systems. Several benchmark datasets, including TeleQuAD and mTeleQuAD, to evaluate telecom QA task were introduced using data which is mainly collected from 3GPP specifications.

In [19], another benchmark dataset for telecom domain (TeleQnA) was introduced and used for evaluation of GPT-3.5 and GPT-4. The results suggest the need for specialized foundation models, trained or fine-tuned on thelecom data, to enable answering complex telecom questions.

While the potential of using NLP and LLMs for question answering in telecom domain has been considered before, the evaluation of current generation of LLMs to reference 3GPP standards and the impact of fine tuning has not been fully explored before.

B. LLM Architectures

LLMs typically refer to Transformer-based language models which have tens, or hundreds, of billions of parameters [2]. The transformer architecture which was first presented in [20] has been widely used in different fields including computer vision and audio processing and has revolutionized the field of NLP as the go-to architecture [21]. A transformer is a sequence-to-sequence model consisting of encoder and decoder stacks. Each encoder consists of a number of identical layers where each layer has a multi-headed self-attention module and a position-wise fully connected feed forward neural network. The decoder similarly consists of a number of identical layers. The decoder additionally performs multi-head attention over the output of the encoder stack.

The transformer architecture was adapted by OpenAI and they released the first GPT model (GPT-1) in 2018 [22]. They showed that unsupervised pre-training of a language model followed by discriminative fine-tuning on specific tasks can lead to large gains. GPT-2 [23] which has 1.5 billion parameters used a similar architecture with some modifications. GPT-3 was introduced in [24] as an autoregressive language model with 175B parameters which was trained with a filtered Common Crawl dataset and was utilized in a few-shot, one-shot, or zero-shot way. GPT-3 used the same architecture as GPT-2 except that alternating dense and locally banded sparse attention patterns were used in the layers of the transformer. OpenAI has also released a set of GPT-3.5 models which are capable of understanding and generating natural language and code, including GPT-3.5 turbo which is optimized for chat. The latest LLM released by OpenAI is GPT-4 [3] which is siad to have about 1.76 trillion parameters and is a multimodal model that can accept image and text inputs and producing text outputs.

In contrast to GPT LLMs, the Meta's LLaMA models [4], [25] are open-source, and use data from a mixture of publicly available sources. In July 2023, LLaMA-2 [4] was released, with versions from 7 to 70 billion parameters. LLaMA-2 has some architectural differences compared to LLaMA-1 including increased context length and grouped-query attention (GQA).

Falcon is an LLM with 180 billion parameters which is trained on the RefinedWeb dataset [26]. The Falcon family of LLMs is also based on the transformer architecture and uses multiquery attention for improved scalability. At the time of writing, Falcon is the largest openly available LLM which is shown to outperform LLaMA-2 70B model and GPT-3.5 on Massive Multitask Language Understanding tasks.

The above models belong to generative QA category, wherein new text is generated based on previous knowledge. Another class of models are the extractive QA models, wherein text is extracted and quoted as is from prior knowledge. In the extractive question-answering setup, the encoder type models such as BERT [27] and RoBERTa [28] have demonstrated significant progress in the past years. Performing on par with estimated human performance on large-scale benchmarks such as SQuAD [29], [30]. In the context of telecom, TeleRoBERTa [17], [18] was introduced as adaptation of the RoBERTa base model to the characteristics the telecommunications domain, trained on a large corpus of text collected from in-domain sources such as 3GPP specification.

TABLE I
LLMS USED FOR THE EXPERIMENTS

Model	Creator	License type	Parameters
GPT-3.5-Turbo	OpenAI	Proprietary	175B
GPT-4	OpenAI	Proprietary	undisclosed
LLaMA-2	Meta	LLaMA-2.0	7B, 13B, 70B
Falcon	Technology Innovation Institute	Apache 2.0	180B
TeleRoBERTa	Ericsson	Proprietary	124M

Table I summarises the different LLM that were considered in this paper.

III. METHODOLOGY

This section describes the methods used to design and develop the system along with its architecture. Conventionally, foundation models are pre-trained with large dataset with point-in-time to ensure their effectiveness. However, there may be instances where it becomes necessary to work with one's own/new dataset. In such cases, two approaches can be used: fine-tuning, i.e., further training of the foundation model with the new dataset for increased accuracy of response, within specific domains, and Retrieval Augmented Generation (RAG) [31] with which the user can feed its own dataset to the foundation model and guide the model response in real time through queries (prompts), enhancing the overall model response. In another context, in specific use cases that LLMs have to be loaded to devices with limited resources, quantization methods can be used, as will be described in Section III.B.

A. Retrieval Augmented Generation

Typically, A RAG system comprises four core components. A document processor that parses different document types and splits a lengthy document into smaller sizes or chunks, which is crucial for enhancing the relevance of content retrieval in response to queries. Additionally, there exists a limit on number of tokens a foundation model can process per operation. Another key component is an embedding model that creates vector representations of the document, which capture the semantic meaning of text, allowing to quickly and efficiently find other pieces of text with similar content. This vector representation created by the embedding model is simply called embedding. Another component is a type of database that can efficiently store and search for these embeddings. This database is called the vectorstore. Lastly, a retriever takes the user's query, which might have context, and semantically searches the vectorstore for similar results. A user query with context is referred to as a prompt.

Prompts can be zero-shot, where a certain instruction is prepended to the user query without providing the model with any direct examples. This makes sure the model focuses on providing desired answers. Few-shot prompting refers to the case where a few examples are prepened to the user query for a more accurate response. Lastly, chain-of-thought prompting, allows for detailed problem-solving by guiding the models through intermediate steps. In this work, we have used zero-shot prompt to get precise answers.

B. Quantization methods for LLMs

In this section, we describe two quantization methods used for faster inference in LLMs. Namely, the Georgi Gerganov's Machine Learning (GGML) and GPT-Generated Unified Format (GGUF) [32]. GGML and GGUF are two quantization methods used in large LLMs to reduce the precision of the model's weights and activations from floating-point numbers to integers. GGML was an early attempt to create a file format for storing GPT models. It allowed models to be shared in a single file, making it convenient for users since it could be run on CPUs, which made them accessible to a wider range of users. On August 21st 2023, GGML was replaced by GGUF which aims to address the limitations of GGML and improve the overall user experience, by offering more flexibility, extensibility, and compatibility with different types of LLMs. For the experimentation of the paper in hand, the GGUF quantization was adopted.

C. Fine-tuning

For fine-tuning we used the SFT method, which uses a labelled dataset with domain data in order to guide the model to a specific task. In the context of this work, the task was to provide QA reference to 3GPP standards, therefore the questions revolved around content included in those standards. Of specific interest, is content that can be misinterpreted by the LLM, such as acronyms or technical jargon, that may also overlap with prior knowledge from another domain that the LLM was trained with. We discuss more details on the fine-tuning approach in section IV-C.

D. System Architecture

In order to interact/query with dataset that is 3GPP documents, we did fine-tuning of a LLM model and also designed and developed a system based on the RAG. This system is *TelcoGenAI*. The architecture of *TelcoGenAI* is illustrated in Figure 2. The main component of the *TelcoGenAI* is *QueryEngine*. This component is based on Langchain [33] framework, an open source framework, designed to create LLM applications. The *QueryEngine* utilizes four main langchain features: document loader, text splitter, embedder and retriever [34]. The document loader, loads the 3GPP specification word/pdf documents. The text splitter, splits the loaded documents into designated chunks sizes (e.g., 1000) with overlap (e.g., 100). Overlap ensures there is context between chunks. The embedder ingests these chunks and creates embeddings using specified LLM, subsequently storing them to one of the popular vectorstore, Facebook AI Similarity Search (FAISS) [35]. The retriever, accepts a user query along with context forming a prompt and generates prompt embeddings using the specified LLM. These embeddings are used to search the vectorstore for the most relevant matches. The top matches and prompt embeddings both, is then sent to the LLM, which generates a response for the user.

IV. Evaluation

A. Introduction

In this section, we present our experiments and results. The models we used for our experiments are illustrated in table I. With the exception of GPT 3.5-Turbo, which uses OpenAI's API, all other models were downloaded and experimented with locally. The reason for selecting to experiment with the aforementioned models out of a plurality of models was twofold. First, we wanted to have a mixture of models of varying complexity, that could potentially be executed in either standard off-the shelf servers, or require dedicated Graphics Processing Unit (GPU) clusters, to address different use-cases and budget. Second, we wanted to have models that could be executed locally as well as models that could be interfaced with remotely. The former could potentially allow prompting or fine-tuning with proprietary information that cannot be disclosed to a third-party infrastructure, via an API.

We have completed two sets of experiments:

- The first set is an evaluation of selected SoA foundation models of varying complexity, as well as the TeleRoBERTa model, using a benchmark we developed in-house known as TeleQuAD [17]. The benchmark is one of the first datasets for 3GPP standards and contains over 4,000 question and answer pairs that can be used for evaluating performance of LLMs. The foundation models were prompted with a template, and embedded 3GPP specifications as context.
- The second set uses learnings from the first set to construct a labeled dataset from both 3GPP specification and other sources, such as content from web sites. This dataset was then used to further train a foundation model using the SFT approach.

To measure performance, we use two types of metrics. First, BERTScore [36], which measures similarity of generated answers of LLMs with TeleQuAD. The process includes transforming the natural language of generated and reference answers to embeddings in vector space using a pretrained embedding model. Next, these vectors are compared using a pairwise cosine-similarity approach. Second, we use GPT-4 to evaluate generated answers of LLMs, using the *QAEvalChain* functionality of langchain. GPT-4 is arguably the most capable LLM to date, and was used to verify whether correctness of the LLM-generated answer by comparing it against a reference answer from TeleQuAD. We refer to this metric as "GPT-4 Ref score".

B. Evaluation of Foundation Model Performance

For this set of experiments, we evaluated the performance of the five LLMs mentioned in section IV-A against TeleQuAD. For local models, we used C++ ports instead of the original

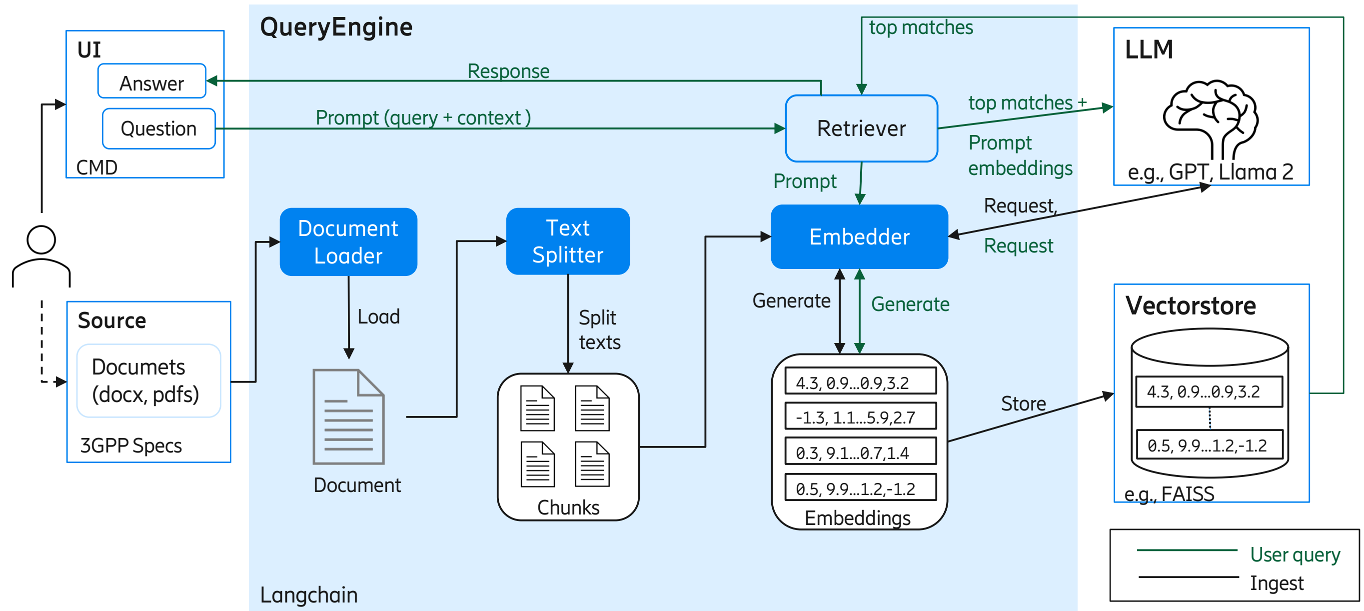

Fig. 2. Architecture of TelcoGenAI

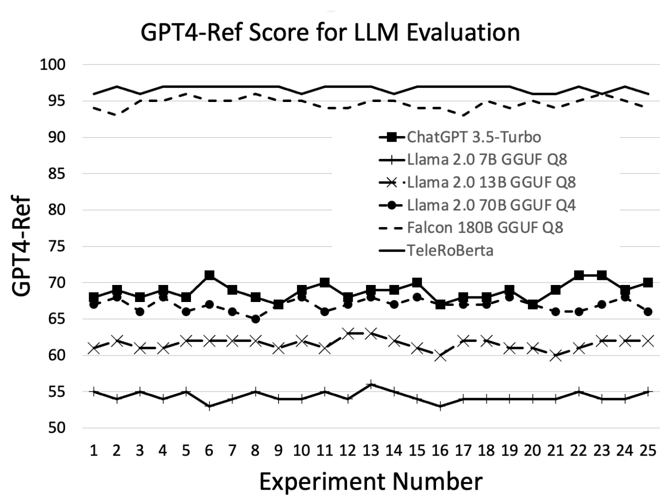

Fig. 3. Performance of foundation LLMs, using TeleQuAD dataset as ground truth and GPT-4 as reference.

model, and 8-bit (Q8) and 4-bit (Q4) introduced under the GGUF format by llama.cpp [37]. The advantage of using such an approach was that we were able to perform inference on Llama and Falcon model variants on commodity hardware, using system random access memory (RAM) instead of GPU memory and without sacrificing model performance. In order to verify performance consistency, the experiments were performed for every LLM 24 times. Figure 3 shows the results for GPT-4 Ref.

We observe that TeleRoBERTa and Falcon 180B performed similarly, answering 92 to 96% of all questions correctly, whereas models with lower number of parameters had lower performance. These measurements were also mirrored when using BERTScore, as shown in Figure 4. The small differences can be attributed to the way the BERTSCore is calculated, which is based on the distance between tokens of answers, which can themselves be based on semantics between words (such as categories they belong to, to establish relative meaning). On the other hand, GPT-4 relies on the large corpus of knowledge it was trained with to make that decision. Another observation is that the statistical dispersion among 24 runs is small, as illustrated in Figures 4 and 3. This means that the performance figures for the LLMs are consistent and therefore can be trusted.

For comparison, we evaluated the TeleRoBERTa model using the machine reading comprehension approach [29]. Given a context paragraph and a question, the model predicts the answer span in the context. This is a different formulation with how the generative LLMs were evaluated, and yields insightful results for comparison. Before evaluation, the model is fine-tuned on a combination of SQuAD-v2 and TeleQuAD with training samples of 100K and 3K, respectively. The training follows a similar approach as discussed in [27], [28]. We observe that the model performs on par or better than the much larger models, such as Llama and GPT-3.5 Turbo, while having only 125M trainable parameters. This is an indication that domain-adapted smaller models can be very competitive on specific tasks. However, it is important to recognize that the benchmark used in this work was annotated for the extractive QA, and hence verbatim answer spans from context will inherently match closer to the ground truth answers. Furthermore, the generative LLMs tend to phrase the answers with variance and extra explanations that may affect the computed scores. For benchmarks that require multi-source content aggregation and complex reasoning, the generative LLMs are expected to perform better than their encoder

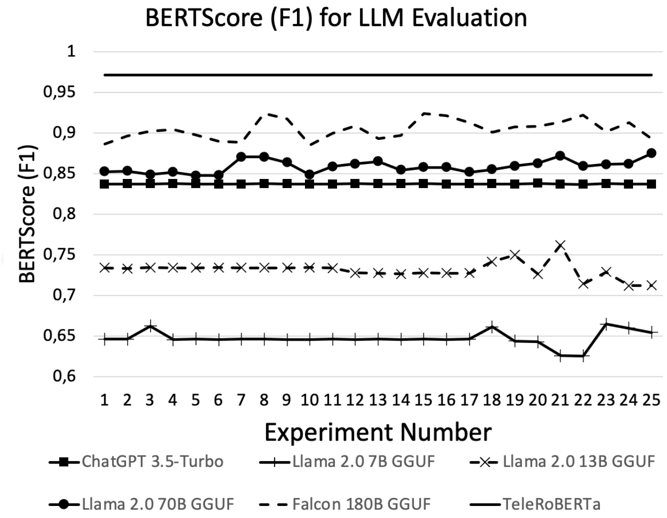

Fig. 4. Performance of foundation LLMs, using BERTScore and TeleQuAD dataset as ground truth.

Table 16C: F-DPCH fields

Slot Format #i	Channel Bit Rate (kbps)	Channel Symbol Rate (ksps)	SF	Bits/ Slot	N_{OFF1} Bits/Slot	N_{TPC} Bits/Slot	N_{OFF2} Bits/Slot
0	3	1.5	256	20	2	2	16
1	3	1.5	256	20	4	2	14
2	3	1.5	256	20	6	2	12
3	3	1.5	256	20	8	2	10
4	3	1.5	256	20	10	2	8
5	3	1.5	256	20	12	2	6
6	3	1.5	256	20	14	2	4
7	3	1.5	256	20	16	2	2
8	3	1.5	256	20	18	2	0
9	3	1.5	256	20	0	2	18

Slot Format 0 of Fractional Downlink Dedicated Physical Channel (F-DPCH), has 3 kbps channel bit rate, 1.5 ksps channel symbol rate, 256 spreading factor (SF), 20 bits per slot, 2 bits for the first off period (N_{OFF1}), 2 bits for Fractional Transmitted Precoding Indicator Channel (F-TPICH) fields (N_{TPC}) and 16 bits for second off period (N_{OFF2}).

Slot Format 1 of Fractional Downlink Dedicated Physical Channel (F-DPCH), has 3 kbps channel bit rate, 1.5 ksps channel symbol rate, 256 spreading factor (SF), 20 bits per slot, 4 bits for the first off period (N_{OFF1}), 2 bits for Fractional Transmitted Precoding Indicator Channel (F-TPICH) fields (N_{TPC}) and 14 bits for second off period (N_{OFF2}).

[…]

Fig. 5. Table transformation: Exemplary table from a 3GPP specification (top) and replacement text (bottom).

counter parts – this exploration is left for future work.

When examining the incorrect responses from different LLMs, we found some common denominators that pointed to potential issues with regards to how the embedded context was represented, but also lack of potential background knowledge. Specifically we noticed the following:

- Technical jargon and abbreviations were misinterpreted by LLMs, leading them to provide answers that were irrelevant to the meaning of the question. This phenomenon is also known as "bias". An example was a question about Home Subscriber Server (HSS), which was interpreted by an LLM as "Home Security Solution".
- Information organized in tables was particularly hard for LLM to identify. For example, several technical standards have a list of what are the mandatory and optional parameters for API requests. When querying the model whether a parameter was mandatory or optional, the network could not identify the cell in the table were this parameter was located, and provide a seemingly random answer that differed accross inferencing sessions. Such phenomena, where an LLM seemingly "guesses" the information due to lack of knowledge or its inability to process knowledge, is known as "hallucinations".
- Cross-reference of information in documents was also difficult for LLMs to follow. For example, such information may be contained as reference to another 3GPP standard or another source of information,

In the next section, we discuss the approach we took to work around these issues.

C. Context Engineering and Fine-tuning

In order to address the issues mentioned in section IV-B, we carried a small-scale experiment on a small subset of TeleQuAD dataset. This subset contained a number of QA pairs that demonstrated these issues. Prior to fine-tuning, we prepared the 3GPP documents used as context. Specifically, we replaced the tables with natural language. We have also replaced the abbreviations in text with an explanation of the abbreviation. This was done to reduce the bias mentioned in the previous section. Figure 5 illustrates an example.

TABLE II
PERFORMANCE OF FINE-TUNED MODEL VERSUS BASELINE

Model Name	Method	BERTScore	GPT-4 Ref
Llama 2.0 7B	Prompted, Contexted with raw data	0,674382848	56
Llama 2.0 13B	Prompted, Contexted with raw data	0,774348293	65
Llama 2.0 7B	Fine-tuned, prompted, contexted with processed data	0,783238483	64

In addition, we included all documents referenced by 3GPP documents as context to help reduce hallucinations. To reduce bias, we also performed fine-tuning. Specifically, we created a labeled dataset of question and answer pairs comprising information that could be overlapping with irrelevant data that the foundation model was originally trained with. For example, we asked simple questions such as "What is HSS?", and answer with "Home Subscriber Server", in order to help guide the model towards the telecom definition of HSS. For fine-tuning we used LlaMA 2 with 7B parameters and trained it using SFT. Results are shown in table II. The BERTScore and GPT-4 Ref score are averages of 24 runs of the three models on the limited test dataset, hence the slight diversion from the data presented in Figures 3 and 4.

The results from the evaluation show an approximate

16% increase in performance of the fine-tuned and properly contexted model, over the baseline LlaMA 2 7B. These performance figures are on-par with the baseline of LlaMA 2 13B, which also means that LLMs with smaller number of parameters - and thus lower computational and storage requirements can offer comparable levels of performance when fine-tuned to larger LLMs.

V. Conclusion

In this paper we investigate the capabilities and limitations of state-of-the-art LLMs as QA assistants for telecom domain.

To assist users to access relevant information faster from ever growing information source of 3GPP specifications, we introduce *TelcoGenAI*, a platform that provies access to difference LLMs.

We also introduce *TeleRoBERTa*, an extractive QA LLM, and compare its performance with the state-of-art foundation generative QA LLMs, such as GPT 3.5-Turbo. For comparison, we use *TeleQuAD*, a benchmark containing QA pairs based on 3GPP standard content. We use two types of metrics to measure accuracy of produced answers, namely BERTScore, and GPT-4 Ref. Whereas the former method is based on statistical evaluation of distance between the reference and produced answer to a question, the latter is based on the use of GPT-4, arguably the highest performing LLM, to evaluate whether a produced answer is similar to the reference answer.

Results not only show that *TeleRoBERTa* performs on-par with the state-of-art foundation LLMs that have an order of magnitude more parameters, but also that the accuracy is consistently high enough for these LLMs to be used as credible digital assistants for 3GPP standards reference. Results also show that through pre-processing of prompt context and use of SFT, accuracy can be further improved.

Establishing a baseline set of LLMs that perform well in 3GPPs specification QA opens the way for many interesting applications, from field service operations such as troubleshooting, commissioning and upgrading of radio base stations, to customer incident management at a Network Operations Centre (NOC) and is part of future work. In addition, as the number of APIs in mobile networks keeps increasing, given that functionality is further compartmentalized and virtualized, software development-related tasks such as genreation of API calls is an interesting area to explore.

Acknowledgment

The authors would like to acknowledge Dr. Ioanna Mitsioni for her review of the paper prior to submission.

References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: http://arxiv.org/abs/1706.03762

[2] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[3] OpenAI, "Gpt-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[4] H. T. et all, "Llama 2: Open foundation and fine-tuned chat models," 2023.

[5] CommonCrawl, "Common crawl maintains a free, open repository of web crawl data that can be used by anyone. (last accessed friday, sep 15)." [Online]. Available: https://commoncrawl.org

[6] Q. Dong, L. Li, D. Dai, C. Zheng, Z. Wu, B. Chang, X. Sun, J. Xu, L. Li, and Z. Sui, "A survey on in-context learning," *arXiv preprint arXiv:2301.00234*, 2023.

[7] J. White, Q. Fu, S. Hays, M. Sandborn, C. Olea, H. Gilbert, A. Elnashar, J. Spencer-Smith, and D. C. Schmidt, "A prompt pattern catalog to enhance prompt engineering with chatgpt," 2023.

[8] Y. Wang, W. Zhong, L. Li, F. Mi, X. Zeng, W. Huang, L. Shang, X. Jiang, and Q. Liu, "Aligning large language models with human: A survey," *arXiv preprint arXiv:2307.12966*, 2023.

[9] A. Karapantelakis, P. Alizadeh, A. Alabassi, K. Dey, and A. Nikou, "Generative ai in mobile networks: a survey," *Annals Telecommunications, Link: https://link.springer.com/article/10.1007/s12243-023-00980-9*, 2023.

[10] L. Bariah, Q. Zhao, H. Zou, Y. Tian, F. Bader, and M. Debbah, "Large language models for telecom: The next big thing?" *arXiv preprint arXiv:2306.10249*, 2023.

[11] Y. Huang, H. Du, X. Zhang, D. Niyato, J. Kang, Z. Xiong, S. Wang, and T. Huang, "Large language models for networking: Applications, enabling techniques, and challenges," *arXiv preprint arXiv:2311.17474*, 2023.

[12] Y. Chen, R. Li, Z. Zhao, C. Peng, J. Wu, E. Hossain, and H. Zhang, "Netgpt: A native-ai network architecture beyond provisioning personalized generative services," *arXiv preprint arXiv:2307.06148*, 2023.

[13] Y. Huang, M. Xu, X. Zhang, D. Niyato, Z. Xiong, S. Wang, and T. Huang, "Ai-generated network design: A diffusion model-based learning approach," *IEEE Network*, pp. 1–1, 2023.

[14] C. Wang, M. Scazzariello, A. Farshin, D. Kostic, and M. Chiesa, "Making network configuration human friendly," *arXiv preprint arXiv:2309.06342*, 2023.

[15] L. Bariah, H. Zou, Q. Zhao, B. Mouhouche, F. Bader, and M. Debbah, "Understanding telecom language through large language models," *arXiv preprint arXiv:2306.07933*, 2023.

[16] S. Soman *et al.*, "Observations on llms for telecom domain: Capabilities and limitations," *arXiv preprint arXiv:2305.13102*, 2023.

[17] H. Holm, "Bidirectional encoder representations from transformers (bert) for question answering in the telecom domain.: Adapting a bert-like language model to the telecom domain using the electra pre-training approach," 2021.

[18] M. Gunnarsson, "Multi-hop neural question answering in the telecom domain," 2021.

[19] 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.

[20] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[21] T. Lin, Y. Wang, X. Liu, and X. Qiu, "A survey of transformers," *AI Open*, 2022.

[22] A. Radford, K. Narasimhan, T. Salimans, I. Sutskever *et al.*, "Improving language understanding by generative pre-training," 2018.

[23] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, I. Sutskever *et al.*, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[24] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[25] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[26] G. Penedo, Q. Malartic, D. Hesslow, R. Cojocaru, A. Cappelli, H. Alobeidli, B. Pannier, E. Almazrouei, and J. Launay, "The refinedweb dataset for falcon llm: outperforming curated corpora with web data, and web data only," *arXiv preprint arXiv:2306.01116*, 2023.

[27] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: pre-training of deep bidirectional transformers for language understanding," *CoRR*, vol. abs/1810.04805, 2018. [Online]. Available: http://arxiv.org/abs/1810.04805

[28] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized BERT pretraining approach," *CoRR*, vol. abs/1907.11692, 2019. [Online]. Available: http://arxiv.org/abs/1907.11692

[29] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "SQuAD: 100,000+ questions for machine comprehension of text," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, Texas: Association for Computational Linguistics, Nov. 2016, pp. 2383–2392. [Online]. Available: https://www.aclweb.org/anthology/D16-1264

[30] P. Rajpurkar, R. Jia, and P. Liang, "Know what you don't know: Unanswerable questions for SQuAD," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, 2018, pp. 784–789. [Online]. Available: https://www.aclweb.org/anthology/P18-2124

[31] P. S. H. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive NLP tasks," *CoRR*, vol. abs/2005.11401, 2020.

[32] G. Gerganov, "Ggml library: Large language models for everyone," *https://github.com/rustformers/llm/tree/main/crates/ggml*, 2022.

[33] "Langchain framework," accessed: 2023-09-05. [Online]. Available: https://python.langchain.com/docs/

[34] "Langchain query engine," accessed: 2023-09-05. [Online]. Available: https://python.langchain.com/docs/modules/data_connection/

[35] "Faiss: A library for efficient similarity search," (Accessed on 10/05/2023). [Online]. Available: https://tinyurl.com/2s3d8kfc

[36] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, "Bertscore: Evaluating text generation with bert," in *International Conference on Learning Representations*, 2020.

[37] "Llama.cpp, a library to perform fast inference of llms," accessed: 2023-09-28. [Online]. Available: https://github.com/ggerganov/llama.cpp