5G INSTRUCT Forge: An Advanced Data Engineering Pipeline for making LLMs learn 5G

Azzedine Idir Ait Said, Abdelkader Mekrache, *Member, IEEE,* Karim Boutiba, *Member, IEEE,* Kostas Ramantas, Adlen Ksentini, *Senior Member, IEEE,* and Moufida Rahmani.

***Abstract*—Large Language Models (LLMs) have transformed various fields with their remarkable ability to comprehend and generate human-like text. Despite these advancements, their effectiveness in specialized domains such as finance, law, medicine, and telecommunications remains limited. To adapt these models to new domains, it is essential to train them on relevant datasets. Fine-tuning is a well-known method for training LLMs on new tasks using specialized datasets. However, generating these specialized datasets presents a critical challenge, as structuring the data appropriately for effective learning is complex. To address this challenge, this paper presents 5G Instruct Forge, an advanced data engineering pipeline designed to create domain-specific datasets for 5G networking, particularly from the 3rd Generation Partnership Project (3GPP) specifications. By processing unstructured documents, i.e., 3GPP Technical Specifications (TSs), into structured formats, our pipeline enables LLMs to be fine-tuned for understanding and generating 5G-related content. As a proof of concept, we generated the OpenAirInterface (OAI) Instruct dataset using our pipeline, utilizing a subset of the 3GPP TSs used to develop OAI. Evaluation results demonstrate that training generic open-source LLMs on this dataset resulted in new 5G-aware LLMs outperforming OpenAI's GPT-4 on 5G-specific tasks.**

***Index Terms*—LLMs, fine-tuning, 5G, 3GPP, OAI.**

I. Introduction

Large Language Models (LLMs) have revolutionized numerous fields with their remarkable ability to understand and generate human-like text [1]. These models require extensive training on vast datasets, such as OpenWebText [2], Common Crawl [3], and Dolma [4], to capture and learn the nuances of human language from diverse linguistic contexts. By leveraging such comprehensive data, LLMs can mimic human language patterns and understand intricacies across different domains and dialects [5], making them indispensable tools for natural language understanding, generation, and a variety of downstream tasks. These tasks include sentiment analysis, machine translation, and summarization [1], which require not only a deep understanding of language but also the ability to infer and reason from context. In addition to commercial models like OpenAI's GPT series, including GPT-3 and GPT-4, there are notable open-source LLMs available, such as Meta's Llama series [4], Google's T5 series [6], and Gemma series [7]. These models provide the research community and industry with powerful tools to build upon and integrate into a wide array of applications, thus democratizing access to cutting-edge Natural Language Processing (NLP) technologies [1].

As the demand for 5G network services continues to grow in today's digital world, networks must be updated to meet their evolving requirements. eXtended Reality (XR) and Virtual Reality (VR) applications, for example, require critical Quality of Service (QoS) in terms of latency and throughput [8]. To address these needs, 5G networks need upgrades with new functionalities. Multiple standardization bodies are actively working to establish guidelines for implementing 5G. The 3rd Generation Partnership Project (3GPP) is making significant efforts to propose Technical Specifications (TSs) for developers and technical personnel to aid in the development of 5G network procedures [9]. However, these TSs are often complex and voluminous, making it challenging for developers and readers to find the necessary information. In this context, chatbot assistants become essential in answering questions about the TSs' content. With the rapid advancement in Generative Artificial Intelligence (GenAI), LLMs offer a promising solution. These models can learn from vast datasets and generate human-like text [1], assisting users in navigating and understanding the extensive 3GPP TSs documents related to 5G networks.

As 5G-aware LLMs offer a promising approach in enhancing the efficiency of information provision and development of 5G networks (and other domains given their standards), creating them is a complex task [10]. Training existing pre-trained LLMs in the new 5G domain is not straightforward, as this training necessitates a high-quality dataset specifically tailored for the given domain [11]. This involves collecting data from 3GPP TSs, cleaning, and generating a structured dataset that is training-optimized (or training-ready), i.e., ready for the LLM to train on. Among these steps, the cleaning process is particularly complex due to challenges in handling figures and paragraphs referencing other paragraphs within the TSs (and from other TS documents). Additionally, data generation presents complexities related to post-processing the cleaned TSs as LLMs require a specific data structure to respond to users' 5G-related queries effectively. Therefore, developing an effective 5G-specific LLM requires a pipeline to

A. Ait Said is with the National Higher School of Computer Science, Algeria, and EURECOM, France (e-mail: ja_aitsaid@esi.dz).

A. Mekrache is with EURECOM, France (e-mail: abdelkader.mekrache@eurecom.fr).

K. Boutiba is with EURECOM, France (e-mail: Karim.boutiba@eurecom.fr).

K. Ramantas is with IQUADRAT, Spain (e-mail: kramantas@iquadrat.com).

A. Ksentini is with EURECOM, France (e-mail: adlen.ksentini@eurecom.fr).

M. Rahmani is with the National Higher School of Computer Science, Algeria (e-mail: m_rahmani@esi.dz).

The Gitlab repository is available at: https://gitlab.eurecom.fr/netsoft/5g-instruct-forge

A. Ait Said and A. Mekrache contributed equally to this paper.

generate these specialized datasets, ensuring that future LLMs can bridge this critical gap in 5G technology understanding and application [12].

To this end, this paper addresses the aforementioned challenges by proposing a data pipeline designed for dataset generation specifically to train (i.e., fine-tune) LLMs for understanding 5G technologies. The pipeline can collect and clean a set of 3GPP TSs and perform structured data generation using existing state-of-the-art LLMs. The resultant data is subsequently used to train open-source LLMs to create new 5G-aware LLMs capable of understanding and generating 5G-related text. The major contributions of the paper are as follows:

- We detail a robust framework for gathering, processing, and cleansing 3GPP TSs. This framework not only simplifies the transformation of highly technical, unstructured, and comprehensive documents into a clean format but also ensures the retention of critical information, making it conducive for LLM training.
- Within the aforementioned framework, we leverage state-of-the-art LLMs to generate prompt/completion pairs. This approach relies on the advancements of powerful LLMs to create 5G-related datasets for training other LLMs, with the goal of developing 5G-aware LLMs.
- As a result of the pipeline, we present an open-source dataset, namely the OAI Instruct dataset[1]. This dataset is uniquely tailored to enhance the comprehension and generation capabilities of LLMs concerning a subset of 22 3GPP TSs.
- To evaluate the pipeline's effectiveness, we perform a specific type of fine-tuning called freeze-tuning to adjust the parameters of open-source LLMs based on our previously crafted LLM, aiming to create 5G-aware LLMs and demonstrate the utility of our dataset.
- We thoroughly evaluate GPT-4[2], open-source LLMs, and fine-tuned (5G-aware) LLMs using the evaluation segment of our newly created dataset. This assessment critically analyzes the enhanced performance of these models on specialized tasks, highlighting the effectiveness of our dataset for both training and evaluation methodologies.

The remaining sections of this paper are structured as follows: Section II describes related works and background. In section III, we illustrate the system design. In Section IV, we present results proving our work's pertinence. Finally, section V concludes the paper.

II. Related Works and Background

In this section, we present related works and background on ways to create LLMs, including training methodologies and use cases, emphasizing the need for specialized datasets. We then review existing literature on LLMs dataset generation.

A. LLMs creation

Creating LLMs involves intricate processes of pre-training and fine-tuning, essential for developing models with specialized capabilities [1]. Pre-training typically includes training on massive text corpora using the Transformer architecture to develop generalized language representations, resulting in foundational LLMs such as Llama3 70B [13], Qwen1.5 110B [14], and Mixtral0.1 8x22B [15]. These foundational models provide a solid base with a broad understanding of language across diverse contexts, making them versatile yet general in their capabilities. Fine-tuning then adapts these pre-trained foundational models to specific tasks or domains by leveraging domain-specific datasets, which are crucial for providing the necessary context and terminology unique to particular fields. This stage results in more specialized LLMs, such as those used for 5G applications, tailored to deliver precise responses in specific sectors. Further emphasizing the importance of domain-specific adaptation, LLMs like BioBERT [16] and FinBERT [17] have garnered significant attention for their ability to leverage specialized knowledge to improve performance in specific fields. For instance, BioBERT, tailored for biomedical text mining, demonstrates substantial performance gains achieved by training on biomedical literature. Similarly, FinBERT, fine-tuned on financial data, significantly improves the handling of financial texts. The use of such models showcases the efficacy of fine-tuning LLMs on domain-specific datasets to enhance their understanding and generation of specialized content, thereby highlighting the crucial role of specialized datasets in enhancing the applicability and performance of LLMs in specialized contexts [1].

Advanced techniques are continuously developed to enhance the outputs of LLMs. Among these, Instruct-type fine-tuning represents a significant evolution in the field of AI. Unlike traditional methods, which adjust models using specific task-related data, Instruct-type datasets such as Alpaca [18], UltraChat[19], and OpenOrca[20] are designed to teach models to follow user instructions in a more generalized manner. These datasets contain varied prompts that guide models in understanding and executing user commands effectively, thus improving their interactive capabilities [1]. This approach not only injects a general ability for instruction following into LLMs but also enhances the user experience by enabling more natural and intuitive interactions. Prompting strategies, such as Retrieval-Augmented Generation (RAG), play a crucial role in enhancing the capabilities of LLMs. RAG is a technique in prompting that involves providing specific inputs to models to generate desired outputs. It dynamically retrieves information from a knowledge base during the generation process, combining the retrieval capabilities of a dense vector search with the generative power of an LLM. This approach, highlighted by platforms like Langchain[3], enables the model to access extensive information beyond its initial training data, improving output accuracy [1]. However, the effectiveness of RAG depends on a large context window and the quality of the knowledge base, which can influence the accuracy of the outputs. These sophisticated techniques collectively enhance the utility of LLMs in various applications, from chatbots to virtual assistants, showcasing their flexible adaptability, which is not seen in more narrowly focused models [1].

[1]https://huggingface.co/datasets/Netsoft/oai-instruct

[2]gpt-4-0125-preview

[3]https://www.langchain.com

B. Datasets generation

The creation and utilization of instruct datasets are pivotal for developing effective domain-specific LLMs, which are essential for achieving high performance in specialized contexts. Instruct datasets, by providing the necessary contextual examples, enable LLMs to comprehend and generate domain-specific content accurately. For example, models like T5, which operates within a unified text-to-text framework, highlight the success of using instruct datasets to attain state-of-the-art performance across various NLP tasks [21]. Moreover, GPT-3's fine-tuning process with instruct datasets has proven its enhanced capabilities in handling tasks that demand deep domain knowledge [22]. This trend underscores the critical role of well-constructed instruct datasets that not only embed detailed domain-specific knowledge but also support the linguistic capabilities required for broad NLP applications. Nowadays, open source LLM developers consistently fine-tune their foundational models on instruct datasets like the open source LLMs Meta-Llama-3-8B[4] and Meta-Llama-3-8B-Instruct[5], which diverge across the LLaMA family's various versions and parameter sizes, reflecting a standard practice in the field.

Research on data generation pipelines provides valuable methodologies for creating high-quality datasets. For instance, researchers in [23] explore techniques to enhance dataset diversity and quality, such as Translation Data Augmentation (TDA). The latter augments training data by altering existing sentences in a parallel corpus to diversify training examples while preserving semantic equivalence. It involves selecting targeted rare words for substitution to provide new contexts for these words, using language models to suggest plausible substitutions, and ensuring that the substitutions and their translations maintain grammatical and semantic coherence. Similarly, the authors of [24] illustrate methodologies for constructing datasets that improve model performance, such as synthetic data generation and multi-task learning. Recently, with the rapid explosion in generative AI, LLMs are being widely used as a key component in dataset generation pipelines [25]. For example, tools like DataDreamer [26] simplify the creation of synthetic datasets and facilitate reproducible LLM workflows, addressing challenges such as model brittleness and reproducibility. Additionally, the open source model Bonito, introduced in [27], complements these capabilities by transforming regular text into specialized training exercises for language models. Bonito utilizes meta-templates from datasets like P3[6] to craft synthetic tasks for various domains, enhancing the diversity and applicability of training materials. However, both tools exhibit limitations in handling data from `.docx` files and struggle with the robotic text format typical of 3GPP specifications and technical reports. Furthermore, they do not effectively manage non-textual data such as figures and tables, which are crucial in these documents, comprising about 66% of the embedded knowledge.

[4]https://huggingface.co/meta-llama/Meta-Llama-3-8B
[5]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[6]https://huggingface.co/datasets/bigscience/P3

C. Research gap

While substantial progress has been made in developing domain-specific LLMs and creating instruct datasets, significant gaps remain in the availability of specialized datasets for the 5G domain. Although [28] and [29] provide methods to generate datasets for Telecom-specific LLMs and for benchmarking LLMs in telecom, respectively, these approaches are broad and cover the entire Telecom domain. Furthermore, despite figures and tables within TSs containing important information that does not exist in the text, these approaches do not take them into account in 3GPP TSs. Thus, an automated pipeline to convert 5G 3GPP TSs (including figures and tables) into a 5G-specialized LLM is absent from current research. Our contribution addresses these gaps by developing an innovative, LLM-centric data generation pipeline specifically for 5G. This pipeline generates instruct datasets from 3GPP TSs, enabling the creation of 5G-aware LLMs.

III. System Design

Our proposed solution, 5G Instruct Forge, presents a comprehensive pipeline designed to transform the 3GPP TSs into a well-structured dataset optimized for fine-tuning, as depicted in Fig. 1. This transformation involves several stages: *(i) Specification gathering,* which can be done either through our automated scripts or manually. The specifications, generally in `.docx/.doc` formats, are then meticulously processed; *(ii) Cleaning and processing,* involves extracting essential elements from each document, such as the table of contents, tables, figures, abbreviations, and definitions. These elements are managed utilizing LLM data generation techniques. Following this, the documents are converted into plain text and concatenated to create an embedding database. This database plays a crucial role in the next stage; *(iii) Data generation using LLMs*, supports the creation of eight distinct task type entries using powerful state-of-the-art LLMs; *(iv) Post-processing,* verifies the correctness and integrity of the generated data.

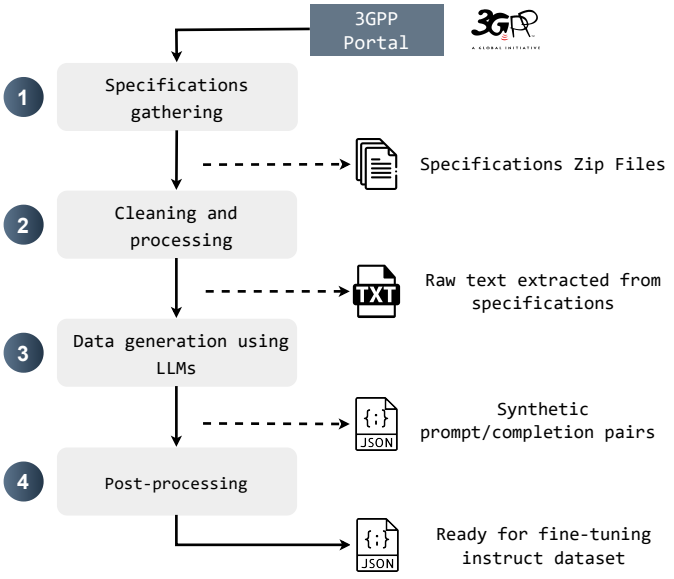


Fig. 1: 5G Instruct Forge pipeline stages.

A. Specifications gathering

The first part of the project involves gathering 3GPP TSs from the 3GPP portal[7]. The latter is a central repository for all 3GPP TSs and reports, which are essential for developing and maintaining global telecommunications standards, particularly for mobile networks. The portal allows users to download all specifications from a specific release or a subset of specifications relevant to their needs. Users can also specify a list of 3GPP specifications by providing the release number and the specific ID of the documents, such as "23.209", where "23" represents the release number and "209" is the specification ID. The downloaded files are provided in `.zip` archives, which include the specifications in `.docx` or `.doc` formats, along with other files in `.yaml` or `.taml` formats. We are only interested in the `.docx` files and the `.doc` files are converted to `.docx` to standardize the process. The 3GPP TSs follow a consistent format and adhere to specific styling policies, including the use of bold text, various heading levels (T1), and structured paragraphs, as shown in Fig. 2. This consistency in formatting makes it possible to systematically extract information using the `python-docx`[8] library.

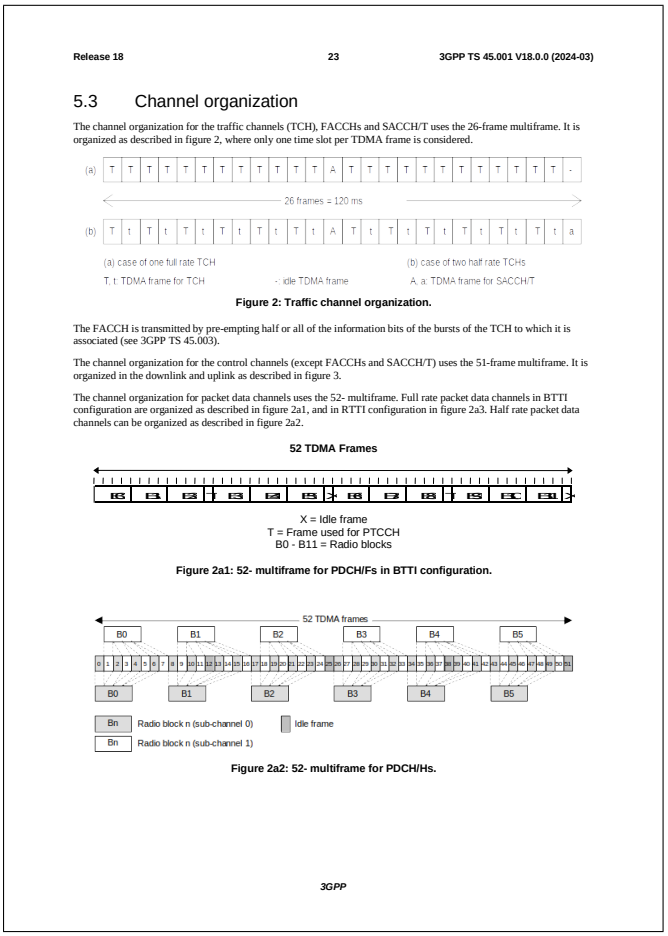


Fig. 2: Page 23 of the 3GPP TS 45.001 V18.0.0 (2024-03)

B. Cleaning and processing

In the cleaning and processing phase, the primary objective is to refine the extracted 3GPP TSs to ensure the dataset is clean, relevant, and structured for fine-tuning purposes. This involves several key steps. First, we remove the initial and final sections of the specifications, as these often contain ancillary information such as titles, 3GPP contact details, addresses, phone numbers, and annexes primarily written for API purposes. These sections are not pertinent to the core 5G knowledge we aim to capture, so we choose to exclude them aiming at streamlining the documents to focus on the valuable content. Next, we extract the table of contents from the main text files and store them separately. They provide a navigational map of the document structure and are useful for referencing and indexing. Additionally, we isolate non-textual data, such as figures and tables, which often contain the bulk of the information in the specifications. Neglecting this data would result in a significant loss of content. Therefore, we treat this non-textual data separately, aiming to convert it into textual formats that an LLM can process and understand. This approach ensures that all critical information is effectively captured and utilized.

Figures and their annotations are processed using a highly accurate Vision-Text model, namely MiniCPM[9], which provides detailed textual descriptions of the figures, ensuring that all visual information is translated into a textual format suitable for LLM training. Fig.3 illustrates the text generation from 3GPP TS figures using the Vision-Text model. The process involves first loading the vision model, then converting the image to PNG format, and finally processing the image using the model to obtain the textual description. Similarly, tables are handled by feeding them into an advanced LLM, which generates detailed descriptions of the tabular data. This approach ensures that complex information contained in tables is accurately represented in a textual form, making it more accessible for the language model.

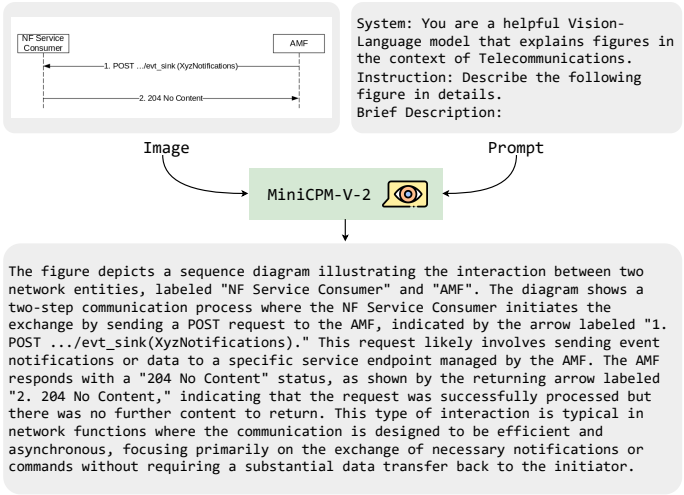


Fig. 3: Figure textual description generation using MiniCPM.

[7]https://portal.3gpp.org/
[8]https://python-docx.readthedocs.io/en/latest/index.html
[9]https://huggingface.co/openbmb/MiniCPM-V-2

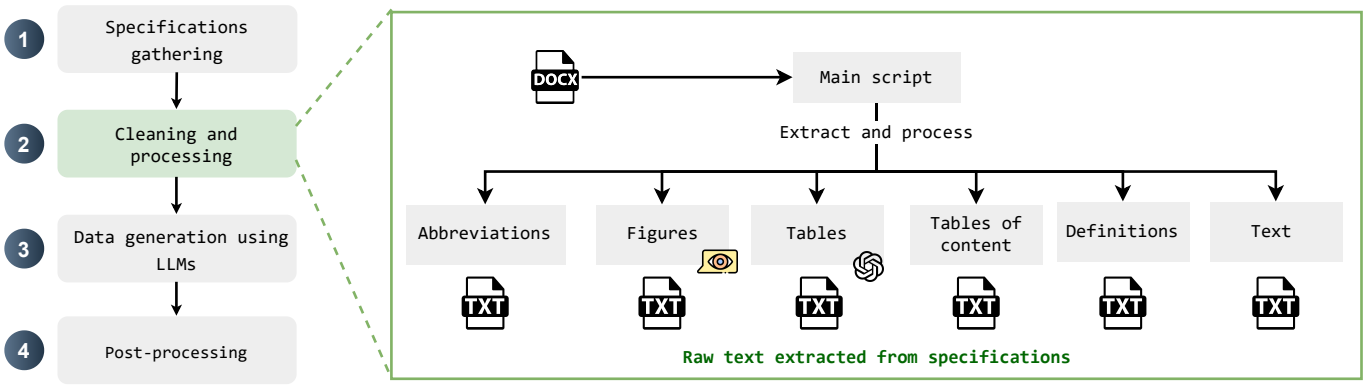


Fig. 4: Illustration of the data cleaning and processing stage.

In addition to processing structural elements of the specifications, we meticulously extract abbreviations and definitions from each document. This step is crucial as these elements encapsulate essential information that is imperative for both human understanding and LLMs to interpret the technical content accurately. Following the individual treatment of figures, tables, tables of contents, abbreviations, and definitions, we then extract the raw text from all `.docx` files of the specifications. This raw text is concatenated into a single, large `.txt` file, which forms the basis for the subsequent stages of our data generation pipeline. This consolidated text file facilitates a streamlined integration and manipulation of the data, ensuring a comprehensive dataset is available for further analysis and machine learning applications. All these steps are summarized in Fig. 4.

C. Data generation using LLMs

Fig.5 illustrates the data generation stage using LLMs. This phase is a critical component of our pipeline, which focuses on transforming clean, raw textual data about 5G into a domain-specific dataset suitable for training LLMs to become 5G while maintaining their general language capabilities or evaluation. Our approach leverages both OpenAI's GPT-3.5 Instruct model[10] and the open-source Llama3 70B model[11] to create this dataset. The GPT-3.5 Instruct model is utilized for its superior accuracy and more controlled output, which reduces the number of inadequate entries during post-processing. Its advanced capabilities minimize the risk of generating inappropriate content. However, the use of GPT-3.5 comes with challenges, such as the need for data transmission to third parties, internet connectivity requirements, and potential latency issues. To address these challenges, we incorporate the Llama3 70B model, which, despite being less accurate due to its smaller size, offers advantages such as faster generation times and local operation with minimal latency. While this may result in a higher rate of inadequate entries that require additional filtering, it provides a valuable balance between control, speed, and data privacy.

The pipeline begins by creating an embedding database from concatenated text files generated in the previous step. This database is crucial as it facilitates the LLMs' contextual understanding during the generation of prompt/completion pairs. The concatenated text files are transformed into embeddings using an OpenAI model called text-embedding-3-large[12], which captures their semantic meanings. The resulting embeddings are essentially vectors of numerical values representing each word in the concatenated files and are stored in a vector database. We used Chroma DB[13] because it efficiently organizes and manages the embeddings, allowing the LLMs to quickly retrieve relevant information when generating responses. By adopting a RAG approach, the pipeline ensures that the LLMs access relevant context, thereby improving the quality and relevance of the generated data. Following the creation of the embedding database, the pipeline proceeds to generate either training data or evaluation data. The dataset for training will be an Instruct dataset used exclusively for training LLMs to create 5G-specialized models. In contrast, the evaluation data will be used to assess the created LLMs' abilities in 5G knowledge. Next, we delve into the details of each part.

1) Training data generation: Our pipeline is designed to produce a variety of training tasks, including question-answering pairs, fill-in-the-gap pairs, text reformulation, title and summary generation, and other specialized tasks. These tasks are important because they enable the LLMs to: (i) Learn raw knowledge through question-answering; (ii) Identify key entities and concepts with fill-in-the-gap tasks; (iii) Explore alternative expressions via text reformulation; and (iv) Correct misconceptions through tasks addressing false claims. Each type of task contributes significantly to the development of the LLM's domain expertise and language understanding. For question-answering tasks, the process involves two stages: generating questions and then generating answers. This two-pass approach ensures that the questions are well-formed and relevant, while the answers are accurate and informative. For simpler tasks, such as filling in the gaps, a single-pass approach is used where the LLM completes missing parts of the text.

2) Evaluation data generation: The evaluation data generation step is crucial for assessing the knowledge and capabilities

[10]https://platform.openai.com/docs/models/gpt-3-5-turbo
[11]https://huggingface.co/meta-llama/Meta-Llama-3-70B-Instruct
[12]https://openai.com/index/new-embedding-models-and-api-updates/
[13]https://www.trychroma.com

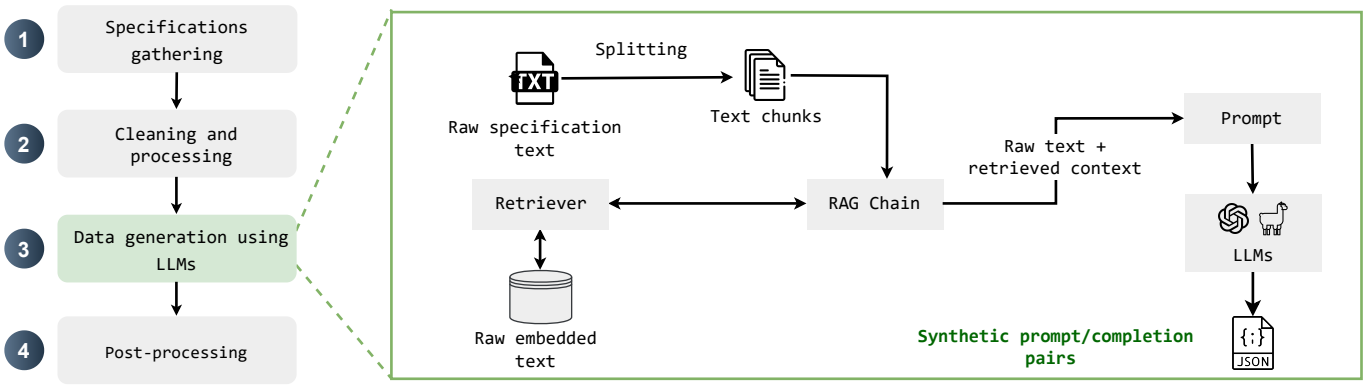


Fig. 5: Illustration of the data generation using LLMs stage.

of the trained LLMs. This step also relies on the concatenated text file containing the 5G specifications, but the structure of the evaluation dataset differs from that of the training data. We propose two types of evaluation datasets to test the LLM's understanding and generation abilities comprehensively: (i) The first type of evaluation dataset resembles an exam designed to assess both the knowledge and understanding of the LLM. This dataset includes columns for "question" and options labeled from 1 to 4, with the correct answer provided separately. To further enhance this dataset and introduce more diversity and challenge for the LLM, we incorporate yes/no questions where the only possible answers are "yes" or "no". This mixed format not only evaluates the model's ability to recall information but also tests its decision-making skills; (ii) The second type of evaluation dataset focuses on assessing the LLM's knowledge and generative capabilities through direct question answering. This dataset is similar in structure to the training data's question-answering task but without specifying an instruction. It includes questions derived from the 5G knowledge base, and the LLM is expected to generate accurate and contextually appropriate answers. This format allows for a straightforward evaluation of the model's ability to generate coherent and relevant responses based on its learned knowledge. By using these varied evaluation datasets, we can comprehensively assess the LLM's proficiency in 5G technology, ensuring that it not only understands the material but can also generate high-quality responses.

D. Post-processing

In the post-processing phase, we focus primarily on verifying that the training and evaluation datasets conform to the predefined format. We systematically check for any entries that do not meet this format, identifying and removing those that are malformed or incomplete. By ensuring that all dataset entries adhere to this structure, we maintain the integrity and consistency necessary for effective training of domain-specific LLMs and their evaluation. This verification step is critical to eliminate noise and ensure that the final dataset is clean and reliable for use in developing 5G expert language models.

IV. Performance Evaluation

The section is structured into three subsections: *(i) Evaluation setup,* which details the experimental setup; *(ii) Evaluation results,* which presents the experiments results; and *(iii) Evaluation conclusion,* which offers additional insights related to the experiments.

A. Evaluation setup

The evaluation preparation includes four steps: generating a 5G-related dataset for LLM training using the pipeline, creating 5G-aware LLMs, deploying 5G-aware LLMs, and preparing the evaluation datasets.

- *(i) Generating a 5G-related dataset:* We selected 22 3GPP TSs that were used to develop OpenAirInterface (OAI)[14], to generate the dataset using the proposed pipeline, named the OAI instruct dataset.
- *(ii) Creating 5G-aware LLMs:* Three open-source state-of-the-art LLMs were fine-tuned using the OAI instruct dataset with the freeze-tuning method within the LLaMA Factory framework [30], ensuring that a percentage of the core model's parameters were preserved while adapting it to the specific task. These LLMs are referenced in Table. I, and the most important hyperparameters for LLM freeze-tuning are referenced in Table. II;
- *(iii) Deploying 5G-aware LLMs:* The resulting 5G-aware LLMs were deployed on a single machine equipped with an Nvidia A100 GPU with 80GB of vRAM, utilizing the Llama Factory project for deployment. In this setup, we set the LLMs' temperature to 0.7 to balance creativity and coherence in the responses;
- *(iv) Preparing the evaluation dataset:* To validate the effectiveness of our methodology and the integration of knowledge into the resulting 5G-aware LLMs, we employed the evaluation benchmark from the OAI instruct dataset. This benchmark, comprising over 9,000 question-and-answer pairs, is specifically designed to measure the model's capacity to respond accurately to novel information about 5G technologies. Additionally, we incorporated a dataset featuring approximately 200 questions, each with four answer options (called 5G

[14]https://openairinterface.org

exam). This additional dataset aims to evaluate not only the LLM's expertise in 5G but also its proficiency in language comprehension. This dual assessment approach addresses concerns regarding potential compromises in the LLM's language capabilities due to the fine-tuning process.

TABLE I: Open-source LLMs used for training.

LLM	Number of parameters	Size	Reference
Llama3	8B	16 Gb	[15]
Solar	10.7B	21 Gb	[16]
Mistral	7B	14 Gb	[17]

TABLE II: Fine-tuning hyperparameters for each open-source LLM.

LLM	Trainable parameters	Number of epochs	Learning rate ($\times 10^{-5}$)	Per-device training batch
Llama3	9%	4	4	4
Solar	12%	4	4	4
Mistral	11%	4	5	4

B. Evaluation results

The evaluation results are structured into four stages: *(i) OAI Instruct dataset,* which presents a dataset generated by the proposed pipeline to create 5G-aware LLMs; *(ii) 5G-aware LLMs training,* which outlines the chosen fine-tuning methodology and includes an ablation study on the hyperparameters of the corresponding fine-tuning approach; *(iii) 5G-aware LLMs quality,* which presents and analyzes the performance of the newly developed 5G-aware LLMs. At this stage, we conduct several evaluations, beginning with the training of the LLMs. We then apply various metrics (e.g., BERTScore, SemScore) to assess the quality of the generated texts. Additionally, we monitor the generation time of these LLMs and perform an exam test to evaluate their understanding of 5G; and *(iv) Expert satisfaction,* which assesses user satisfaction with the newly created 5G-aware LLMs.

1) ***OAI Instruct dataset****:* As a proof of concept for our pipeline, we generated a dataset named OAI Instruct. This dataset was created by applying our data generation pipeline to a set of 22 3GPP TSs, upon which the renowned open-source implementation of 5G networks, OAI, is built. The resulting dataset, which is available at [18], will be used to create a specialized LLM capable of interacting with and understanding aspects of OAI (i.e., 5G). The OAI Instruct dataset includes 87,719 entries in the training set and 9,557 entries in the test set with a total size of ≈100Mb (≈80Mb for training & ≈6Mb for evaluation). The training dataset comprises columns for 'instruction', 'task_type', 'input', 'completion', and 'prompt', with a diverse range of instructions and task types. The test dataset is structured differently, featuring 'completion', 'prompt', and 'completion2' columns to evaluate the model's performance using known metrics, such as BERTScore[31] and SemScore [32], which require multiple references, hence the use of two completions instead of just one. Examples of entries in the instruct-type dataset are shown in Fig.6.

In comparison to other domain-specific instruct datasets found in the literature, the finance-alpaca dataset[19] is an illustrative example, with a size of 42.9 MB. It has successfully facilitated the fine-tuning of models such as distilgpt2-finance[20] and Llama-personal-finance[21], featuring a structure that includes the key fields of 'instruction', 'input', and 'output', similar to our dataset. Another noteworthy dataset is lawinstruct[22], which, due to its larger size and broader source inclusion, supports a more extensive range of fine-tuned LLMs. Our goal underscores the effectiveness of modest-sized datasets, suggesting that a well-structured dataset of ≈80Mb, coupled with the right fine-tuning approach and hyperparameters, can proficiently develop an LLM expert capable of learning, memorizing, and reasoning within a targeted subfield. This approach aligns with our strategy of creating highly specialized, efficient LLMs without the need for excessively large or comprehensive datasets.

2) ***5G-aware LLMs training****:* To demonstrate the effectiveness of our dataset, we initially employ it to fine-tune LLMs. Fine-tuning involves adapting pre-trained models to specific tasks or domains using specialized datasets, which mathematically entails adjusting the model's parameters θ to minimize a loss function $L(\theta)$ across the dataset. The objective of fine-tuning can be formalized as:

$$\theta^* = \arg\min_{\theta} \sum_{i=1}^{N} L(y_i, f(x_i; \theta)) \tag{1}$$

where x_i represents the input data, y_i the corresponding labels, $f(x_i;\theta)$ the model's predictions, and N the number of training examples. The loss function L typically quantifies the discrepancy between the model's predictions and the actual labels, such as the cross-entropy loss in classification tasks. In practical terms, fine-tuning utilizes gradient descent-based optimization algorithms to update the parameters. The gradient $\nabla_\theta L$ of the loss function concerning the parameters is computed, leading to the parameter updates:

$$\theta \leftarrow \theta - \eta \nabla_\theta L \tag{2}$$

where η is the learning rate. An efficient approach known as Parameter-Efficient Fine-Tuning (PEFT) [33] involves updating only a subset of the model's parameters, thus reducing computational demands. Commonly, this entails freezing most of the model's layers and fine-tuning only the upper layers, expressed as:

$$\theta_f = \theta_f - \eta \nabla_{\theta_f} L, \quad \theta_u = \text{constant} \tag{3}$$

Here, θ_f represents the fine-tuned parameters, while θ_u denotes the unfrozen parameters. Techniques like LoRA [34]

[15]https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct
[16]https://huggingface.co/upstage/SOLAR-10.7B-Instruct-v1.0
[17]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.1
[18]https://huggingface.co/datasets/Netsoft/oai-instruct
[19]https://huggingface.co/datasets/gbharti/finance-alpaca
[20]https://huggingface.co/lxyuan/distilgpt2-finetuned-finance
[21]https://huggingface.co/dmedhi/llama-3-personal-finance-8b-bnb-4bit
[22]https://huggingface.co/datasets/lawinstruct/lawinstruct

```
{
      "task_type": "question answering",
      "input": "What information does the AGCF send to the V-PSAP in the Npcf_UEPolicyControl Create Request message?",
      "completion": "The AGCF sends various pieces of information to the V-PSAP including the subscriber's SUPI ...",
      "instruction": "Answer the following question concisely."
},
```

Learning raw knowledge

```
{
      "task_type": "filling the gaps",
      "input": "When performing redundant transmissions for one or more QoS flows within a __ session, the __ provides two DL CN
      Tunnel Info (for __) to the UPF (PSA), indicating that one serves as the __ tunnel. ",
      "completion": "The gaps in the previous text can be completed by the following words: PDU,SMF,N9 and redundancy",
      "instruction": "Complete the text by adding the missing words."
},
```

Identifying Key Entities

```
{
      "task_type": "reformulation",
      "input": "The priority can be given to selecting PGW-C plus SMF for UEs supporting both EPC and 5GC NAS.",
      "completion": "The selection of PGW-C combined with SMF should be prioritized for UEs that are compatible with both EPC and
      5GC NAS.",
      instruction": "Rewrite the provided text differently."
},
```

Exploring Alternate Expressions

```
{
      "task_type": "false_claim_correction",
      "input": "The DL Buffering Suggested Packet Count IE indicates the minimum number of downlink data packets suggested to be
      buffered in the UP function for this PFCP session.",
      "completion": "The DL Buffering Suggested Packet Count IE does not indicate the minimum number of downlink data packets
      suggested to be buffered in the UP function for a given PFCP session. Instead, it indicates the maximum number of downlink
      data packets that can be suggested for buffering based on the UP function's capabilities, as per the UDBC feature support.",
      "instruction": "Locate any incorrect claims in the text and make corrections."
}
```

Correcting Misconceptions

Fig. 6: Example entries of OAI Instruct dataset, generated using 5G Instruct Forge.

adjust a minimal percentage of the total parameters through small, trainable modules. However, we opted not to use LoRA due to the significant domain shift involved in our project. LoRA is better suited for minor modifications, such as subtle linguistic shifts or learning from minimal data. Instead, we employed the freezing method, which enables the integration of substantial new knowledge without compromising the original LLM's language capabilities. This approach effectively manages significant domain transitions while preserving the model's linguistic integrity [33].

Indeed, freeze-tuning requires several parameters to be set, which are specified in Table II. We conducted an ablation study to determine the optimal hyperparameter configuration, which includes:

- *Percentage of trainable parameters:* This parameter underwent an ablation study to determine its optimal value, significantly affecting model performance for each open-source LLM. For example, Fig. 7 illustrates the ablation study of the Mistral LLM. This figure shows the performance of the 5G-aware Mistral LLM using freeze tuning on MMLU benchmark [35] and the 5G exam from our resulting OAI Instruct dataset. This latter involved an examination format with 200 questions, each presenting four potential answers but only one correct response. The objective of the fine-tuning is to inject knowledge into an LLM without compromising its existing knowledge. Therefore, the MMLU score measures whether the LLM has forgotten its default knowledge, while the 5G exam score assesses the knowledge acquired. From the figure, we observe that as the percentage of trainable parameters increases, the LLM learns the new knowledge better (yellow bars) without forgetting its old knowledge (blue bars). However, when training more than 11% in the case of Mistral, the LLM tends to forget much of its default knowledge and struggles to respond accurately to 5G questions because it loses the ability to answer questions based on its prior knowledge. Therefore, we choose 11% as the percentage of trainable parameters for the Mistral LLM.
- *Number of Epochs:* We chose 4 epochs based on preliminary experiments that showed diminishing returns on performance beyond this point. This allows the model to learn adequately without overfitting.
- *Learning Rate:* A learning rate of 4×10^{-5} was selected to ensure effective convergence. This range has been shown to provide a good balance between rapid convergence and stability during training, as demonstrated in the state of the art.
- *Per-Device Training Batch Size:* We set the batch size to 4, which allows for efficient use of computational resources while maintaining stability in gradient updates. This size was chosen to prevent memory overflow while ensuring that the model receives enough data for updates.

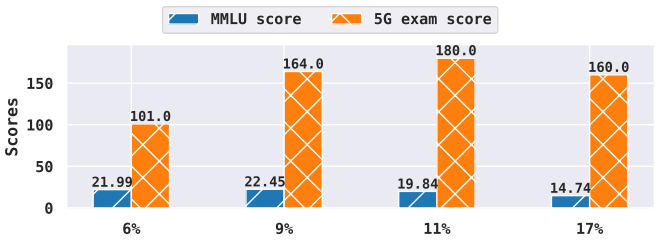
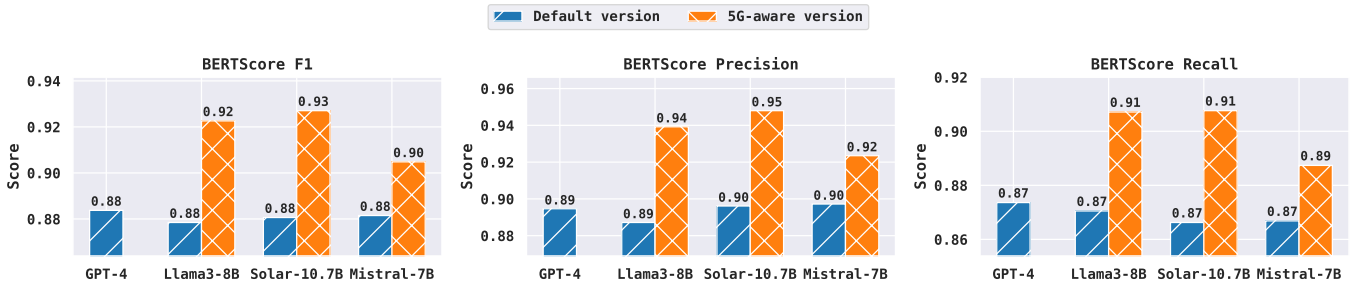
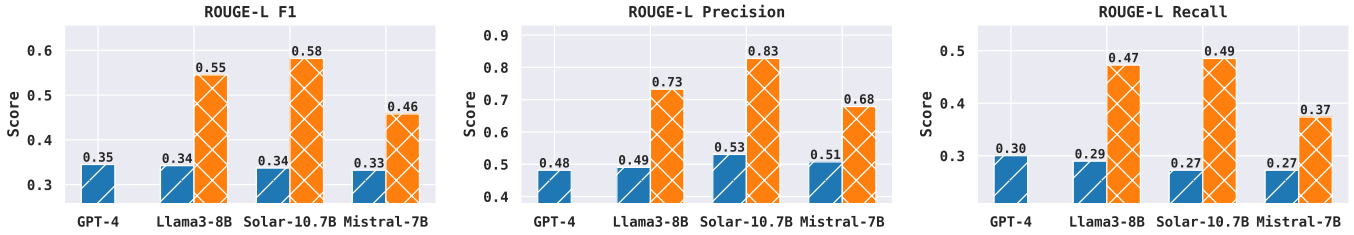


Fig. 7: Impact of the percentage of trained parameters on MMLU [35] and 5G exam scores.

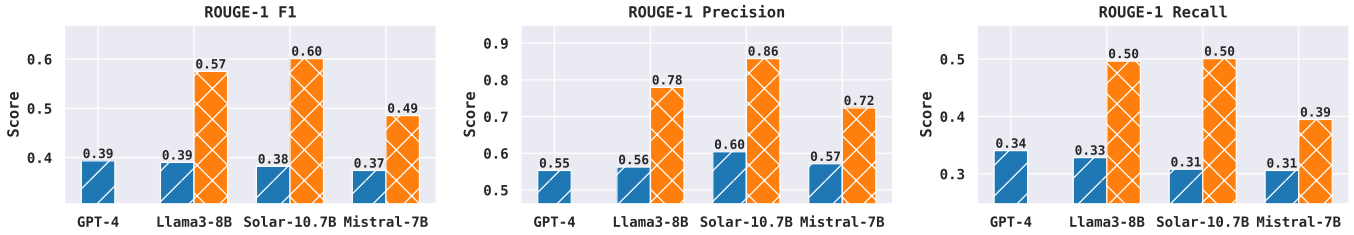
*3) **5G-aware LLMs quality**:* Following the fine-tuning process, we conducted a rigorous evaluation that consisted of comparing the LLM's answers to the benchmark questions



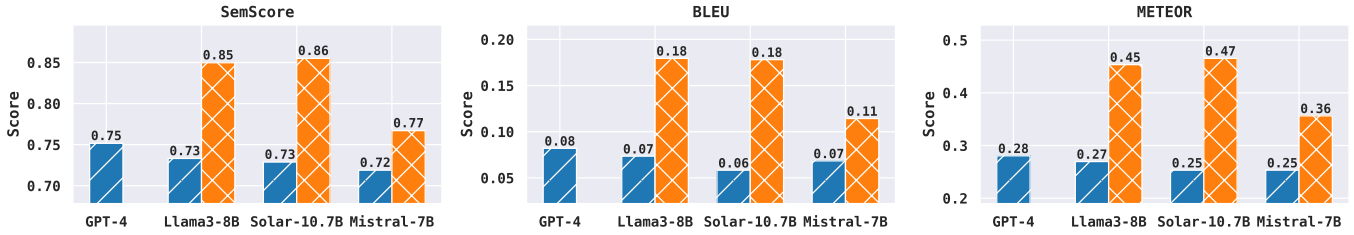
(a) BERTScore metrics.



(b) ROUGE-L metrics.



(c) ROUGE-1 metrics.



(d) SemScore, BLEU and METEOR metrics.

Fig. 8: Comparison of LLMs metrics. Bars colored in [HTML]1f77b4blue represent the Default versions of the models, while bars colored in [HTML]ff7f0eorange represent the 5G-aware versions.

against the reference answers from the dataset. To quantify the similarity between the generated and reference answers, we employed:

- *BERTScore [31]:* This metric uses cosine similarities between token embeddings from models like BERT to evaluate textual similarity and model performance. BERTScore includes three key metrics: precision, recall, and F1 score. Precision calculates the relevance of the candidate text by measuring the cosine similarity of each token to its closest counterpart in the reference text. Recall evaluates how comprehensively the candidate text covers the reference text. The F1 score combines these metrics, providing a balanced measure of textual accuracy and relevance, making BERTScore particularly effective at capturing paraphrases and maintaining semantic accuracy across different sentence structures.
- *ROUGE [36]:* This metric assesses the quality of summaries by computing overlap statistics between the generated text and a set of reference texts. ROUGE includes several key measures, such as ROUGE-N and ROUGE-L. ROUGE-N (e.g. ROUGE-1) measures the overlap of n-grams between the generated text and the references, serving as a proxy for precision and recall at the n-gram level. ROUGE-L focuses on the longest common subsequence, evaluating the fluency and order of the generated text relative to the references. Together, these

metrics provide a comprehensive evaluation of textual coherence, consistency, and relevance, making ROUGE particularly effective for assessing the quality of text summarization tasks.

- *SemScore [32]:* This metric evaluates the semantic textual similarity of text generated by models. It consists of comparing the models' output directly with target responses. SemScore computes the cosine similarity between the embeddings of the model response and the target, using high-quality transformer models such as MPNet-Base [37] to generate these embeddings. This approach allows SemScore to effectively assess whether the generated text is contextually appropriate and semantically aligned with the target responses.
- *BLEU [38]:* This metric evaluates the quality of machine-generated text by measuring how well the output aligns with a set of reference texts. BLEU includes several key components, including precision scores that assess the overlap between machine-generated text and reference texts combined using a geometric mean. It also incorporates a brevity penalty to discourage overly short responses. This penalty ensures that the candidate texts not only align well with the reference texts but also cover an adequate length, providing a balanced measure of linguistic accuracy and completeness.
- *METEOR [39]:* This metric evaluates the quality of machine-generated text by assessing both exact word matches and semantic similarity between the candidate text and reference texts. METEOR considers synonyms and stemming, providing a more nuanced evaluation of semantic accuracy. This approach helps capture the meaning of the text rather than just use the exact word, making METEOR particularly effective at ensuring translational adequacy and fluency while maintaining semantic integrity across different languages and structures.

From Fig.8, we observe several key improvements across different subfigures. In Fig.8a, the BERTScore precision, recall, and F1 scores show significant enhancements for the 5G-aware versions compared to the default version, indicating effective assimilation of domain-specific 5G knowledge. Additionally, these 5G-aware LLMs outperform OpenAI's GPT-4[23], with improvements of 5%, 6%, and 4% in BERTScore's F1, precision, and recall metrics, respectively. Fig.8b and Fig.8c illustrate notable gains in the ROUGE-L and ROUGE-1 metrics for the 5G-aware models. The enhanced ROUGE-L precision, recall, and F1 scores in Fig.8b suggest better content and structure preservation. Similarly, the improved ROUGE-1 scores in Fig.8c reflect better content retention and contextual relevance. In both cases, the 5G-aware LLMs outperform GPT-4. Finally, Fig. 8d highlights superior performance in SemScore, BLEU, and METEOR metrics for the 5G-aware LLMs, indicating better alignment with reference texts and enhanced semantic accuracy. Overall, GPT-4's default version exhibits limitations in handling specialized topics like 5G, underscoring the need for fine-tuning. Additionally, GPT-4's proprietary nature poses accessibility and cost issues, emphasizing the importance of fine-tuning accessible LLMs to address knowledge gaps in evolving technological fields.

Fig. 9 shows the Perplexity [40] metric for the GPT-4 LLM and both the default and 5G-aware open-source LLMs, calculated using the generated outputs from the previous benchmark questions. Perplexity is a standard metric used to evaluate language models by measuring how well a model predicts a sample of text. It quantifies the model's uncertainty when generating or interpreting a sequence of words. Specifically, perplexity is the exponentiation of the average negative log-likelihood of the correct word sequence under the model's predicted probability distribution:

$$\mathrm{PPL}(p) = \exp\left(-\frac{1}{N}\sum_{i=1}^{N}\log p(w_i|w_1, w_2, \ldots, w_{i-1})\right) \quad (4)$$

where N is the total number of words in the sequence, and $p(w_i|w_1, w_2, \ldots, w_{i-1})$ represents the conditional probability of word w_i given the preceding words in the sequence. Lower perplexity indicates that the model is more confident and accurate in its predictions. For open-source LLMs, we used the corresponding tokenizer to calculate the score, whereas, for GPT-4, we used the GPT-2 tokenizer[24], as it is open-source. From the figure, we can see that the Perplexity score of the newly trained LLMs is lower than that of the default version. This means that 5G-aware LLMs were more confident in generating 5G-related responses, demonstrating that the training was successful (as a compelling example, Solar's perplexity decreased from 878.31 to 21.30 after training).

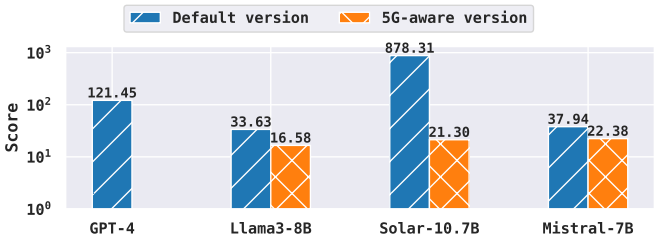


Fig. 9: Perplexity score.

Fig. 10 shows the mean generation time for GPT-4, the default, and 5G-aware versions of open-source LLMs for responses regarding the previous evaluation. From the figure, we can see that, despite the added internet latency, GPT-4 ranks first with an average time of 1.52 seconds compared to the default versions of the LLMs. However, open-source LLMs are as fast despite their smaller size. Moreover, we can see that our proposed fine-tuning does not affect generation time significantly, as the difference between the default versions and the 5G-aware versions is minimal. Nevertheless, since these times are longer than GPT-4's (except for 5G-aware Mistral), the research community should investigate inference speed techniques, such as [41], so that these LLMs can be used in 5G decision-making problems, which require fast decision speeds.

In addition, in Fig. 11, we aimed to precisely gauge the understanding of the resulting 5G-aware LLMs on the 5G

[23]gpt-4-0125-preview

[24]https://huggingface.co/openai-community/gpt2

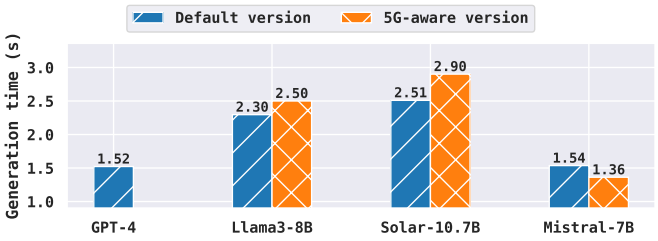


Fig. 10: Mean generation time on the Q/A evaluation dataset.

exam from OAI Instruct. This method allowed us to directly measure each model's comprehension of the material rather than just its ability to generate plausible text. The results from this examination reveal significant distinctions in model performance, particularly highlighting the superior understanding of 5G topics by 5G-aware LLMs compared to GPT-4, which scored 156 correct answers. Fine-Tuned Llama3, Solar-10.7B, and Mistral all demonstrated a higher number of correct responses, i.e., 186, 176, and 180, respectively, suggesting that the fine-tuning process has effectively enhanced their ability to grasp and accurately answer questions about specialized and technical content. This indicates not only an improved familiarity with the specific language and 5G concepts but also an enhanced capability to discriminate between closely related information. The Solar model initially struggled, not due to a lack of 5G knowledge but because of its difficulty in following instructions and handling multiple-choice questions. Post fine-tuning, Solar10.7B showed marked improvement in instruction adherence and began successfully answering such questions, thereby doubling its efficacy by integrating new 5G knowledge. Mistral, while proficient in following instructions, showed a noticeable deficiency in 5G-specific knowledge compared to newer models, reflecting its training focus on different domains.

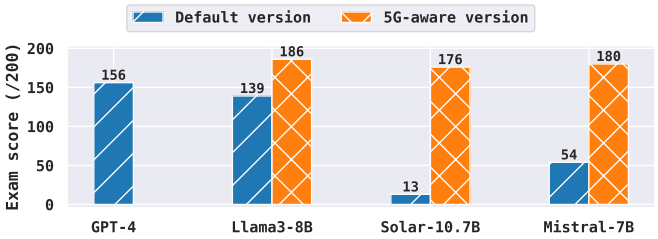


Fig. 11: Exam results of different LLMs.

4) ***Expert satisfaction****:* To enhance the evaluation of LLMs and gain more detailed insights into the quality of generated responses, we enlisted experts to verify the completions. These experts rated the quality of the LLMs' answers on a scale of 1 to 5. As illustrated in Table III, the overall results showed significant improvements in the Q/A parts of the 5G-aware LLMs compared to their default versions. Specifically, Llama3 ranks first with an improvement of ≈108%, followed by Solar with ≈100%, and Mistral with ≈90%. This showcases that trained LLMs can respond to 5G-related Q/A questions better than their default versions. This step, a critical part of our assessment methodology, provides an empirical measure of the LLMs' performance in generating accurate and relevant content within the field.

TABLE III: Improvement in Expert Satisfaction.

LLM	Default version (/5)	5G-aware version (/5)	Improvement (%)
Llama3-8B	1.55	3.22	107.74%
Solar-10.7B	1.25	2.50	100%
Mistral-7B	2.00	3.80	90%

C. Evaluation conclusion

The specialized LLMs developed in our study demonstrate strong potential for advanced future network management applications, such as making informed decisions in self-healing systems, detecting anomalies in logs, and managing self-regulating systems. These capabilities are crucial as we move towards 6G technologies, highlighting the importance of specialized LLMs in the future of telecommunications. Below, we provide some conclusions on: (i) the effect of dataset size on LLM quality, (ii) LLM cost-effectiveness, and (iii) the generalizability of newly created LLMs.

1) Dataset size efficiency: After fine-tuning three LLMs on our proof-of-concept dataset, OAI Instruct, we evaluated their ability to learn new 5G-specific knowledge and compared their performance to the more general but less specialized GPT-4 model. Our results show that our method effectively enables LLMs to acquire 5G knowledge, even with a relatively small training dataset of approximately 80 MB. Increasing the dataset size could further enhance the LLMs' specialization, potentially allowing them to cover the entire 5G domain. However, due to high costs, the current focus in both industry and research is on developing specialized rather than general LLMs [42].

2) LLM cost-effectiveness: Using 5G Instruct Forge, the first step is dataset generation, which was conducted with OpenAI's GPT-4 and Llama3 LLMs. To minimize costs, we can rely solely on free open-source LLMs to generate this dataset. In this regard, we utilized a single GPU with quantization techniques for an open-source LLM that is not inherently specialized in 5G. By adopting a RAG approach, we efficiently generated a domain-specific 5G dataset. Data generation can be accomplished in just a few minutes using a local open-source LLM running on a single GPU, making it a resource-efficient process. The second step is fine-tuning, which also requires only one GPU and takes approximately 15 hours. With a minimal investment of a few minutes for data generation and 15 hours for fine-tuning on one GPU, we developed a powerful domain-specific LLM with fewer than 10 billion parameters. For inference, the created LLM is compact and efficient, designed to run on a single GPU while consuming significantly less energy than larger LLMs. It is also local and free compared to GPT-4, outperforming it in 5G-related tasks. Thus, for 5G applications, we can confidently utilize these small, cost-effective models for various use cases, ensuring privacy and security, such as in local anomaly detection and

critical 5G information management on private networks. We firmly believe that creating domain-specific LLMs offers the most cost-effective solution, as described in [42].

3) LLM generalizability: It should be noted that the 5G-aware LLMs are trained on a set of 3GPP TSs using the freeze-tuning method, meaning only knowledge from these TSs is injected into the LLMs. They retain their previous capabilities while incorporating this additional knowledge. However, since LLMs can reason, if there is information in other TSs that can be deduced from the TSs used to train the LLM, we believe that the LLM can infer that information. Conversely, for new knowledge, these LLMs will not be able to provide a response. To adapt them, we need to use the 5G Instruct Forge again to create the embedding database from the new TSs and initiate fine-tuning to inject the new information into the LLM's knowledge. This way, they will retain both their old knowledge and acquire the new knowledge. Depending on the new TSs, the fine-tuning time will vary. As a reference, in our fine-tuning process, we used 22 TSs, which took approximately 15 hours; thus, the time required will differ based on the new set of TSs.

V. Conclusion

In this paper, we introduced the "5G Instruct Forge," a cutting-edge data engineering pipeline designed to improve LLM training using domain-specific datasets from 3GPP TSs. Our evaluations show that LLMs trained with our OAI Instruct dataset outperform conventional models like GPT-4 in 5G-specific tasks, demonstrating significant performance improvements. This work advances the use of LLMs in telecommunications and provides a framework that can be extended to other technological domains. Looking ahead, our research has important implications for the development of future networks like 6G, which will rely on technologies such as self-healing systems and zero-touch management systems. These advanced networks will depend on automation capabilities to enable dynamic and efficient network management without human intervention.

Acknowledgment

This work is supported partially by the European Union's Horizon Program under the 6G-Bricks (Grant No. 101096954) and the Sunrise-6G (Grant No. 101139257) projects.

References

[1] Wayne Xin Zhao et al. "A survey of large language models". In: *arXiv preprint arXiv:2303.18223* (2023).

[2] A. Gokaslan and V. Cohen. *OpenWebText Corpus*. Skylion007.github.io. 2019. URL: http://Skylion007.github.io/OpenWebTextCorpus.

[3] Christian Buck, Kenneth Heafield, and Bas Van Ooyen. "N-gram Counts and Language Models from the Common Crawl." In: *LREC*. Vol. 2. 2014, p. 4.

[4] Luca Soldaini et al. "Dolma: An Open Corpus of Three Trillion Tokens for Language Model Pretraining Research". In: *arXiv preprint arXiv:2402.00159* (2024).

[5] Erik Cambria and Bebo White. "Jumping NLP Curves: A Review of Natural Language Processing Research [Review Article]". In: *Computational Intelligence Magazine, IEEE* 9 (May 2014), pp. 48–57. DOI: 10.1109/MCI.2014.2307227.

[6] Hyung Won Chung et al. *Scaling Instruction-Finetuned Language Models*. 2022. DOI: 10.48550/ARXIV.2210.11416. URL: https://arxiv.org/abs/2210.11416.

[7] Gemma Team et al. "Gemma: Open models based on gemini research and technology". In: *arXiv preprint arXiv:2403.08295* (2024).

[8] Zoran Bojkovic et al. "6G ultra-low latency communication in future mobile XR applications". In: *Advances in Signal Processing and Intelligent Recognition Systems: 6th International Symposium, SIRS 2020, Chennai, India, October 14–17, 2020, Revised Selected Papers 6*. Springer. 2021, pp. 302–312.

[9] Nhu-Ngoc Dao et al. "A review on new technologies in 3GPP standards for 5G access and beyond". In: *Computer Networks* (2024), p. 110370.

[10] Yiheng Liu et al. "Understanding llms: A comprehensive overview from training to inference". In: *arXiv preprint arXiv:2401.02038* (2024).

[11] Abdelkader Mekrache, Adlen Ksentini, and Christos Verikoukis. "Intent-Based Management of Next-Generation Networks: an LLM-centric Approach". In: *IEEE Network* (2024), pp. 1–1. DOI: 10.1109/MNET.2024.3420120.

[12] Hao Zhou et al. "Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities". In: *arXiv preprint arXiv:2405.10825* (2024).

[13] AI@Meta. "Llama 3 Model Card". In: (2024). URL: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md.

[14] Jinze Bai et al. "Qwen Technical Report". In: *arXiv preprint arXiv:2309.16609* (2023).

[15] Albert Q Jiang et al. "Mixtral of experts". In: *arXiv preprint arXiv:2401.04088* (2024).

[16] Jinhyuk Lee et al. "BioBERT: a pre-trained biomedical language representation model for biomedical text mining". In: *Bioinformatics* 36.4 (Sept. 2019), pp. 1234–1240. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btz682. eprint: https://academic.oup.com/bioinformatics/article-pdf/36/4/1234/48983216/bioinformatics_36_4_1234.pdf. URL: https://doi.org/10.1093/bioinformatics/btz682.

[17] Yi Yang, Mark Christopher Siy Uy, and Allen Huang. "Finbert: A pretrained language model for financial communications". In: *arXiv preprint arXiv:2006.08097* (2020).

[18] Hugo Touvron et al. "Llama: Open and efficient foundation language models". In: *arXiv preprint arXiv:2302.13971* (2023).

[19] Ning Ding et al. "Enhancing chat language models by scaling high-quality instructional conversations". In: *arXiv preprint arXiv:2305.14233* (2023).

[20] Subhabrata Mukherjee et al. "Orca: Progressive learning from complex explanation traces of gpt-4". In: *arXiv preprint arXiv:2306.02707* (2023).

[21] Colin Raffel et al. "Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer". In: *Journal of Machine Learning Research* 21.140 (2020), pp. 1–67. URL: http://jmlr.org/papers/v21/20-074.html.

[22] Long Ouyang et al. "Training language models to follow instructions with human feedback". In: *Advances in neural information processing systems* 35 (2022), pp. 27730–27744.

[23] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. "Data augmentation for low-resource neural machine translation". In: *arXiv preprint arXiv:1705.00440* (2017).

[24] Heereen Shim et al. "Synthetic Data Generation and Multi-Task Learning for Extracting Temporal Information from Health-Related Narrative Text". In: *Proceedings of the Seventh Workshop on Noisy User-generated Text (W-NUT 2021)*. Ed. by Wei Xu et al. Online: Association for Computational Linguistics, Nov. 2021, pp. 260–273. DOI: 10.18653/v1/2021.wnut-1.29. URL: https://aclanthology.org/2021.wnut-1.29.

[25] Mitesh Mangaonkar and Venkata Karthik Penikalapati. "Enhancing Production Data Pipeline Monitoring and Reliability through Large Language Models (LLMs)". In: *Eduzone International peer reviewed/refereed academic multidisciplinary journal* 13 (Jan. 2024), pp. 51–56.

[26] Ajay Patel, Colin Raffel, and Chris Callison-Burch. "DataDreamer: A Tool for Synthetic Data Generation and Reproducible LLM Workflows". In: *arXiv preprint arXiv:2402.10379* (2024).

[27] Nihal V Nayak et al. "Learning to generate instruction tuning datasets for zero-shot task adaptation". In: *arXiv preprint arXiv:2402.18334* (2024).

[28] Hang Zou et al. "TelecomGPT: A Framework to Build Telecom-Specfic Large Language Models". In: *arXiv preprint arXiv:2407.09424* (2024).

[29] Ali Maatouk et al. “Teleqna: A benchmark dataset to assess large language models telecommunications knowledge”. In: *arXiv preprint arXiv:2310.15051* (2023).

[30] Yaowei Zheng et al. “LlamaFactory: Unified Efficient Fine-Tuning of 100+ Language Models”. In: *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*. Bangkok, Thailand: Association for Computational Linguistics, 2024. URL: http://arxiv.org/abs/2403.13372.

[31] Tianyi Zhang et al. “Bertscore: Evaluating text generation with bert”. In: *arXiv preprint arXiv:1904.09675* (2019).

[32] Ansar Aynetdinov and Alan Akbik. “SemScore: Automated Evaluation of Instruction-Tuned LLMs based on Semantic Textual Similarity”. In: *arXiv preprint arXiv:2401.17072* (2024).

[33] Ning Ding et al. “Parameter-efficient fine-tuning of large-scale pre-trained language models”. In: *Nature Machine Intelligence* 5.3 (2023), pp. 220–235.

[34] Edward J Hu et al. “Lora: Low-rank adaptation of large language models”. In: *arXiv preprint arXiv:2106.09685* (2021).

[35] Dan Hendrycks et al. “Measuring massive multitask language understanding”. In: *arXiv preprint arXiv:2009.03300* (2020).

[36] Chin-Yew Lin. “ROUGE: A Package for Automatic Evaluation of Summaries”. In: *Text Summarization Branches Out*. Barcelona, Spain: Association for Computational Linguistics, July 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013.

[37] Kaitao Song et al. “Mpnet: Masked and permuted pre-training for language understanding”. In: *Advances in neural information processing systems* 33 (2020), pp. 16857–16867.

[38] Kishore Papineni et al. “Bleu: a Method for Automatic Evaluation of Machine Translation”. In: *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Ed. by Pierre Isabelle, Eugene Charniak, and Dekang Lin. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, July 2002, pp. 311–318. DOI: 10.3115/1073083.1073135. URL: https://aclanthology.org/P02-1040.

[39] Alon Lavie and Abhaya Agarwal. “Meteor: an automatic metric for MT evaluation with high levels of correlation with human judgments”. In: *Proceedings of the Second Workshop on Statistical Machine Translation*. StatMT ’07. Prague, Czech Republic: Association for Computational Linguistics, 2007, pp. 228–231.

[40] Fred Jelinek et al. “Perplexity—a measure of the difficulty of speech recognition tasks”. In: *The Journal of the Acoustical Society of America* 62.S1 (1977), S63–S63.

[41] Piotr Nawrot et al. “Dynamic memory compression: Retrofitting llms for accelerated inference”. In: *arXiv preprint arXiv:2403.09636* (2024).

[42] Zheng Zhang et al. “Balancing specialized and general skills in llms: The impact of modern tuning and data strategy”. In: *arXiv preprint arXiv:2310.04945* (2023).

Azzedine Idir Ait Said is a computer systems engineer who graduated with honors from the École Supérieure d’Informatique, Algiers. He completed a research internship at EURECOM, focusing on integrating large language models (LLMs) for anomaly detection in 5G/6G networks under Prof. Adlen Ksentini and Abdelkader Mekrache. He holds an engineering diploma and a master’s in computer science, with research interests in artificial intelligence, 5G/6G technologies, and advancing LLM integration in complex network environments.

Abdelkader Mekrache (Member, IEEE) is a PhD candidate at EURECOM’s Communication Systems Department. His primary focus is on advanced network management frameworks in next-generation wireless networks under the supervision of Prof. Adlen Ksentini. He is an active participant in collaborative research and notably contributes to the OAI project, as well as multiple European projects, including 6G-Bricks, 6G-Intense, and Sunrise-6G.

Karim Boutiba (Member, IEEE) is a researcher at the Communication Systems Department of EURECOM. He is working toward upgrading 5G systems to 6G using EURECOM’s OpenAirInterface (OA)I-based test platform. He obtained his PhD degree from Sorbonne University in 2024. He was involved in collaborative research projects and an active contributor to the OAI projects. His research interests include Next-Generation Networking, 5G New Radio (NR), Network Slicing, Open RAN, Optimization algorithms and Reinforcement Learning for 5G networks and beyond.

Kostas Ramantas has received the Diploma of Computer Engineering, the MSc degree in Computer Science and the PhD degree from the University of Patras, Greece, in 2006, 2008 and 2012 respectively. He has been the recipient of two national scholarships. In June 2013, he joined IQUADRAT as a senior researcher and has co-supervised 4 PhD students. He was involved in multiple E.C. funded projects as a WPL, TM, and PC. He has published more than 35 journal and conference papers.

Adlen Ksentini (Senior Member, IEEE) is a professor in the Communication Systems Department of EURECOM. He is leading the Network softwarization group activities related to Network softwarization, 5G/6G, and Edge Computing. Adlen Ksentini’s research interests are Network Sofwerization and Network Cloudification, focusing on topics related to network virtualization, Software Defined Networking (SDN), and Edge Computing for 5G and 6G networks. He has been participating to several H2020 and Horizon Europe projects on 5G and beyond, such as 5G!Pagoda, 5GTransformer, 5G!Drones, MonB5G, ImagineB5G, 6GBricks, 6G-Intense, Sunrise-6G and AC3. He is the technical manager of 6G-Intense and AC3, on zero-touch management of 6G resources and applications, and Cloud Edge Continuum, respectively. He is interested in the system and architectural issues but also in algorithm problems related to those topics, using Markov Chains, Optimization algorithms, and Machine Learning (ML). Adlen Ksentini has given several tutorials in IEEE international conferences, IEEE Globecom 2015, IEEEE CCNC 2017/2018/2023, IEEE ICC 2017, IEEE/IFIP IM 2017, IEEE School 2019. Adlen Ksentini is a member of the OAI board of directors, where he is in charge of OAI 5G Core Network and O-RAN management (O1, E2) for OAI RAN activities.

Moufida Rahmani is a Associate Professor at the National Higher School of Computer Science (ESI) in Algiers, Algeria. She has been teaching there since October 2021. She obtained her Bachelor’s degree in Computer Science in July 2009 and her Master’s degree in Networking and Distributed Systems in July 2011, both from the University of Sciences and Technology Houari Boumediene (USTHB) in Algiers, Algeria. She pursued her Ph.D. at the same university and successfully defended her doctoral thesis in November 2020. Since 2011, she has been a member of the Mobility research team at the Laboratory of Computer Systems (LSI) at USTHB. Her research interests include computer networks and distributed systems, wireless sensor networks, drone networks, 5G, IoT, and related technologies.