WiFiQnA: a WiFi Dataset for Large Language Models

Anouar Zouhri*, Lynda Zitoune†, Iyad Lahsen-Cherif*
*INPT, CS department, Rabat, Morocco. Email: zouhri.anouar@master.inpt.ac.ma, lahsencherif@inpt.ac.ma
†Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des Signaux et Systèmes
3, rue Joliot Curie, 91190 Gif-sur-Yvette, France. Email: lynda.zitoune@l2s.centralesupelec.fr

***Abstract*—The increasing complexity of modern WiFi networks aligns them more closely with cellular systems. This convergence underscores the need for WiFi-specific LLMs, akin to ongoing efforts in 5G. An essential initial step is the design of a WiFi dataset compatible with LLM requirements, structurally coherent, and containing both technical and general information to ensure broad applicability. This work introduces WiFiQnA, a curated dataset of WiFi-related multiple-choice questions designed for LLM fine-tuning. We define two multiple-choice question (MCQ) formats: general knowledge and procedural configuration/troubleshooting questions. We develop a multi-step generation framework using three LLMs for question generation and four for validation. The process integrates filtered telecom datasets, WiFi-specific sources, and tailored prompts, ensuring semantic diversity, accuracy, and reliability.**

***Index Terms*—WiFi networks, LLMs, Dataset generation, multiple-choice question (MCQ).**

I. Introduction

The utilization of Large Language Models (LLMs) is a new trend applied in the telecommunications field, to solve issues related to network resources management by using a self-improvement method with a reward function to understand the user feedback [1]. The goal is to guide the model in providing responses that match human preferences [2]. Hence, LLM models can optimize power allocation, assist the spectrum detection, and understand protocols using techniques such as black box optimizer [3, 4]. Furthermore, they can generate heuristic algorithms to optimize resource allocation [5]. It is a powerful tool for complex telecom-related issues, and several efforts have been made in this direction [6]. However, the telecoms field is vast, encompassing various wireless technologies and deployments. While the underlying challenges related to complexity, optimization, planning, and debugging are similar across these communication models, the solutions remain inherently dependent on their intrinsic features. Modern WiFi networks experience an increasing complexity due to the integration of advanced capabilities such as MU-MIMO, Multi-AP coordination, and Multi-band operation, bringing them closer to cellular networks, to accommodate high device densities across large areas and diversity of service demands. Therefore, there is a growing need for the development of LLMs tailored for WiFi, similar to the enthusiasm currently seen for 5G and upcoming 6G networks. Without such advancements, the interfaces defined by standardization bodies [7] as gateways become obstacles or bottlenecks, hindering the technological convergence between the two worlds.

It is crucial at the first step to identify and design a suitable dataset tailored for Wi-Fi networks [8]. This dataset should be compatible with LLMs, meaning it must be understandable, easy to split, and tokenize. Furthermore, it should contain accurate technical information as well as general data for its universality. Regarding the related work, six open-source datasets are available for free utilization. Two are designed for telecoms in general, such as TeleData [9] and TeleEval [9]. TspecLLM [10] compiled from 3GPP standard and SpecG 5G [11] for 5G networks, split into two datasets: 5Sum for summaries and 5SC security-based classification purposes. TeleQnA [12] contains 9999 multiple-choice questions (MCQs) with just 531 MCQs, insufficient for training dedicated LLMs for WiFi. This limitation motivates us to create a new dataset called WiFiQnA.

The main contribution of this work is presents the architecture and the process we developed to generate the WiFiQnA dataset. First, we explored existing telecom datasets and data sources (articles, books, blogs) related to WiFi standards. Then, we designed two MCQ formats: one for general WiFi knowledge, and another focused on configuration and troubleshooting, where answers are step-by-step procedures. The dataset was generated using an architecture that combines three approaches involving three different LLMs and validated using four different LLMs. We followed three main steps to generate MCQs on WiFi. We first prepare and filter existing datasets, WiFi-oriented external sources, and leverage existing LLM knowledge. We define two tailored prompts to guide question generation, one for each format. Finally, for the validation step, eliminating duplicates, checking similarity, and verifying answer accuracy, we use four LLMs to ensure reliability and correctness by filtering out incorrect or misleading questions.

This paper is organized as follows. Section II provides an overview of relevant open-source telecom datasets, highlighting their main characteristics, use cases, and setup procedures. In Section III, we present the WiFiQnA dataset, focusing in particular on the structure of the adopted MCQ format. Section IV outlines the methodology used to generate the dataset, describing in detail the three complementary approaches and the overall architecture supporting the generation process.

979-8-3503-9281-4/25/$31.00 ©2025 IEEE

Section V illustrates a practical scenario involving model selection and filtering strategies. The results of our approach are discussed in Section VI, emphasizing the properties and strengths of the final dataset. Finally, Section VII summarizes our contributions and outlines directions for future research using the WiFiQnA dataset.

II. Related work

Since 2023, several datasets have been proposed in the literature to speed up the development of LLMs for telecoms. Here, we focus on existing open-source datasets and summarize the steps followed for designing them and their corresponding data sources. TeleData [9] is derived from four main sources: scientific papers from arXiv, 3GPP standards, Wikipedia articles, and websites taken from Common Crawl dumps. Using keywords, these sources are filtered using large language models (LLMs) to find relevant telecommunication materials and topics. Afterwards, the data is cleaned, formatted, and standardized. TeleData is about 2.5 billion tokens, designed to train LLMs for telecommunications tasks. It is composed of four parts: the *ID*, a unique identifier for each data sample; the *Category*, for the sample type; the *Content* includes the full text of the sample; and the Metadata, extra information about the sample, stored as a JSON object [9]. TeleEval [9], a dataset generated from TeleData, contains 750 K question-and-answer pairs about general telecommunications knowledge and standards. This dataset was created using an LLM-based framework by giving paragraphs of TeleData to the Mixtral-8x7B-Instruct model [13]. The dataset has three parts: the *Statement* (the question), the *Answer*, and the *ID*, to show which part of TeleData was used to create the question and answer. The SPEC5G dataset [11] for 5G network specifications contains 3,5 million sentences and 134 million words, derived from 13 K documents on cellular networks and 13 websites. It is split into two sub-datasets: 5GSum and 5GSC for summarizing and classification tasks, respectively. The TSpec-LLM dataset [10] is derived from the 3GPP standards (Release 8 to Release 19) converted into Markdown and DOCX formats. It contains 30 K documents and 535 million words. TSpec-LLM keeps all the pages, unlike other datasets like SPEC5G, which remove some information like tables and figures. It also changes math formulas into LaTeX to make them easy to read and analyze. This dataset is useful for telecommunications research and machine learning applications, such as using it in the context of RAG research [10]. The ORAN-Bench-13K is designed to evaluate the performance of LLMs in Open Radio Access Networks (O-RAN). It includes 14 K MCQs (Multi Choice Questions) derived from 116 O-RAN specification documents, categorized into three difficulty levels: easy, intermediate, and difficult. The data structure includes a unique identifier for each question, the difficulty level, the question text, a list of answer choices (typically four), and the correct answer [14]. The TeleQNA dataset designed to assess LLM knowledge in telecommunications [12], contains 10 K MCQs organized into five categories: *Lexicon* covering general telecoms terminology; *Research Overview* providing broad insights into telecoms research; *Research Publications*, focusing on in-depth inquiries from conference proceedings and journal transactions; *Standards* summarizing telecoms standards from 3GPP and IEEE; and *Standards Specifications*, detailing the technical specifications of telecoms systems. TeleQNA is the only dataset that involves the IEEE standards and specifications, with its last two categories being particularly relevant to our work, as they contain detailed information on WiFi networks. All these datasets are classified in the table I, depending on the targeted tasks and the training data size.

TABLE I: Summary of Telecom Datasets

Dataset	Task	Size
TeleData	Pre-training NLP models for telecom	2.5B tokens
TeleEval	QA generation and evaluation for telecom	750K QA pairs
SPEC5G	Analysis of 5G protocol specifications	3.5M sentences, 134M words
5GSum	5G technical texts summarization	713 articles
5GSC	5G texts classification for security	2,401 sentences
TSpec-LLM	Processing 3GPP documentation	535 million words
ORAN-Bench-13K	Benchmarking LLMs for O-RAN	13.9K MCQs
TeleQNA	Evaluation of LLM knowledge in telecom	10K MCQs

After analyzing TeleQnA, we identified only 543 WiFi-related questions, which is insufficient to tune an LLM. TeleData contains raw information in the form of paragraphs and needs to be processed as done in TeleEval. However, TeleEval still remains general, as it only contains questions and their answers without explanation or demonstration of how the correct answer was chosen. Another point that remains unexplored within the listed datasets is the resolution of practical issues and configurations. None of them includes questions or data for configuration or resolution steps for a typical use case or an issue. Based on these observations, we focused in this work to create a more complete and specific dataset for WiFi networks, covering almost all versions from WiFi 1 to WiFi 6, and going further into configuration questions.

III. WiFiQnA Dataset format

As derived from the TeleQnA dataset, WiFiQnA naturally contains multiple-choice questions, a well-suited and well-structured format for LLMs of the major dataset cited in the table I. Each QA is composed of a question, options, an answer, and the associated explanations. **The Question** is represented in two types: *(i)* direct question to address standard information and general knowledge about IEEE 802.11 and WiFi. *(ii)* Troubleshooting or configuration questions reflecting real-world situations. With such types of questions, more information needs to be added to the dataset regarding device configuration, network setup, as well as fault, troubleshooting, and configuration issues. **The options** include one correct answer and three incorrect answers, whatever the question type. The incorrect answers provide additional information and variation, enriching the dataset. To maintain a balance between difficulty and feasibility, we have fixed the number of choices at four for the MCQs. Offering four options allows for a thorough assessment of knowledge while simplifying the

decision-making process. Adding extra options could lead to confusion. For a direct question, each option is composed of a sentence that directly answers the question. This is usually the most common format for MCQs, box 1 illustrates this type of option. Whereas, for the troubleshooting question, an option is a combination of steps to follow in order to complete the required task, as shown in the box 2. **The answer** indicates the correct choice among the four proposed options. It includes the word option and a number indicating the ID of the correct option, followed by the answer phrase describing the selected option. Sometimes, it also contains a paragraph to demonstrate and explain the correctness of the selected option, meaning the answer indicates the chosen option and its context. Finally, **the explanation** part contains a detailed and technical explanation to answer the question and explain why the correct answer is chosen among the options. Furthermore, it provides additional information to add insights to the dataset. It contains important and detailed information to prove that the chosen option is correct. Some examples of such MCQ are given in section VI to highlight these new formats.

Box 1 : Direct Question format

Question: The question text
- *Option 1*: Answer A
- *Option 2*: Answer B
- *Option 3*: Answer C
- *Option 4*: Answer D

Answer: *Option 1*: Option A
Explanation: Detailed explanation of why the proposed answer is the correct one.

Box 2 : Configuration Question format

Question: The configuration question text
- *Option 1*:
  - Step 1: ... - Step 2: ... - Step 3: ... - Step 4: ...
- *Option 2*:
  - Step 1: ... - Step 2: ... - Step 3: ...
- *Option 3*:
  - Step 1: ... - Step 2: ... - Step 3: ...
- *Option 4*:
  - Step 1: ... - Step 2: ... - Step 3: ... - Step 4: ... - Step 5: ...

Answer: *Option X*, with steps to perform the configuration.
Explanation: Detailed technical explanation of why this option is the best.

IV. Dataset generation: general architecture

To build a comprehensive dataset that encompasses the most important aspects of WiFi networks, we adopt a balanced approach by combining sources on WiFi versions [8] with the IEEE 802.11 protocol stack [15]. To achieve this, we develop a general architecture that implements three processes, which can be used separately or together, enabling the generation of a diverse range of MCQs while minimizing redundancy and accommodating limited computing resources if necessary. The proposed architecture operates independently of the LLM models and available computing resources. Nevertheless, the main consideration focuses on utilizing LLM APIs where questions are generated via a structured prompt that aligns with our question formats. Other criteria, such as the number of supported tokens, whether the API model is free or not, and whether it is open-source or not, are to be considered. In the following, we outline these processes and provide details of the steps we followed: data resource preparation, question generation, filtering for similarity detection and removal, and finally, the validation step.

A. Input data source preparation

We diversified the data sources to create a comprehensive and consistent dataset. We considered existing WiFi datasets like TeleEval [9] and TeleQnA dataset [12]. Additionally, we incorporated relevant documents, such as research journals, IEEE standards, conference articles, and books related to WiFi. Therefore, we define appropriate preparation and filtering for these data sources.

The TeleEval dataset [9] contains questions related to different versions of IEEE 802.11 in the form of statements and responses. Additionally, the TeleQnA dataset [12] contains 543 WiFi-related questions and answers in multiple-choice format. Therefore, we extract and gather all questions in a new file, and use it as a source to generate our dataset, as shown in the figure 1.

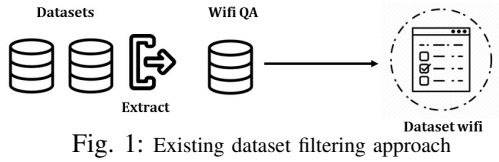

Fig. 1: Existing dataset filtering approach

For the documentary resources, such as scientific publications, standard documents, and books, we first convert PDF documents into text format to efficiently process all relevant information for better analysis. We use Optical Character Recognition (OCR) technology [16] for PDFs, scanned books, and technical reports to extract only the most relevant sections. Then, we use an LLM model to identify and group important paragraphs from different sources into a clean file. The main part of our approach is removing pages with irrelevant information, such as tables of contents and appendices. Additionally, we organize relevant paragraphs and keep a trace of their sources to identify the origin of each paragraph, useful to provide and to enrich the explanation part of the MCQ, as shown in Figure 2. However, processing a graph or an image is still tricky. The explanation of these graphs or images is incomplete in general, so these resources become useless. Converting the images and graphs themselves into paragraphs is cost-effective, since the dedicated models are not free like GPT-4-Vision or Claude 3 (Anthropic), and utilize a significant amount of resources.

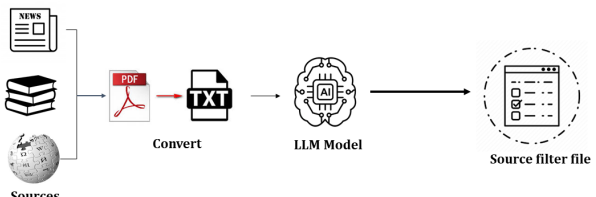

Fig. 2: Documentary sources filtering approach

B. The dataset generation

Figure 3 presents the complete architecture of the proposed framework. The proposed pipelines are to be general, reflecting a broad approach. The selection of LLMs (Large Language Model) depends on various factors, including available hardware resources, budget (for paid models), and time constraints, as new LLM models are released regularly, which can shift the landscape. Therefore, the steps outlined serve as fundamental principles, while the choice of model should be tailored to the specific application and tasks at hand. We first designed a structured prompt for the expected format of the generated data. The prompt is used to create both direct and configuration-related questions. Hence, two prompts are integrated into our code to ensure consistency in question generation. Moreover, to increase the number and diversity of questions, we proposed three different approaches:

1) Dataset-based question generation: The selected LLM model is instructed to generate new questions using the dataset obtained following the approach in Figure 1. The content of the API request used to connect to the LLM includes the predefined prompts with a detailed explanation of the question format to be generated, as well as the filtered dataset, which serves as a source of inspiration. Hence, the LLM model generates diverse questions on the same context, adopting the question format. The responses received by the model API contain all the requested questions, following the exact format defined in the prompt.

2) paragraph-based question generation: We use the same API technique and configuration as previously; we generate new questions considering the cleaned and filtered paragraph-based file as generated following the process in Figure 2. The only difference is the LLM model used for generating questions. To ensure that the model will generate the question based on those paragraphs, we also modified the prompt of the two question formats in our code by adding a script that demands the model to take information from each paragraph in the file and generate questions.

3) Model-based question generation: This approach did not rely on any pre-prepared input sources. Instead, we directly asked existing LLM models to generate questions based on their own knowledge, to enhance the diversity of our dataset. Testing several models allows us to test their capacity to create the maximum number of unique questions.

To further reduce duplicate questions, we used different LLM models for each approach. This strategy enables us to generate a wide variety of questions while also assessing each model's ability to create the maximum number of unique questions.

C. Filtering phase

To identify and eliminate similar questions and ensure minimal redundancy, we established a filtering method based on two techniques. **Direct repetition removal**, to eliminate identical questions (wording and meaning) that appear multiple times in the dataset by comparing the sentences of the questions. **AI-based similarity detection**, uses an LLM model to identify questions that are not identical but share similar wording or structure. The model operates with a similarity threshold. If the threshold is set too low, the model may classify partially similar questions as duplicates, including those that use common words but have different meanings. Conversely, if the threshold is set too high, only questions that have exactly the same meaning will be filtered out. To ensure effective filtering, we establish an optimal similarity threshold that allows the model to remove only questions with the same meaning, with the same answer and explanation.

D. Validation phase

We utilize LLM models to verify the accuracy of the generated questions. This process involves prompting the LLM model to determine whether the answer generated for each question is true, based on the model's knowledge. It is judicious to use more than two models for validation to enhance its reliability. A majority vote among the models is then used to decide if the generated answer is correct.

V. Application: WiFiQnA dataset generation

The steps discussed in the previous section outline the roadmap we followed to create the WiFiQnA dataset. Each step is essential for generating relevant questions that provide valuable information for training or fine-tuning LLM models tailored to WiFi networks. For the generation phase, we have selected three LLM models, primarily based on their release dates and performance. These models, as Gemini and Gamma 2-9B from Google and Qwen 2.5 from Alibaba, are among the latest released as of the time this article is written. Gamma adapted to generate WiFi-related questions using its knowledge, and Qwen 2.5, suitable to create questions based on texts and paragraphs. Additionally, they meet the token count criteria, enabling the generation of multiple questions with a single API request. *Gemini Flash 2.0* [17], the last version released, offers advantages such as higher speed, multi-modal generation, and a context window of one million tokens. We apply this model in the first approach, fed by the extracted questions' dataset. *Qwen 2.5* [18], also based on a decoder-only architecture, includes models ranging from 0.5 billion to 72 billion parameters and was trained on 18 trillion tokens. In our case, we chose the model with 32B parameters to get as many questions as possible for the second approach, i.e., fed by the extracted sources' dataset. *Gemma2-9B* [19], based on a decoder-only transformer architecture, has 9 billion parameters and was trained on 8 billion tokens, mainly using data in the English language. We apply this model to generate questions based on its own knowledge.

For filtering purposes, we applied the *all-MiniLM-L6-v2* moedl capable of comparing the similarity between questions and removing duplicates using the cosine similarity method [20]. The model converts the questions into tokens, generates vector representations, and then calculates their cosine similarity. To assess similarity, it is necessary to establish a threshold that defines whether we want to detect similarity in wording or meaning. In our case, we selected a high similarity threshold

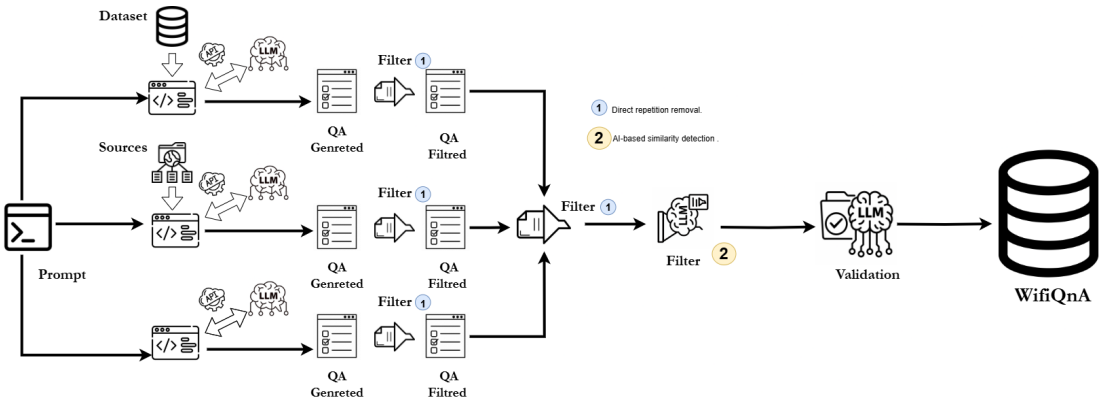

Fig. 3: The framework architecture for the dataset generation

of 0.95 to avoid removing questions with slight differences related to WiFi versions. Therefore, selecting the appropriate similarity threshold is a critical element in our approach. Finally, for the same reasons related to the number of tokens, stability, and reliability in generation, we utilized the same LLMs, Gemini Flash 2.0 and Gemma2-9B, for the validation phase. Additionally, we added two other models. The first model is Qwen Turbo [21], which is based on Qwen 2.5 and can handle an input length of up to 1 million tokens. The second model is Gemini Flash 1.5 8B [22], an earlier version of Gemini Flash 2.0. It offers high efficiency and delivers responses more rapidly even with few parameters. Introducing different models (four in our case) improves the accuracy of the validation. The additional two models prevent the issue where the models used for generating the questions would also validate them, which could lead to incorrect questions being mistakenly considered correct.

VI. Test and obtained results

Our test approach involves asking the three LLM models we utilized in each method to generate both the configuration and the direct questions simultaneously. This process ensures that the questions benefit from all the steps of the previous three approaches. All questions resulting from the three approaches are merged in a single file to form 12,097 MCQ of the WiFiQnA dataset. Table II gives a classification of the obtained questions regarding the question type (4,033 configuration questions and 8,064 direct questions) and the used approach.

After the filtering step, using the first filter to remove duplicate questions, and the second filter with *all-MiniLM-L6-v2* model to identify and remove lexically similar questions, only 11,994 questions were retained. The application of the four models *Gemini Flash 2.0*, *Gemma2-9B*, *Qwen Turbo* and *Gemini Flash 1.5B*, to check the intersection of incorrect question IDs between the results of each model, allows to identification of 514 incorrect questions with false answers such that 41 configuration questions and 473 direct questions. This distribution seems reasonable due to the significant number of direct questions compared to configuration questions. Ultimately, our dataset contains 11,480 correct questions, comprising 3,960 questions for configuration and 7,520 for direct questions.

TABLE II: Number of MCQs and classification

	Configuration MCQs	Direct MCQs
Approach 1	904	1,625
Approach 2	2,113	4,278
Approach 3	1,016	2,161
Total Question (TQ)	4,033	8,064
TQ after filtering	4,001	7,993
TQ after validation	3,960	7,520

The difference between the number of direct and the number of configuration questions is related to the number of tokens and words processed by the LLM model. Direct questions have fewer words because of their format, their options are short, and the explanation does not exceed one sentence. An example is given in Box 3. However, configuration questions present complete step-by-step procedures to follow, as illustrated in Box 4. Hence, the APIs of the three LLM models, Gemini Flash 2.0, Gemma2-9B, and Qwen 2.5, restrict the number of tokens that can be generated in this case.

As a result, the number of questions depends on the LLM models and the input data sources. As observed, the second approach generates the highest number of questions due to the rich input data. A single PDF file or a well-structured paragraph can yield multiple questions. The model can extract and rephrase the content from a paragraph into question-answer pairs, either in multiple-choice format or through stepwise decomposition, and provide incorrect answer options. In contrast, the first approach, based on a fixed set of 531 questions, is limited to generating additional questions through rephrasing or simple recombination of existing questions. The final approach involves generating questions independently of a given input source; the LLM model generates questions based on its knowledge. However, the questions produced using these approaches undergo filtering and validation processes to ensure their accuracy.

Box 3 : Direct Question Example

Question 6020: In IEEE 802.11ax, what mechanism is introduced to enhance Quality of Service, by providing scheduled transmission opportunities that reduce contention and improve the CSMA/CA system throughput?

- *Option 1*: Spatial Reuse
- *Option 2*: BSS Color
- *Option 3*: Ultra-reliable Low Latency Communication
- *Option 4*: Target Wake Time (TWT)

Answer: *Option 4*: Target Wake Time (TWT)
Explanation: The Target Wake Time mechanism in IEEE 802.11ax is designed to reduce the power consumption of devices while improving network efficiency.

Box 4 : Configuration Question Example

Question 2: You need to set up a new WiFi network using IEEE 802.11a in a crowded office environment with potential interference from other wireless devices. How do you configure the Physical Layer settings to ensure optimal performance?

- *Option 1*:
  - Step 1: Set the modulation type to BPSK.
  - Step 2: Choose the 5 GHz frequency band.
- *Option 2*:
  - Step 1: Set the modulation type to OFDM.
  - Step 2: Choose the 5 GHz frequency band.
  - Step 3: Enable DFS to avoid congested channels.
- *Option 3:*
  - Step 1: Set the modulation type to QPSK.
  - Step 2: Choose the 2.4 GHz frequency band.
  - Step 3: Enable WMM for multimedia applications.
- *Option 4:*
  - Step 1: Set the modulation type to QAM.
  - Step 2: Choose the 5 GHz frequency band.

Answer: *Option 2*
Explanation: Here's why option 2 is the best choice:

- **OFDM:** offering higher data rates and better resistance to interference, and **5 GHz Frequency Band: offers wider bandwidth and less congestion compared to the 2.4 GHz band, DFS:** is essential for avoiding interference with other wireless systems operating in the 5 GHz band.

VII. Conclusion

The absence of LLM-compliant WiFi datasets poses a significant challenge in developing LLMs for such networks, which are increasingly complex nowadays. Therefore, this work contributes to designing and implementing a framework for generating the WiFiQnA dataset. Two multiple-choice question (MCQ) formats are defined: one focusing on general WiFi knowledge and the other on configuration and troubleshooting, which includes step-by-step answers. The dataset generation follows a three-stage architecture, utilizing different LLMs for question generation and four LLMs for validation. The process consists of three steps: (1) selecting and filtering relevant data sources, (2) prompt design tailored for each MCQ type, and (3) multi-model validation to eliminate duplicates, evaluate semantic similarity, and ensure the accuracy of answers. The proposed pipeline offers a general framework adaptable to various LLMs, with model selection influenced by hardware, budget, and time constraints. Given the rapid evolution of LLMs, this approach establishes foundational principles while allowing flexibility for task-specific customization. In future work, we expect to utilize the WiFiQnA dataset for fine-tuning LLM models focused on WiFi-related management tasks.

References

[1] Sulaiman Aftan and Habib Shah. "Using the arabert model for customer satisfaction classification of telecom sectors in saudi arabia". In: *Brain Sciences* 13.1 (2023), p. 147.

[2] Minae Kwon et al. "Reward design with language models". In: *arXiv preprint arXiv:2303.00001* (2023).

[3] Serena Booth et al. "The perils of trial-and-error reward design: misdesign through overfitting and invalid task specifications". In: *Proc. of the AAAI Conference on Artificial Intelligence*. Vol. 37. 5. 2023, pp. 5920–5929.

[4] 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).

[5] Michal Pluhacek et al. "Leveraging large language models for the generation of novel metaheuristic optimization algorithms". In: *Proc. of the Companion Conference on Genetic and Evolutionary Computation*. 2023, pp. 1812–1820.

[6] Humza Naveed et al. "A comprehensive overview of large language models". In: *arXiv preprint arXiv:2307.06435* (2023).

[7] Xingqin Lin. "An overview of 5G advanced evolution in 3GPP release 18". In: *IEEE Communications Standards Magazine* 6.3 (2022), pp. 77–83.

[8] Fen Liu et al. "Survey on WiFi-based indoor positioning techniques". In: *IET communications* 14.9 (2020), pp. 1372–1383.

[9] Ali Maatouk et al. "Tele-LLMs: A series of specialized large language models for telecommunications". In: *arXiv preprint arXiv:2409.05314* (2024).

[10] Rasoul Nikbakht, Mohamed Benzaghta, and Giovanni Geraci. "Tspec-llm: An open-source dataset for llm understanding of 3gpp specifications". In: *arXiv preprint arXiv:2406.01768* (2024).

[11] Imtiaz Karim et al. "SPEC5G: A dataset for 5G cellular network protocol analysis". In: *arXiv preprint arXiv:2301.09201* (2023).

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

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

[14] Pranshav Gajjar and Vijay K Shah. "Oran-bench-13k: An open source benchmark for assessing llms in open radio access networks". In: *arXiv preprint arXiv:2407.06245* (2024).

[15] Hua Zhu et al. "A survey of quality of service in IEEE 802.11 networks". In: *IEEE wireless communications* 11.4 (2004), pp. 6–14.

[16] Ray Smith. "An overview of the Tesseract OCR engine". In: *9th Int. conference on document analysis and recognition*. Vol. 2. IEEE. 2007, pp. 629–633.

[17] Omer Aydin et al. "Generative AI in Academic Writing: A Comparison of DeepSeek, Qwen, ChatGPT, Gemini, Llama, Mistral, and Gemma". In: *arXiv preprint arXiv:2503.04765* (2025).

[18] An Yang et al. "Qwen2. 5 technical report". In: *arXiv preprint arXiv:2412.15115* (2024).

[19] Gemma Team et al. "Gemma 2: Improving open language models at a practical size". In: *arXiv preprint arXiv:2408.00118* (2024).

[20] Baoli Li and Liping Han. "Distance weighted cosine similarity measure for text classification". In: *Intelligent Data Engineering and Automated Learning–IDEAL 2013*. Springer. 2013, pp. 611–618.

[21] Rui Yang et al. "Enhancing text-based knowledge graph completion with zero-shot large language models: A focus on semantic enhancement". In: *Knowledge-Based Systems* 300 (2024), p. 112155.

[22] Gemini Team et al. "Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context". In: *arXiv preprint arXiv:2403.05530* (2024).