RESEARCH-ARTICLE

Can we make FCC Experts out of LLMs?

ATUL BANSAL, Carnegie Mellon University, Pittsburgh, PA, United States

VERONICA MURIGA, Carnegie Mellon University, Pittsburgh, PA, United States

JASON LI, Carnegie Mellon University, Pittsburgh, PA, United States

LUCY DUAN, Carnegie Mellon University, Pittsburgh, PA, United States

SWARUN KUMAR, Carnegie Mellon University, Pittsburgh, PA, United States

Published: 26 February 2025

HOTMOBILE '25: The 26th International Workshop on Mobile Computing Systems and Applications
February 26 - 27, 2025
CA, La Quinta, USA

Conference Sponsors:
SIGMOBILE

Can we make FCC Experts out of LLMs?

Atul Bansal, Veronica Muriga, Jason Li, Lucy Duan and Swarun Kumar
Carnegie Mellon University
United States of America

ABSTRACT

This paper investigates whether Large Language Models (LLMs) can provide wireless expertise, particularly in the context of spectrum regulations. For any wireless system designer, defining an effective set of wireless parameters, such as the frequency operating band and total radiated power, is one of the very first steps in designing any wireless system. The current approach to do this relies on human FCC experts, who have to consult large volumes of techno-legal documentation on government regulations, such as the FCC rulebook and other industry standards. To ensure the exhaustive coverage of various use cases of wireless systems, these documentations tend to be quite long and very hard to parse through, even for an FCC expert. Given the success of LLMs in providing expert and accurate responses to varied other domains, it is natural to wonder if they could also assist wireless system designers in ensuring their compliance to these techno-legal rules and regulations. In this paper, we present WiLL, a system built on a state-of-the-art LLM, designed to be an assistive tool for wireless system designers to ensure their wireless system complies with regulations. Specifically, we focus on ensuring compliance with FCC regulations for various wireless systems that are designed for technologies such as WiFi, Bluetooth, LoRaWAN and Ultra Wide Band (UWB). We observe that WiLL achieves an accuracy of 78.57%, as compared to an average 51.77% accuracy of off-the-shelf state-of-the-art Large Language Models.

CCS CONCEPTS

• **Networks** → **Wireless access points, base stations and infrastructure**; **Network manageability**; *Mobile networks*; • **Computing methodologies** → **Natural language processing**.

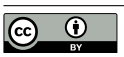
1 INTRODUCTION

In this paper, we explore if the ever-evolving capabilities of Large Language Models (LLMs) can assist wireless system designers, with a special focus on FCC spectrum regulations. Designing a complete wireless system requires expertise on various wireless communication parameters such as operating frequency, bandwidth, transmitted power, bit error rate, overall network performance etc. Among these design decisions, the first step any designer takes is to ensure that the wireless system is compliant with the spectrum regulations to ensure fair and efficient access to limited spectrum resources across all relevant stakeholders. To achieve this, the designer must pore through extensive documentation on spectrum regulations (for example FCC regulations in the United States) to decide system parameters. Effectively implementing and understanding this documentation requires significant technical and legal expertise. Spectrum expertise is therefore inaccessible for many wireless system designers, especially radio enthusiasts and small businesses.

Recent advancements in natural language processing have led to the development of Large Language Models (LLMs) such as ChatGPT[1], which have greatly improved the creation of natural language interfaces for data interaction. Systems built on these models have demonstrated state-of-the-art performance on standard text-to-text datasets. Further, these models have also been extended to work for domain-specific problems such as software code review[20], responding to database queries[12], and biomedical research [24]. Indeed, there has also been prior research on LLMs in the wireless context [21, 22]. However, these works remain focused on either providing a detailed explanation of various wireless system parameters or discussing protocol specifications of specific wireless technologies. While these models act as a good resource for understanding wireless systems, they are not intended to support network design. Hence, a natural question to ask is: ***Can we exploit the power of LLMs to assist in designing a wireless system, specifically, to comply with spectrum regulation?***

We present WiLL, a system that exploits the power of existing Large Language Models (LLMs) to assist amateur wireless enthusiasts in designing a wireless system. Specifically, we focus on the first step that every wireless system designer needs to undertake – ensuring that their network design complies with FCC regulations. We observe that WiLL provides accurate responses to a wide variety of transmitter design questions across various wireless technologies such as WiFi, Bluetooth, Ultra Wide Band, LoRa etc, achieving an accuracy of 78.57% as compared to the average accuracy of 51.77% by the state-of-the-art Large Language Models such as ChatGPT, LLaMa and GPT-4.

We first motivate the need for WiLL by evaluating how existing state-of-the-art Large Language Models perform on a dataset of wireless system design queries and corresponding responses. This dataset is developed by an experienced wireless systems researcher by generating a wide range of queries and answering them after carefully consulting the FCC regulations [5]. The dataset includes various design questions across wireless technologies such as WiFi, Bluetooth, Ultra Wide Band and LoRa. We observe that state-of-the-art LLMs do not perform well in answering these queries. The main reason for this is that wireless system design involves highly specialized queries. Indeed, LLMs are known to be weak to respond to highly specialized queries in niche domains [14] and FCC regulations are no exception. In the U.S, FCC spectrum regulations are described using specialized terminology with various exceptions and use cases based on applications, frequency of operation, etc. A generic language model such as ChatGPT struggles to understand the relationships between these terminologies and misses relevant textual patterns that are required to interpret the wireless transmitter specifications correctly. Further, FCC regulations are typically based on emission power limits that are highly dependent



HOTMOBILE '25, February 26-27, 2025, La Quinta, CA, USA

ACM ISBN 979-8-4007-1403-0/25/02.
https://doi.org/10.1145/3708468.3711885

on wireless transmission parameters such as bandwidth and antenna characteristics, and require numerical computations to verify if the transmission follows FCC regulations. While ChatGPT excels at logical reasoning, generative models are largely not adept at performing numerical calculations, which results in mistakes.

To tackle this challenge, WiLL exploit various technological advances in **context-based learning** or **Retrieval Augmented Generation (RAG)**. This is a technique that involves creating a domain-specific database of embeddings with each embedding representing various chunks of text in the database, followed by correlating user query embeddings with the database to retrieve chunks of text that are most relevant to the user query. The user query is then augmented with these contextual chunks to create a new input prompt. However, naïvely performing context learning does not work well. The main reason for this lies in primitive chunking strategies for the domain-specific embeddings that fail to capture the inherent structure of the regulation document, resulting in individual chunks that consist of a mixture of different (and often irrelevant) information. Performing correlation of the user query embedding with such an embedding database results in contextual chunks, that may unfavorably impact the overall performance.

To address this problem, we take insight from how a human wireless expert parses FCC regulations – in a hierarchical manner. Just as humans use an *index* to effectively and quickly parse through any book or a document, we develop a novel chunking strategy that creates individual chunks using the hierarchical structure of the index of FCC spectrum regulations. Emulating human-like parsing techniques to generate accurate context enables WiLL to effectively exploit the context-based learning capabilities of LLMs to generate accurate answers.

Our evaluation (Sec. 4) shows that WiLL outperforms state-of-the-art LLMs, demonstrating an accuracy of 78.57% compared to an average 51.77% accuracy achieved by state-of-the-art models. The code of WiLL and question-answer dataset used to evaluate WiLL is publicly available here - https://github.com/Jasonic121/WiLL.

2 RELATED WORK

Applications of LLMs: Large Language Models (LLMs) have recently gained immense popularity due to their performance in various text-to-text applications such as web search, software code generation, text-to-image generators, etc. [2, 3]. Foundation models such as GPT-4 [32] can even process and generate outputs across text, audio, and image modalities in real-time.

Engineering LLMs: Even though such models are developed as an all-purpose tool to answer varied queries, they fail with queries that are highly specialized and context-dependent. To address this, researchers have developed techniques such as few-shot learning [26], finetuning [6], and prompt engineering [4], that allow these large pre-trained models to respond well to contextual queries. Prior work has used these techniques in different domains such as medicine, healthcare, and biomedical research [18], SQL queries [28], software code review [13] etc. There also exist techniques that represent the contextual database into graphs (GraphRAG)[9] to enhance the performance of RAGs. However, generating such a representation is complex because of the presence of graphs. All the modules in the RAG pipeline need to be re-designed in order to support Graph inputs or outputs. In contrast, WiLL performs a simple but effective chunking operation to represent contextual database into a special type of graph ($\sim$ trees) and uses state-of-the-art text embedding models to perform contextual embedding.

LLMs in Networking: LLMs have also aided computer networking. Indeed, for the past few years, networking researchers have explored how LLMs can assist in solving real-world problems across layers of the network stack. These include network management by handling network topology and communication graphs [16], incident management in networks [8], network device configuration [25, 27], network security protocol fuzzing[17] etc. Some early work has also emerged in employing LLMs for studying specific wireless protocols [22, 31]. These LLM models are fine-tuned on contexts such as 3GPP protocols [10–12] to create natural language interfaces. While there has been work done in developing LLM's for studying specific wireless protocols, the secret sauce of WiLL lies in the novel hierarchical chunking method that emulates a human in parsing large documents. This hierarchical chunking technique is complimentary with the various fine tuning approaches already explored in the literature and both of these techniques can be used together to improve the performance. Apart from checking compliance for FCC regulations, WiLL can also be generalized across various other networking protocols as discussed in Sec. 6.

In this paper, we discuss how WiLL can assist wireless network designers in crafting a wireless system with a special focus on ensuring compliance with FCC regulations. We also discuss how WiLL can be extended to other aspects of wireless system design as well.

3 STATE-OF-THE-ART LLM PERFORMANCE

In this section, we discuss the performance of state-of-the-art LLMs perform as an assistive tool for wireless system designers, with a specific focus on FCC regulations. To motivate the need for a better approach for using off-the-shelf LLMs in this scenario, we study the performance of state-of-the-art LLMs, such as ChatGPT, GPT-4 and LLaMa, when they are presented with queries regarding FCC regulations.

For this study, we create a set of sixty question/answer pairs with the help of an experienced researcher in wireless system design. Most of these questions describe various properties of a wireless transmitter such as maximum transmit power, bandwidth, antenna gain, operating frequency etc. Depending on the FCC regulations of these properties, the transmitters described in the prompts would either be in compliance or violation of the FCC regulations. We expect these LLMs to output the correct inference (if they comply or not) along with the proper explanation of their answers. We input these questions in various LLMs such as ChatGPT, GPT4 and LLaMA, and verify the correctness of LLM outputs by comparing them with answers provided by a human expert, in our case an experienced wireless researcher. .

To provide a representative example for qualitative understanding, Fig. 1 shows an expert-generated question and ChatGPT's response. The query checks whether the wireless transmitter with the given specifications follows FCC regulations for Ultra Wide Band (UWB) transmissions. We observe two classes of error in the generated response:

Models	Accuracy
ChatGPT	46.42%
GPT-4	73.20%
LLaMa	35.71%

Table 1: Performance of state-of-the-art LLMs on the wireless system design queries

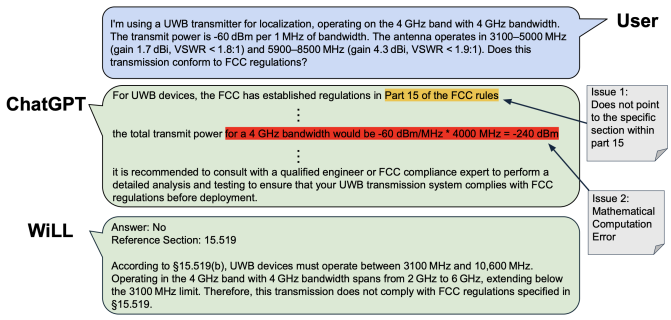


Figure 1: An expert-generated question and ChatGPT's response

- **Incomplete information:** FCC regulations governing UWB transmissions are mentioned in *Section 15* of the entire rulebook. This is correctly mentioned in the first sentence of the generated answer by ChatGPT. However, it does not provide the correct subsection in which the answer is present, thus the output generated by the off-the-shelf ChatGPT is incomplete.
- **Mathematical Computation Errors:** As highlighted in Issue 2 on Figure 1, ChatGPT is prone to making errors in mathematical calculations. The power limits are given in "dBm/MHz" and calculating total power using this method should multiply power by the bandwidth in the logarithmic domain. However, the model multiplies the 4 GHz bandwidth with the power spectral density in an inaccurate manner, generating a wrong answer. This example shows that LLMs often fail at mathematical computations.

We observe that current state-of-the-art LLMs perform very poorly in answering queries related to FCC regulations. The main reason for this poor performance is the presence of specialized information in such queries. This is a well-known problem that language models struggle with [18]. A key reason for the poor performance of LLMs in such specialized and domain-specific tasks is the algorithm that these LLMs use to generate new text. Given a 'context' –that is, a fixed number of preceding tokens– state-of-the-art LLMs maximize token probabilities for each next possible word/token. These token probabilities are learned by training the LLM over a vast corpus of data gathered from various sources such as books, news articles, and scientific journals. When queries are provided from a specialized domain, the generated context probabilities are biased due to the presence of irrelevant information from other domains in the vast corpus of data, leading to prediction inaccuracies.

We observe similar behavior across all the question-answer pairs from our expert-generated dataset. Table 1 is a quantitative summary of how multiple state-of-the-art LLM models perform with the dataset. We observe that all the models perform very poorly with accuracy ranging from 73.20% accuracy in GPT-4 to 35.71% accuracy in LlaMa.

4 WILL SYSTEM DESIGN

In this section, we explore various techniques that WiLL uses to adapt LLMs to the specific domain of FCC regulations, particularly for question answering tasks.

4.1 Context-Based Learning

A key reason why LLMs perform poorly when presented with domain-specific tasks correctly is because they lack correct context. Expert-generated questions have a very specialized context of wireless networks and off-the-shelf LLMs need this contextual information to perform well on such queries. Context-based learning or Retrieval Augmented Generation [19] has emerged as a useful tool in improving the performance of LLMs in specialized domains. In such systems, the user query is appended with appropriate "context" that biases the model towards looking for answers in the specialized domain rather than providing general answers, thus improving performance. However, this requires a grand corpus of data that encapsulates all the specialized information possible. For WiLL this corpus is the FCC regulations[5].

The flowchart shown in Figure 2 demonstrates a typical RAG pipeline. First, the data corpus is embedded into a mathematical vector database in the offline phase. This is done by breaking down the corpus into chunks of text and providing each chunk as input to an embedding model. This embedding model outputs a mathematical vector corresponding to each chunk, which when combined over all chunks of the corpus, creates a vector database. In our system, for a given user query, rather than inputting the query into an LLM, we modify the query by augmenting it with some specialized information known as "context". The context is generated with the help of the corpus database. The user query is transformed into a mathematical vector using the same embedding model that was used in the offline phase, and a similarity approximation algorithm (called a Retriever) is used to extract vectors from the database that have the most similarity to the user query vector. These vectors now serve as a context for the user query. We append this context to the original user query to generate an augmented query, which is the new input to the off-the-shelf LLM. The addition of specialized context to the user query significantly improves the ability of the LLM in generating correct and comprehensive responses.

We observe that even after appending specialized context to the user query, the LLM's response still has errors. Adding context helps to resolve mathematical computation errors, but the LLM still provides incomplete "*Section*" information. After further analysis, we discovered that the performance of the RAG approach greatly depends on the accuracy of the context being generated, since the model prioritizes the context information to generate its own response. If there are errors in the context, the LLM model can perform even worse than the non-context learning based approaches. We observed that WiLL was not generating accurate context information due to inefficient chunking of the data corpus, which resulted in errors. To mitigate this, we explore improvements in the chunking technique, detailed in Sec. 4.2.

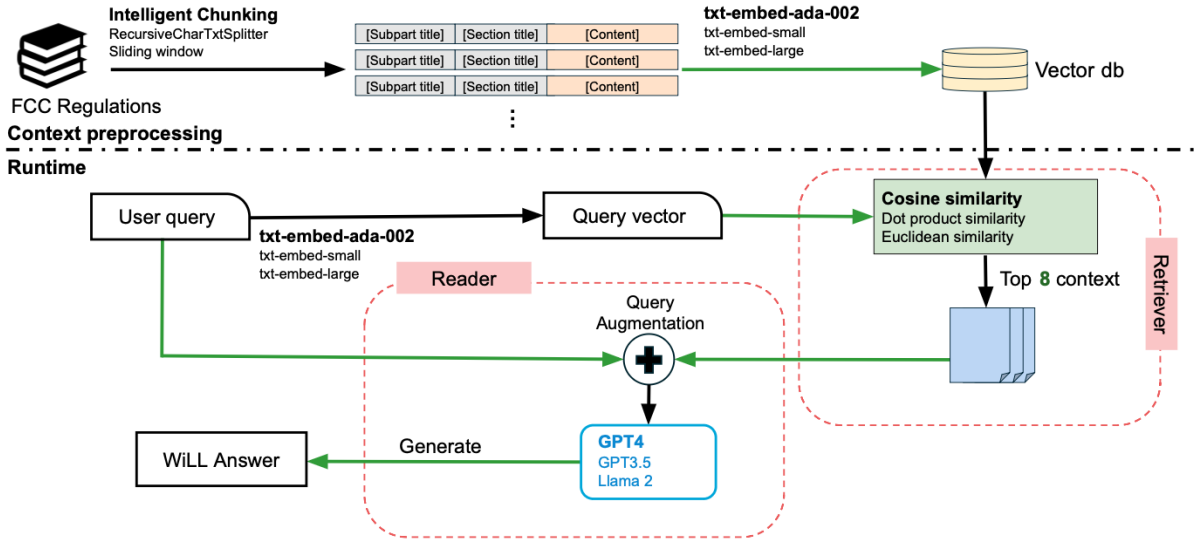


Figure 2: Overview of the context learning process

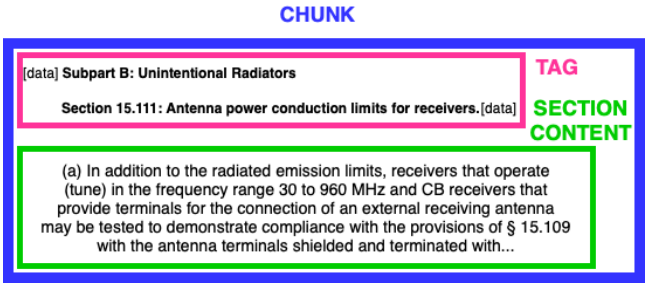


Figure 3: Illustration of hierarchical chunking.

4.2 Intelligent Chunking

The FCC regulation handbook is highly hierarchical, with multiple sections for various classes of applications. These sections are further divided into subsections, with each subsection talking about different frequency bands, specific application cases etc. This has been done to enable a wireless expert to parse through these regulations in an effective manner – similar to humans using an *Index* of a document to quickly parse through and extract the required information.

Using fixed-size chunking methods to chunk the FCC regulations document which is structured in a hierarchical manner can result in overlapping or missing contexts between different sections. This leads to inaccurate inference on the part of the reader model, since the context it receives will include relevant sections incorrectly bundled together with irrelevant sections in a single chunk, decreasing the model's semantic comprehension accuracy. To address this problem, WiLL develops a novel chunking strategy that emulates human-like parsing techniques in our pipeline to generate chunks that maintain the hierarchical structure of the document, as illustrated in Fig. 3. Specifically, we consider section and subsection partitions while creating chunks, by designing the chunking algorithm to be sensitive to textual indicators of partitions, such as new lines and section labels. This simple consideration in the very first step of the pipeline, where we partition the knowledge base, has dramatic consequences on downstream performance. Fig. 1 shows an example of WiLL's performance with intelligent chunking.

5 EVALUATION

In this section, we evaluate WiLL's performance on the question-answer dataset compiled for testing. This dataset contains a set of sixty question/answer pairs with each question describing various properties of a wireless transmitter such as maximum transmit power, bandwidth, antenna gain, operating frequency etc. and the corresponding answer mentions if the transmitter comply with FCC regulations. This set of question/answer pairs was generated by a human expert in wireless system design who manually went through FCC regulations to infer if the described wireless transmitter adheres to it. The questions spanned over four major wireless technologies - WiFi, Bluetooth, UWB and LoRaWAN. The dataset with the corresponding question/answer pairs is publicly available in this Github repo - https://github.com/Jasonic121/WiLL. Furthermore, we also analyze how WiLL's performance responds to changing various parameters in the pipeline in this section.

5.1 WiLL's Accuracy

We evaluate WiLL against other state-of-the-art reader models such as GPT-4, ChatGPT, LLaMa etc. We develop WiLL using the Intelligent chunking method to perform chunking, `text-embedding-ada-002` as our embedding model and cosine similarity as our similarity metric to extract the top 8 most similar chunks as our context to the GPT4 reader model. We observe that WiLL provides an accuracy of 78.57% as compared to 46.42% accuracy of ChatGPT, 35.71% accuracy of LLaMa and 73.2% accuracy of GPT-4, as shown in Table 1. Finetuned GPT-4 performed even worse than off-the-shelf GPT-4 with only 51.79% accuracy. This is because we developed WiLL with only a small dataset, while finetuning requires a large dataset for effective training, and compiling that size of dataset would require manual and time-consuming labor.

Technique	Parameter Tested	Accuracy
Chunking	**Intelligent/hierarchy-based**	**78.57%**
	RecursiveCharacterTextSplitter	62.50%
	SlidingWindow	62.50%

Table 2: Performance of the reader model with intelligent chunking vs. naive chunking techniques.

Technique	Parameter Tested	Accuracy
Similarity function	**Cosine**	**78.57%**
	Dot Product	76.79%
	Euclidean	71.43%
Embedding model	**text-embedding-ada-002**	**78.57%**
	text-embedding-large	73.21%
	text-embedding-small	67.86%
Reader Model	**GPT-4**	**78.57%**
	LlaMa	71.43%
	GPT-3.5	64.29%
	Finetuned GPT-4	51.79%

Table 3: Results of varying the similarity function and embedding techniques with intelligent chunking

5.2 WiLL's Intelligent Chunking

To demonstrate the advantages of hierarchy-based intelligent chunking over other chunking methods, we develop variations of WiLL with different chunking. RecursiveCharacter and SlidingWindow are two examples of naive chunking algorithms. RecursiveCharacter works by detecting markdown separators and splitting the document according to these delimiters. Where no markdown delimiters are observed, a maximum chunk size typically of 1024 characters is defined, after which a new chunk is automatically set. SlidingWindow works by defining fixed chunk sizes, typically of 1000 characters, and defining a stride of 250 characters. SlidingWindow is even more naive than RecursiveCharacter, since this method does not take into account markdown delimiters. Table 2 shows how WiLL outperforms both of the other chunking methods by a 16% margin.

5.3 Embedding

We also evaluate how different embedding models affect the performance of WiLL. We evaluate three models, `text-embedding-ada-002`, `text-embedding-small` , and `text-embedding-large`, and present the results in Table 3. Though `text-embedding-small` is one of the newer and more efficient embedding models offered by the OpenAI platform, we find that it is outperformed by the other models we test against. The `text-embedding-large` model shows a wider distribution of magnitudes, with more vectors having larger magnitudes. The `text-embedding-ada-002` model displays a narrower, more concentrated distribution of magnitudes, indicating that the relative directions of the vectors are more important than their magnitudes for capturing semantic relationships [7]. Therefore we find that `text-embedding-ada-002` is the ideal embedding model for our use case.

k-value	5	6	7	**8**	9	10	11	12
Accuracy(%)	62.50	69.64	69.64	**78.57**	73.21	69.64	75.00	73.21

Table 4: Accuracy of the reader model with different k values.

5.4 Similarity function

We evaluate the ability of different similarity approximation algorithms in comparing the embedding of the user query against embeddings of the chunks of the FCC knowledge base in Table 3. Specifically, we test cosine, euclidian and dot product algorithms, and find that cosine similarity works best for our use case. Given that dot product similarity quantifies differences in both angle and magnitude of vectors, while cosine similarity only considers the angle between vectors, this implies that angle magnitude is a more significant parameter in comparing text embeddings for information retrieval [7].

5.5 Amount of Context

WiLL generates its context by extracting the top 8 closest vectors to the user query from the vector database. We evaluate how the number of extracted vectors (k in our case) affect the performance of WiLL in Table 4. We observe that as we increase the value of k, the accuracy of WiLL increases with accuracy starting from 62.50% at $k = 5$, hitting a maximum of 78.57% at $k = 8$, and then reducing to 73.21% at $k = 12$. This experiment demonstrates a tradeoff between the accuracy of the resulting model and the amount of context we append to the query. If the value of k is set to a large value, the model is presented with a large number of contextual vectors, some of which may not be relevant to the user query, explaining the lower accuracy observed for higher k values. In contrast, if the k value is too small, the embedding model may miss out on the crucial context needed to infer a correct answer. It is therefore important to select the value of k carefully to ensure optimal context-based learning.

6 DISCUSSION AND FUTURE WORK

In this section, we will discuss how WiLL can be generalized to other technologies, using multimodal inputs for WiLL and our dataset limitations.

6.1 Generalizing WiLL

Extension to other technologies. We acknowledge that currently, WiLL only supports queries about the compliance of wireless technologies such as WiFi, Bluetooth, UWB, LoRa etc. with FCC regulations. However, FCC regulations encapsulate the complicance of a large number of other wireless technologies and applications. These include cellular, NFC, RFID, DVB-T technologies along with various satellite frequency bands, military bands, amateur radio bands, emergency communication bands etc. WiLL can be extended to work with these technologies by including the corresponding FCC regulations in the corpus using our hierarchical chunking method. Furthermore, apart from FCC regulations which are the general set of regulations every technology has to comply, there are technology-specific documents that also need to be adhered to while developing a wireless system. These include 3GPP protocols

for cellular technologies, 802.11 protocols for WLAN technologies, SIG protocols for Bluetooth technologies etc. These documents also have a hierarchical structure similar to FCC documents, thus enabling WiLL to possess knowledge about a wide variety of networking protocols.

Incorporating multimodal input. Currently, WiLL only supports text-based queries on FCC compliance. However, various properties of wireless transmitters such as frequency response, return loss ratio, clock oscillator properties are represented in the form of a plot or a graph dependent on other transmitter properties such as frequency, temperature, time etc. Thus, a text-only based FCC compliance system would be very limited in its capability. We envision that advances in multimodal LLM's could be applied to expand WiLL's ability to learn from pictures, graphs and even PDF files of hardware datasheets. There has already been much work done in the field of multimodal LLM's such as in GPT-4v that leverages modality encoders to process photo or video input into a compact representation, and a modality interface to align different data representations [29, 30]. Implementing these expansions in WiLL's pipeline would make it possible for pictures, graphs and PDF files to be included in the intelligent chunking process illustrated in Figure 2. We leave this to future work.

6.2 Dataset-Related Challenges

The performance of WiLL is limited by the size of the dataset used. Given that the knowledge base for WiLL is highly specialized, a subject matter expert is required to manually compile question-answer pairs to generate the dataset. This manual effort is time-consuming and limits the number of generated question-answer pairs to only sixty. Due to this, we could not exploit various optimization techniques used in LLM domains such as fine-tuning to achieve better accuracy. These optimization techniques require a sizably large and diverse dataset for any meaningful improvement in accuracy to be observed. A dedicated large database with question-answer queries comprising a wide variety of scenarios, technologies and applications would significantly improve WiLL. A possible way to increase the size of the data set would be to leverage synthetic data to automate the generation of these question-answer pairs, as shown in [15]. There also exists some previous work that generates synthetic databases using LLMs[23].

7 CONCLUSION

To conclude, we present WiLL, a system designed on top of a context-based learning approach for Large Language Models. We develop a novel chunking strategy to ensure accurate context is appended to the user query. We achieve an accuracy of 78.57% in answering the specialized user queries compared to an average 51.77% accuracy achieved by the state-of-the-art LLM models. In the future, we will explore expanding on WiLL's dataset of question-answer pairs to improve system accuracy. We will also study generalizing WiLL to other radio technologies as well as technology/standard-specific policies beyond spectrum policy.

Acknowledgments: We thank NSF (2114733, 2030154, 2106921, 2007786), the ONR and Cylab Enterprise for supporting this project.

REFERENCES

[1] 2024. Introducing ChatGPT. https://openai.com/index/chatgpt/.
[2] Josh Achiam et al. 2023. Gpt-4 technical report. *arXiv:2303.08774* (2023).
[3] Yupeng Chang et al. 2023. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology* (2023).
[4] Banghao Chen et al. 2023. Unleashing the potential of prompt engineering in large language models: a comprehensive review. *arXiv:2310.14735* (2023).
[5] Federal Communications Commission. 2024. Rules & Regulations for Title 47 — fcc.gov. https://www.fcc.gov/wireless/bureau-divisions/technologies-systems-and-innovation-division/rules-regulations-title-47. [Accessed 03-09-2024].
[6] Ning Ding et al. 2023. Parameter-efficient fine-tuning of large-scale pre-trained language models. *Nature Machine Intelligence* 5, 3 (2023), 220–235.
[7] D Gunawan et al. 2018. The Implementation of Cosine Similarity to Calculate Text Relevance between Two Documents. *Journal of Physics: Conference Series* 978, 1 (mar 2018), 012120. https://doi.org/10.1088/1742-6596/978/1/012120
[8] Pouya Hamadanian et al. 2023. A Holistic View of AI-driven Network Incident Management. In *Proceedings of the 22nd ACM HotNets Workshop*. 180–188.
[9] Haoyu Han, Yu Wang, Harry Shomer, Kai Guo, Jiayuan Ding, Yongjia Lei, Mahantesh Halappanavar, Ryan A Rossi, Subhabrata Mukherjee, Xianfeng Tang, et al. 2024. Retrieval-Augmented Generation with Graphs (GraphRAG). *arXiv preprint arXiv:2501.00309* (2024).
[10] Khen Bo Kan, Hyunsu Mun, Guohong Cao, and Youngseok Lee. 2024. Mobile-llama: Instruction fine-tuning open-source llm for network analysis in 5g networks. *IEEE Network* (2024).
[11] Athanasios Karapantelakis et al. 2024. Using large language models to understand telecom standards. In *ICMLCN*. IEEE, 440–446.
[12] Manikanta Kotaru. 2023. Adapting Foundation Models for Operator Data Analytics. In *HotNets*. 172–179.
[13] Feng Lin et al. 2024. When llm-based code generation meets the software development process. *arXiv preprint arXiv:2403.15852* (2024).
[14] Chen Ling et al. 2023. Domain specialization as the key to make large language models disruptive: A comprehensive survey. *arXiv:2305.18703* (2023).
[15] Yingzhou Lu, Minjie Shen, Huazheng Wang, Xiao Wang, Capucine van Rechem, Tianfan Fu, and Wenqi Wei. 2024. Machine Learning for Synthetic Data Generation: A Review. arXiv:2302.04062 [cs.LG] https://arxiv.org/abs/2302.04062
[16] Sathiya Kumaran Mani et al. 2023. Enhancing network management using code generated by large language models. In *HotNets*. 196–204.
[17] Ruijie Meng, Martin Mirchev, Marcel Böhme, and Abhik Roychoudhury. 2024. Large language model guided protocol fuzzing. In *Proceedings of the 31st Annual Network and Distributed System Security Symposium (NDSS)*.
[18] Soumen Pal et al. 2024. A domain-specific next-generation large language model (LLM) or ChatGPT is required for biomedical engineering and research. *Annals of Biomedical Engineering* 52, 3 (2024), 451–454.
[19] Dong Qingxiu et al. 2023. A Survey for In-context Learning. *ArXiv* abs/2301.00234 (2023). https://api.semanticscholar.org/CorpusID:263886074
[20] Zeeshan Rasheed et al. 2024. AI-powered Code Review with LLMs: Early Results. *arXiv preprint arXiv:2404.18496* (2024).
[21] Nurullah Sevim et al. 2024. Large Language Models (LLMs) Assisted Wireless Network Deployment in Urban Settings. *arXiv preprint arXiv:2405.13356* (2024).
[22] Jiawei Shao et al. 2024. WirelessLLM: Empowering Large Language Models Towards Wireless Intelligence. *arXiv:2405.17053* (2024).
[23] Chunliang Tao, Xiaojing Fan, and Yahe Yang. 2024. Harnessing llms for api interactions: A framework for classification and synthetic data generation. *arXiv preprint arXiv:2409.11703* (2024).
[24] Surendrabikram Thapa et al. 2023. ChatGPT, bard, and large language models for biomedical research: opportunities and pitfalls. *Annals of biomedical engineering* 51, 12 (2023), 2647–2651.
[25] Changjie Wang, Mariano Scazzariello, Alireza Farshin, Simone Ferlin, Dejan Kostić, and Marco Chiesa. 2024. NetConfEval: Can LLMs Facilitate Network Configuration? *Proceedings of the ACM on Networking* 2, CoNEXT2 (2024), 1–25.
[26] Yaqing Wang et al. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM computing surveys (csur)* 53, 3 (2020), 1–34.
[27] Duo Wu, Xianda Wang, Yaqi Qiao, Zhi Wang, Junchen Jiang, Shuguang Cui, and Fangxin Wang. 2024. Netllm: Adapting large language models for networking. In *Proceedings of the ACM SIGCOMM 2024 Conference*. 661–678.
[28] Xiaojun Xu et al. 2017. Sqlnet: Generating structured queries from natural language without reinforcement learning. *arXiv preprint arXiv:1711.04436* (2017).
[29] Shukang Yin et al. 2024. A survey on multimodal large language models. *National Science Review* 11, 12 (11 2024), nwae403.
[30] Duzhen Zhang et al. 2024. MM-LLMs: Recent Advances in MultiModal Large Language Models. In *ACL 2024*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). ACL, Bangkok, Thailand. https://aclanthology.org/2024.findings-acl.738/
[31] Hao Zhou et al. 2024. Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities. *arXiv preprint arXiv:2405.10825* (2024).
[32] Ning Zhu et al. 2024. OpenAI's GPT-4o in surgical oncology: revolutionary advances in generative artificial intelligence. *European Journal of Cancer* (2024).