Telco-RAG: Navigating the Challenges of Retrieval Augmented Language Models for Telecommunications

Andrei-Laurentiu Bornea*, Fadhel Ayed*, Antonio De Domenico*, Nicola Piovesan*, Ali Maatouk+

*Paris Research Center, Huawei Technologies, Boulogne-Billancourt, France

+Yale University, New Haven, Connecticut, USA

Abstract—The application of Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) systems in the telecommunication domain presents unique challenges, primarily due to the complex nature of telecom standard documents and the rapid evolution of the field. The paper introduces Telco-RAG,[1] an open-source RAG framework designed to handle the specific needs of telecommunications standards, particularly 3rd Generation Partnership Project (3GPP) documents. Telco-RAG addresses the critical challenges of implementing a RAG pipeline on highly technical content, paving the way for applying LLMs in telecommunications and offering guidelines for RAG implementation in other technical domains.

I. INTRODUCTION

Large language models (LLMs) are designed to understand, generate, and process text by leveraging extensive training data. These models, built upon architectures such as Transformers, employ deep learning techniques to analyze and predict language patterns [1]. Their capabilities are largely attributed to the vast amount of text they process during training, allowing LLMs to develop a nuanced understanding of language, context, and even idiomatic expressions. The utility of LLMs extends across various domains, among which telecommunications, where models can improve operational efficiency and enhance customer satisfaction [2].

Standalone language models rely solely on their internal representations and learned parameters to generate text, which showcases modest knowledge in technical domains such as telecommunication standard documents [3]. Two primary methodologies have emerged to address this challenge: fine-tuning and retrieval-augmented generation (RAG). Fine-tuning enables language model specialization via further training of a fraction of the parameters using a domain-specific dataset. However, fine-tuning can incur a high computational cost [4] and is not suited for rapidly evolving domains where new knowledge needs to be incorporated on a regular basis. RAG stands out as an appealing alternative due to its cost-effectiveness, adaptability, and scalability [5]. In the RAG paradigm, knowledge from external sources is fetched in real-time when a query is addressed to the system. This is particularly tailored for quickly evolving fields [6].

In the telecommunication industry, a retrieval-augmented language model that masters complex industry-specific knowledge, such as the content of technical standards, would hold significant practical value [7]. For instance, it would allow the development of an advanced chatbot for professionals.

[1] https://github.com/netop-team/telco-rag

Such a tool would increase the accuracy and speed with which telecommunications professionals access and comply with international standards, fostering quicker development cycles and improved regulatory adherence.

In this work, we concentrated our efforts on telecommunication standards, and specifically 3rd Generation Partnership Project (3GPP) documents. This focus was motivated by the aforementioned practical utility of a chatbot specialized in 3GPP and by the observation that even state-of-the-art language models, such as GPT-4, exhibit scarce knowledge of this content [3].

We have identified that the conventional RAG setup, which typically extracts three to five data segments of 512 tokens each [8], does not adequately meet the intricate demands of telecommunications standards. Consequently, we have developed a specialized RAG pipeline named Telco-RAG specifically optimized for 3GPP documents.

Besides, through our design and methodology, we aim to provide generally applicable guidelines to overcome the common challenges faced when implementing an RAG pipeline in highly technical domains. These include identifying the most impactful hyperparameters to tune, recommending default settings [9], reducing the high random access memory (RAM) usage, and refining the user's query [10]. We expect that the Telco-RAG, which we make publicly available as an open-source chatbot for 3GPP standards, and the associated results will contribute substantially to integrating AI in the telecommunications field.

II. METHODOLOGY

RAGs improve the quality of the LLM-generated responses by providing the LLM with external sources of knowledge based on a large corpus of documents. RAG pipelines start by splitting the document corpora into fixed-sized long (chunk size) segments called chunks. Using an embedding model, each chunk is transformed into a vectorial representation capturing the semantics of the segment. When a query is presented, the system identifies the relevant chunks by computing a similarity between the chunks' embeddings and the query's embedding. Lastly, RAG presents the relevant chunks, called the context, alongside the query to a LLM that generates the final response.

Any implementation of a RAG system for telecommunications will face four critical challenges: sensitivity to hyperparameters [9], vague user queries [10], high RAM requirements, and sensitivity to the quality of the prompts [11]. In particular,
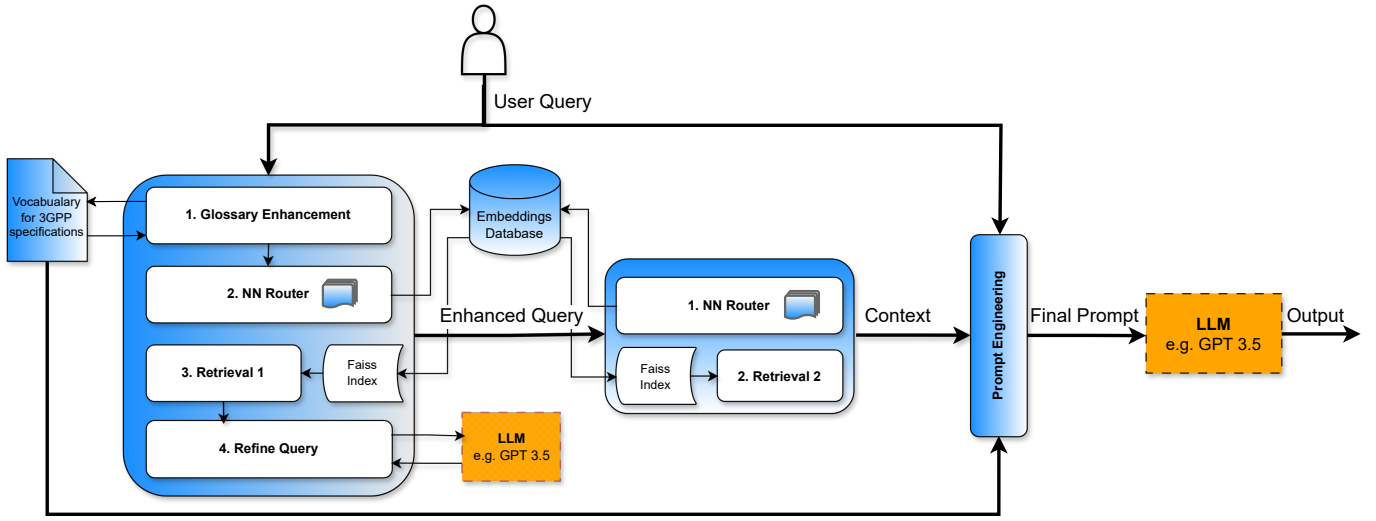
Fig. 1. The proposed Telco-RAG architecture.

poor prompts affect the capacity of the LLMs to comprehend the context of the queries and correctly reply to them. Moreover, vague queries limit the precision of the retrieval stage.

Fig. 1 depicts the proposed Telco-RAG tailored for LLM deployment in the telecommunications sector. The proposed system aims to improve the retrieval and processing of technical documents, focusing specifically on 3GPP standards. It features a dual-stage pipeline, including a query enhancement stage and a retrieval stage. The query enhancement stage includes four steps: initially, it employs a custom glossary of technical terms to augment the query, enhancing contextual understanding. Subsequently, a neural network (NN) router selectively identifies relevant documents from the document corpus. This sub-selection optimizes the accuracy and efficiency, reducing the number of documents loaded in the preliminary retrieval (step 3), which provides the first round of context used to further refine the queries (step 4). Following this, the retrieval stage utilizes the NN router to select the documents (step 1) on which the RAG realizes the second retrieval (step 2). Note that using the improved query boosts the accuracy of the second retrieval thanks to the more accurate embedding representations. The pipeline finalizes with a generating component, relying on a state-of-the-art language model such as GPT-3.5, which generates responses based on retrieved context.

A. Hyperparameters Optimization

As numerous studies have shown, hyperparameter optimization can provide large gains for retrieval-augmented models (see [8], [12]). Therefore, using a synthetic dataset constructed for this purpose, we conducted a meticulous optimization of the chunk size, context length, indexing strategy, and embedding models (see Sec. III). These hyperparameters are explained below

- **Chunk Size:** Determines the length of each text segment the RAG processes at once.

- **Context Length:** Length of the context yielded by retrieval component.
- **Embedding Models:** Algorithms that transform text into numerical representations.
- **Indexing Strategy:** The FAISS index[2] [13] by which the model assesses the relevance of each text chunk related to the given query.

B. Query Augmentation

Numerous studies indicate significant improvements when augmenting vague queries in RAG pipelines [10]. In telecom documents, two major issues arise with vague queries: the abundance of technical terms and abbreviations in questions, and the inability of the RAG to discern user intent, leading to the retrieval of irrelevant, albeit similar, information.

1) Lexicon-enhanced Queries: In this section, we address the challenge posed by the prevalence of technical terms and abbreviations in questions, which are often difficult to capture accurately in the embedding space. To tackle this issue, we utilized the "Vocabulary for 3GPP Specifications" [14] to construct two dictionaries: one for abbreviations and another for terms with their definitions. Integrating these dictionaries into our query enhancement block of the pipeline- see Fig. 1- allowed us to refine the embedding process. For each question, we enriched the embedding with the definitions of relevant terms from the dictionaries. This process refines the similarity evaluation between the question and potential answers by incorporating domain-specific knowledge. We also integrated relevant terms from the dictionaries in our final prompt, built using the retrieved context, user query, and the defined terms and definitions. This ensures that the LLM was prompted with the necessary technical vocabulary and definitions to process

---

[2]FAISS is a library by Facebook AI Research optimized for efficient similarity search and clustering of dense vectors in large datasets on modern CPUs and GPUs.

the question effectively. This method was employed in the Glossary Enhancement block of our pipeline, see Fig. 1.

2) Generating Candidate Answers: We use a language model to generate all plausible answers based on the preliminary context selected in Retrieval 1. Then, we add these generated candidate answers to enhance the user's query, clarifying its intent and preventing the retrieval of irrelevant information. The embedded enhanced query improves the identification of the relevant information in the corpora, yielding a superior final answer quality.

C. Enhancing the RAM Usage of the Telco-RAG

For large document corpora, the dataset of the embedded chunks becomes so voluminous that it exceeds the limitations of RAM capacities. Besides, we show in this work that for highly technical documents, smaller chunks yield better performance (see Sec. III-A2). However, the smaller the chunks, the more the text segments to be processed by the RAG, which increases the required RAM resources. To deal with this issue, we recall that the 3GPP standards categorize specifications into 18 distinct series [15]. Each series provides the technical details of a specific aspect of mobile telecommunications technologies (radio access, core network components, security, etc). To improve the RAG usage efficiency, we developed an NN router tailored to predict relevant 3GPP series based on queries. This model enables selective loading of embeddings, thus drastically reducing the RAM usage.

The architecture of the NN router incorporates two distinct input channels. The first channel processes input 1, a 1024-sized vector embedding the initial user query, while the second channel processes input 2, an 18-sized vector. Each entry of this vector is defined as the inner product between query embeddings and the embedding of each 3GPP series summary description, generated through a dedicated LLM.

Central to our model are two adjustable trainable parameters, α and β, which modulate the influence exerted by each input stream on the resultant output. The overall architecture is illustrated in Fig. 2.

For the processing of the embedded query, our model implements a series of linear transformations that reduce its dimensionality from 1024 to 256. This reduction incorporates dropout layers to mitigate overfitting and a batch normalization layer to enhance training stability. Concurrently, the second input stream begins with a dimensionality of 18, preprocessed through a softmax layer, which is then expanded to 256 dimensions, to process jointly the contributions from both input streams in the decision-making process. The outputs from these pathways are weighted by 2 trainable parameters, α and β. These weighted outputs are summed up into a unified representation, which our neural network model utilizes to ascertain the target 3GPP series with heightened accuracy.

Integrating this NN model into Telco-RAG framework significantly elevates the ability to discern and categorize standards-related queries, paving the way for more targeted and efficient information retrieval.
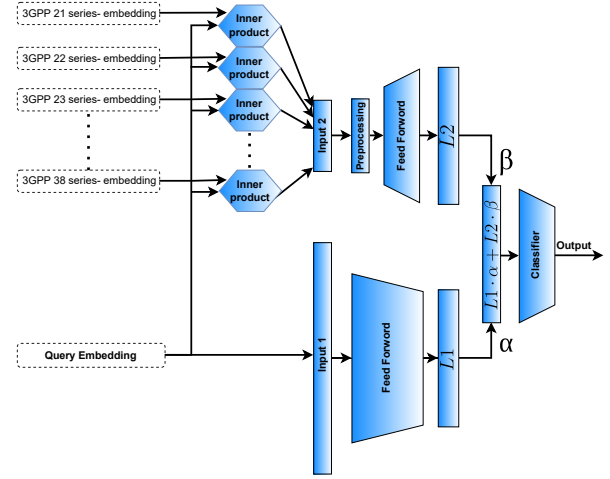

Fig. 2. The proposed NN router architecture.

To train the NN router, we created a synthetic dataset comprising 30,000 questions from 500 documents from 3GPP Release 18, and their originating series that served as target labels. The adoption of synthetic data for training and testing our NN router reduces the risk of overfitting the dataset on which we test Telco-RAG pipeline [16].

D. Prompt Engineering

Prompt engineering plays a crucial role in RAG systems, particularly in ensuring that the RAG maintains focus on the user's question while comprehending the broader context [11].

In our study, we designed a structured, dialogue-oriented prompt, as prompt engineering literature has shown better LLM performance with this format [17]. More specifically, the final prompt of Telco-RAG starts with the query followed by the definitions of the terms and abbreviations. After that, the prompt includes the generated context. Importantly, the proposed prompt includes the query repetition, before the related options and query instruction helping the model to effectively generate a relevant response. The designed format of the LLM prompt is as follows:

---

*Please provide the answers to the following multiple-choice question: <Question>
*Terms and Definitions: <Defined Terms>
*Abbreviations: <Abbreviations>
*Considering the following context: <Retrieved Context>
*Please provide the answers to the following multiple-choice question: <Question>
*Options: <Options>
*Write only the option number corresponding to the correct answer.

III. EXPERIMENTAL RESULTS

In this section, we present the performance of Telco-RAG framework in enhancing the capabilities of LLMs applied to

Name	Embedding Model	Chunk Size	Context Length	Candidate Answers	Glossary Enhancement	NN Router	Enhanced Final Prompt
Benchmark RAG	-3-large[1024 size]	500	1500-2500	IndexFlatIP	No	No	No
Telco-RAG	-3-large[1024 size]	125	1500-2500	IndexFlatIP	Yes	Yes	Yes

the telecom domain. To achieve this task, we have used two sets of multiple choice questions (MCQs), one optimization set and one evaluation set, specifically created to assess the knowledge of LLMs in telecommunications. The optimization set is composed of 2000 MCQs generated following the methodology presented in [3] and based on documents from 3GPP Rel.18. The second set consists of the 1840 TeleQnA MCQs related to 3GPP documentations [3]. The purpose of Telco-RAG is to effectively help professionals with complex queries from telecom domain. The MCQ format, though very convenient for evaluation purposes, does not realistically correspond to the type of queries that will be submitted to the system. i.e., the user will likely do not provide any option to the LLM. Hence, we decided not to include the options in the retrieval process and use them solely to assess Telco-RAG accuracy. In the following results, accuracy measures the fraction of correct answers of Telco-RAG to the queries in the datasets.

Table I presents the main parameters of Telco-RAG and the RAG benchmark architecture compared throughout the following experiments.

A. Hyperparameters Optimization

1) Selecting the Embedding Model: In this experiment, we compare the performance of two OpenAI embedding models for the Telco-RAG framework: 1) Text-embedding-3-large and text-embedding-ada-002 [18]. Text-embedding-3-large extends the capabilities of its predecessor text-embedding-ada-002. Text-embedding-3-large is trained using Matryoshka Representation Learning [19], a technique that allows the shortening of the embedding vectors, which reduces computational and RAM requirements, while preserving a stronger performance. Our results show that, on average, the text-embedding-3-large model, with a fixed embedding dimension of 1024, improves the accuracy of Telco-RAG by 2.29%, over the text-embedding-ada-002 model.

2) Chunk Size Optimization: We have assessed the influence of varying chunk sizes— 125, 250, and 500 tokens—on the accuracy of RAG systems. Importantly, there is an inverse relationship between chunk size and Telco-RAGaccuracy. These results highlight the critical importance of optimizing chunk size, which has led to an average improvement of 2.9% in accuracy when selecting as chunk size 125 tokens instead of 500 tokens, for equal context length.

3) Context Length Optimization: Fig. 3 shows the linear regression fitted on the RAG accuracy computed for a diverse set of context lengths, with different configurations. The results
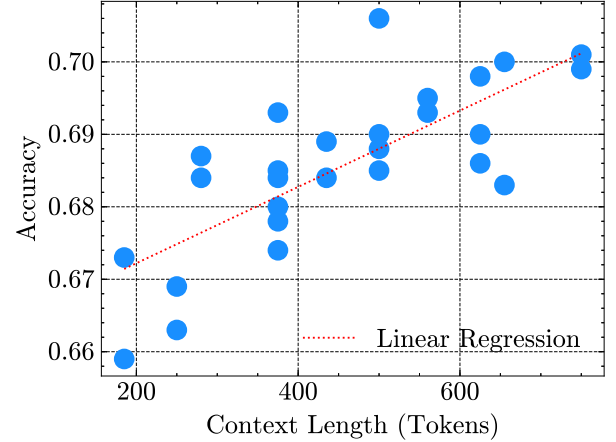

Fig. 3. RAG's accuracy vs context length.

show an ascending trend of the accuracy as a function of context length. As a side note, we have noticed a drop in performance when the context length gets larger than 1500 tokens. However, this is alleviated by presenting the query twice, before and after the context, as discussed in Sec. II-D.

4) Indexing Strategy Selection: In our research, we have evaluated the impact of different indexing strategies in the accuracy of Telco-RAG: 1) IndexFlatL2, 2) IndexFlatIP, and 3) IndexHNSW. IndexFlatL2 is based on the Euclidean distance while IndexFlatIP uses Euclidean dot product. In constrast, IndexHNSW is an approximate method for efficient searching in high-dimensional data spaces using Euclidean distance. IndexHSNW has shown considerably inferior performance compared to IndexFlatIP and IndexFlatL2. Importantly, despite marginal differences in terms of accuracy, IndexFlatIP has outperformed IndexFlatL2 in 80% of our experiments.

B. Query Augmentation

In this section we evaluate the gain in accuracy brought by enhancing the user queries through the methodology described in Sec. II-B.

1) Lexicon-enhanced Queries: To validate the effectiveness of this approach, we applied it to a subset of lexicon-focused questions from TeleQnA [3], which were designed to evaluate the understanding of abbreviations and technical terms within the telecommunications sector. Our results presented in Table II have shown that the designed RAG framework enhances the baseline LLM accuracy on lexicon questions, i.e., from 80.2 % to 84.8%. However, Lexicon-enhanced queries have achieved an accuracy rate exceeding 90% on these questions, gaining

6% compared to the same RAG pipeline without the lexicon enhancement.

Baseline (No Context)	Benchmark RAG	Telco-RAG
80.2%	84.8%	90.8%

2) Enhancing User's Query With Candidate Answers: To retrieve a better context, we enhance the user's query with candidate answers generated by an LLM (step 4 of the query's enhancement stage). Table III presents the accuracy of Telco-RAG with and without the usage of these candidate answers. Specifically, we can observe that for the text-embed-ada-002 embedding model, the addition of candidate answers considerably improves the query embedding representations, bringing a 3.56% average accuracy gain. The accuracy of the RAG with text-embed-ada-002 including refined queries is larger than the one achieved using text-embed-3-large without refined queries. Furthermore, with text-embed-3-large, we observe a gain of 2.06% on average accuracy when using candidate answers in the retrieval process.

Embedding Model	Chunk Size	Context Length	Initial Accuracy	Refined Accuracy
Text-embed-ada-002	125	750	0.729	**0.777 (+4.8%)**
Text-embed-ada-002	250	2000	0.770	**0.795 (+2.5%)**
Text-embed-ada-002	500	2000	0.740	**0.774 (+3.4%)**
Text-embed-3-large	125	750	0.744	**0.780 (+3.6%)**
Text-embed-3-large	250	2000	0.784	**0.796 (+1.2%)**
Text-embed-3-large	500	2000	0.774	**0.788 (+1.4%)**

C. RAM Usage Analysis in the Telco-RAG

Selecting a 125-token chunk size increases the RAM requirements of the Telco-RAG (see Sec. II-C). However, the integration of the designed NN router can tackle this issue. Fig. 5 presents the histogram of RAM usage for the 2000 MCQs in the optimization set. NN router dynamically selects the number of documents processed by the Telco-RAG pipeline based on their relevance to the query, as opposed to a fixed number of documents processed by the Benchmark RAG architecture. This method introduces variability in RAM usage among different queries, which results in the probability density function (PDF) in Fig. 5. Our results show that the NN-enhanced RAG model leads in average to a RAM consumption of 1.25 GB, thus reducing of 45% the requirement of 2.3 GB -obtained by the Benchmark RAG solution.

As presented in Sec. II-C, the proposed NN router reduces the RAM usage of the designed RAG by selecting a limited number of retrieved documents, whose content relates with the query. To assess the capability of the NN router, we have

compared it to GPT 3.5 and GPT 4 and provided to each model the questions of our optimization dataset including with each of their prompt the related 3GPP series description, Therefore, we have asked the three model to indicate the top k most related 3GPP series. Table IV shows the accuracy determined by subsequently verifying if the correct 3GPP series was among the top k retrieved ones.

The results in Table IV highlight that the proposed NN model outperforms both GPT 3.5 and GPT 4 in identifying relevant 3GPP Series for a query, leading to an average accuracy gain of 37.8% and 11.1%, respectively.

Top k	NN Router	GPT 3.5	GPT 4
k=1	**51.3%**	19.9%	30.4%
k=3	**80.6%**	36.6%	70.8%
k=5	**88.3%**	50.3%	85.6%

The ability of the designed NN router to accurately deduce the applicable 3GPP series for a given query reduces the consideration of irrelevant content. This reduction not only lowers the computational complexity of the retrieval steps but also the overall resources needed for processing the retrieved content.

D. Enhanced Prompt Formatting

In this section, we highlight the accuracy gain brought by the prompt presented in Sec. II-D, which we have designed for LLM answering MCQs related to telecom domain. Our analysis of the results revealed a 4.6% average gain in accuracy, compared to the original JSON format of TeleQnA questions This result suggests that human-like query structures can significantly elevate the contextual understanding and accuracy of LLM models.

E. Overall Performance

In this section, we present the accuracy of the Telco-RAG on the evaluation MCQs, i.e., 1840 3GPP-related questions from TeleQnA [3]. Specifically, we consider three groups of MCQs, Rel. 17 MCQs, Rel. 18 MCQs, and the overall set of TeleQnA MCQs related to 3GPP documentations. For each of these sets of MCQs, we compare the performance of GPT 3.5 with Telco-RAG, GPT 3.5 with the Benchmark RAG, and GPT 3.5 without RAG. Fig. 4 highlights that Telco-RAG leads to notable gains in all the experiments. Importantly, Telco-RAG results an average improvement of 6.6% and 14.45% compared to GPT 3.5 with and without the Benchmark RAG.

IV. CONCLUSIONS

This paper presented Telco-RAG, a novel RAG framework for processing 3GPP telecommunications standards and supporting LLM in telecom use cases. We have demonstrated that refinements in chunk sizes, embedding models, indexing strategies, and query structuring significantly boost RAG
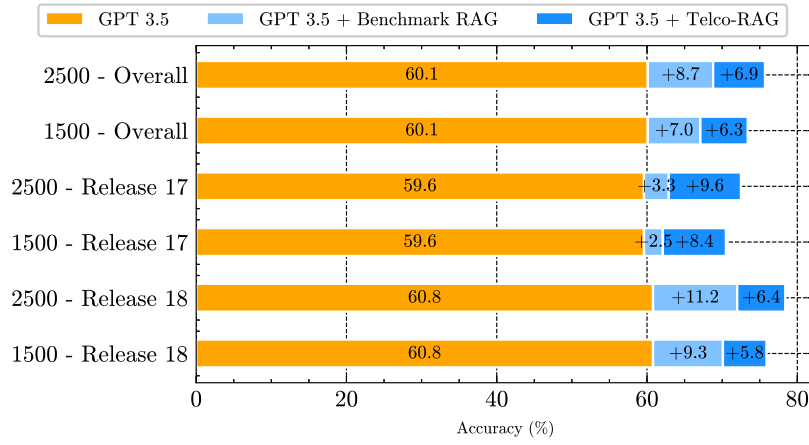
Fig. 4. Comparison the accuracy of Telco-RAG system with a baseline GPT 3.5 with/without the Benchmark RAG on the TeleQnA questions related to 3GPP documents.
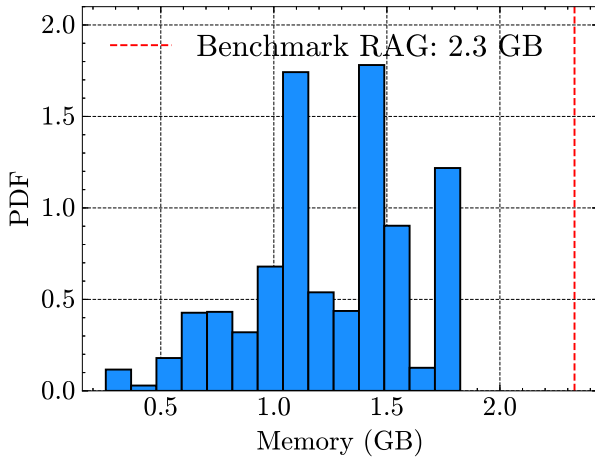

Fig. 5. PDF of the RAM usage of Telco-RAG vs Benchmark RAG.

system performance and accuracy. The provided solutions are general and can deal with frequent challenges encountered in building RAG pipelines for highly technical domains. We expect that the Telco-RAG, which we make publicly available, and the associated results will contribute substantially to the integration of AI in the telecommunications field.

REFERENCES

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems (NIPS)*, 2017, pp. 5998–6008.

[2] A. Maatouk, N. Piovesan, F. Ayed, A. D. Domenico, and M. Debbah, "Large Language Models for Telecom: Forthcoming Impact on the Industry," *IEEE Communications Magazine*, pp. 1–7, 2024.

[3] A. Maatouk, F. Ayed, N. Piovesan, A. De Domenico, M. Debbah, and Z.-Q. Luo, "TeleQnA: A Benchmark Dataset to Assess Large Language Models Telecommunications Knowledge," *arXiv preprint arXiv:2310.15051*, 2023.

[4] N. C. Thompson, K. Greenewald, K. Lee, and G. F. Manso, "The computational limits of deep learning," *arXiv preprint arXiv:2007.05558*, 2020.

[5] O. Ovadia, M. Brief, M. Mishaeli, and O. Elisha, "Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLMs," *arXiv preprint arXiv:2312.05934*, 2024.

[6] A. Balaguer, V. Benara, R. Cunha, R. Estevão, T. Hendry, D. Holstein, J. Marsman, N. Mecklenburg, S. Malvar, L. O. Nunes *et al.*, "RAG vs Fine-tuning: Pipelines, Tradeoffs, and a Case Study on Agriculture," *arXiv preprint arXiv:2401.08406*, 2024.

[7] N. Piovesan, A. De Domenico, and F. Ayed, "Telecom language models: Must they be large?" *IEEE 35th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, 2024.

[8] LlamaIndex, "Evaluating the Ideal Chunk Size for a RAG System Using LlamaIndex," LlamaIndex Blog, 2024. [Online]. Available: https://www.llamaindex.ai/blog/evaluating-the-ideal-chunk-size-for-a-rag-system-using-llamaindex-6207e5d3fec5

[9] P. Finardi, L. Avila, R. Castaldoni, P. Gengo, C. Larcher, M. Piau, P. Costa, and V. Caridá, "The Chronicles of RAG: The Retriever, the Chunk and the Generator," *arXiv preprint arXiv:2401.07883*, 2024.

[10] C.-M. Chan, C. Xu, R. Yuan, H. Luo, W. Xue, Y. Guo, and J. Fu, "RQ-RAG: Learning to Refine Queries for Retrieval Augmented Generation," in *arXiv preprint arXiv:2404.00610*, 2024.

[11] Banghao Chen, Zhaofeng Zhang, Nicolas Langren, Shengxin Zhu, "Unleashing the potential of prompt engineering in Large Language Models: a comprehensive review," *arXiv preprint arXiv:2310.14735*, 2023.

[12] S. Siriwardhana, R. Weerasekera, E. Wen, T. Kaluarachchi, R. Rana, and S. Nanayakkara, "Improving the Domain Adaptation of Retrieval Augmented Generation (RAG) Models for Open Domain Question Answering," *Transactions of the Association for Computational Linguistics*, vol. 11, pp. 1–17, 01 2023.

[13] J. Johnson, M. Douze, and H. Jégou, "Faiss: Facebook ai similarity search," https://github.com/facebookresearch/faiss, 2017.

[14] 3GPP TSG SA, "TR 21.905, Vocabulary for 3GPP Specifications," *V17.2.0*, March 2024.

[15] 3GPP, "Specifications by series." [Online]. Available: https://www.3gpp.org/specifications-technologies/specifications-by-series

[16] F. Gilardi, M. Alizadeh, and M. Kubli, "ChatGPT Outperforms Crowd-Workers for Text-Annotation Tasks," *Proceedings of the National Academy of Sciences*, vol. 120, no. 30, 2023.

[17] J. Gao, M. Galley, and L. Li, "Neural Approaches to Conversational AI," in *Annual Meeting of the Association for Computational Linguistics (ACL): Tutorial Abstracts*, 2019.

[18] OpenAI, "New embedding models and api updates," https://openai.com/blog/new-embedding-models-and-api-updates, 2023, accessed: 2024-04-18.

[19] A. Kusupati, G. Bhatt, A. Rege, M. Wallingford, A. Sinha, V. Ramanujan, W. Howard-Snyder, K. Chen, S. Kakade, P. Jain, and A. Farhadi, "Matryoshka Representation Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.