

Verified TelcoRAG: A Verified Local-Rag for Telecommunication Uses

Yasin Salajeghe*, Ali Majlesi Kopae*, Shaghayegh Najjari[†], Iman Ahmadi*,
Mohammad Amin Dehmolaee[‡], Babak Hossein Khalaj*[†]

**Department of Electrical Engineering, Sharif University of Technology, Tehran, Iran*

Emails: yasin.salajeghe79@sharif.edu, ali.majlesi@sharif.edu, Iman.ahmadi@ee.sharif.edu, khalaj@sharif.edu

*[†]Center for Information Systems & Data Science, Institute for Convergence Science & Technology,
Sharif University of Technology, Tehran, Iran*

Email: shaghayegh.najjari@sharif.edu

[‡]Asr Goyesh Pardaz, Tehran, Iran

Email: amindhm1313@gmail.com

Abstract—Large Language Models (LLMs) typically require vast computational resources for training and inference, making their deployment in resource-constrained environments difficult. This limitation has driven researchers to explore alternative strategies, such as using smaller, more efficient models that can still deliver high-quality results. By incorporating Retrieval-Augmented Generation (RAG), these smaller models can compensate for their limited training data and parameters by dynamically retrieving relevant information from external knowledge sources during inference.

We present Verified TelcoRAG, a specialized framework that adapts RAG for smaller language models, specifically Phi2, to address queries related to 3rd Generation Partnership Project (3GPP) telecommunications documents. Verified TelcoRAG builds one existing SpeculativeRAG method by not only generating responses, but also verifying multiple candidate responses from various retrieval sources to ensure greater accuracy. This verification process allows for more reliable and precise answers, particularly in specialized fields such as telecommunications. In this work, we achieved a 3% improvement in the performance of the SpeculativeRAG model, making Verified TelcoRAG a cutting-edge approach in this research domain.

Index Terms—LLM, RAG, TelcoRAG, Speculative RAG, Telecommunication.

I. INTRODUCTION

This section discusses the integration of RAG systems with LLMs to improve answer relevance and accuracy by retrieving contextually appropriate information during inference. RAG addresses the inherent limitations of standalone LLMs by incorporating external knowledge sources, enhancing response quality.

With the increasing computational demands of LLMs, the development of smaller, more efficient models has gained traction. These compact models aim to perform well on constrained hardware but still encounter knowledge limitations. Combining RAG with smaller LLMs is an effective strategy to overcome these limitations by enriching generation with retrieved data.

RAG systems also enhance local document handling by automating the extraction of meaningful information rather

than returning raw search results. This capability significantly boosts productivity and reduces the manual effort required to access specific content.

Previous works have explored various paradigms for integrating retrieval into the generation process. Retrieved data can serve as augmented input [1], [2], latent information during intermediate stages [3], [4], modifications to final output logits [5], [6], or mechanisms to alter or skip stages of generation [7].

Retrieval methods typically rely on similarity-based matching between queries and contexts. These approaches include vector space models such as TF-IDF and Word2Vec [8], probabilistic models like BM25 [9], neural models such as BERT [10], and rule-based systems. Each method has distinct advantages depending on task complexity and required precision.

Robust reasoning remains a persistent challenge for LLMs, which context length extensions alone cannot fully solve [11]–[13]. Many RAG systems improve contextual quality but introduce latency or require multiple optimization steps [14], [15], limiting their practical deployment.

To address this, SPECULATIVE RAG [16], inspired by speculative decoding [17], enhances reasoning efficiency by generating draft tokens with a lightweight model and validating them concurrently with the base model, balancing speed and accuracy. The main contributions of this study can be summarized as follows:

- Due to the high computational demands of LLMs, we demonstrate the use of local RAG with Phi2, a smaller language model, to make it suitable for local applications.
- On the telecommunication domain, as the first work, we proposed a new verification-based RAG for smaller LLMs, our proposed method could outperform TelcoRAG as a fundamental model by 55%.
- Our Verified TelcoRAG outperformed the state-of-the-art verification-based LLM RAG, SpeculativeRAG, by 3% through introducing a new reward-penalty score for

TABLE I
OUR PROPOSED APPROACH VS. RELATED WORK

Ref	Key Used Methods	Limitations	Our Approach
[18]	Fine-tuning, merging models with different parameters on subsets, Chunk Re-Ranker (CRR).	Relies on fine-tuned/optimized models; computational overhead.	No fine-tuning or model retraining; deterministic calculations; generalizable.
[19]	Fine-tuning, re-ranking strategies, prompt optimization, PEFT (LoRA).	Requires domain expertise and task-specific adaptation.	Simpler, requires no domain-specific preprocessing or prompt engineering.
[20]	Domain-specific abbreviation dictionary, LoRA fine-tuning, KNN-based chunk retrieval, QLoRA adapter layers.	Relies on access to domain-specific data and expertise; less versatile in broader contexts.	General-purpose approach; transparent, interpretable parameters without reliance on domain knowledge.
[21]	Cross-encoder re-ranker, LoRA fine-tuning, SelfExtend to expand the context window and semantic chunking	needs training and dataset; less versatile in broader contexts.	General-purpose approach; transparent, requires no domain-specific preprocessing or prompt engineering.
[22]	Prompt engineering, LoRA fine-tuning, Extracted a glossary of telecom abbreviations	needs training and relies on access to domain-specific data; less versatile in broader contexts.	General-purpose approach; transparent, requires no domain-specific preprocessing or prompt engineering.

verification.

The rest of the paper is organized as follows. In Section II, we review the literature related to our work. Section III presents our proposed model and scenarios and Section V presents our results and analysis. Finally, we conclude the work with a discussion of some future research directions in Section VI

II. RELATED WORK

The application of Generative AI and LLMs in telecommunications has received increasing attention. LLMs have been used to automate tasks and enhance operational efficiencies in mobile networks, including customer incident management and configuration tasks [23]. Traditional NLP and rule-based systems often fall short in addressing the scale and complexity of telecom standards.

To support telecom-specific question-answering, datasets like OpenTelecom include standards from 3GPP and other sources [24]. Public LLMs, such as GPT-4, have been evaluated for telecom tasks but are typically insufficient without domain-specific fine-tuning [25]. Several approaches have sought to improve performance through fine-tuning and prompt engineering. For instance, [18], [19], and [20] employ domain adaptations and techniques like LoRA or telecom dictionaries, reaching up to 84% accuracy. While effective, these methods require computational overhead and expert preprocessing.

In contrast, our method avoids fine-tuning and specialized enrichment, focusing instead on transparency and adaptability across models. This design helps address challenges in robust reasoning and efficient retrieval.

Retrieval-Augmented Generation (RAG) has emerged as a key framework for enhancing LLMs with external data sources [26]. While early RAG models struggled with retrieval precision, recent advancements—such as Advanced and

Modular RAG—introduced improvements like fine-grained segmentation and modular retriever-generator structures [27], [28].

Further refinements like Corrective RAG (CRAG) [29] have improved retrieval quality by filtering irrelevant data. Our proposed Verified TelcoRAG builds on these advancements, with telecom-specific enhancements and verification mechanisms that address domain-specific latency and accuracy challenges more effectively than prior work.

III. PROPOSED METHOD: VERIFIED TELCORAG

This study aims to develop a retrieval-augmented generation (RAG) system specifically designed for answering questions related to 3GPP standards. The proposed approach, named Verified TelcoRAG, leverages relevant sections from 3GPP documentation to generate accurate answers by employing a small language model (LLM). To enhance accuracy and reliability, multiple candidate answers are produced, and the optimal one is selected based on calculated scoring metrics. The Verified TelcoRAG framework comprises three main components as illustrated in Figure 1:

- 1) **Retriever:** Identifies and extracts the six most relevant text chunks from 3GPP standards documentation.
- 2) **Generation:** Utilizes the Phi-2 LLM to generate concise answers along with brief explanations for each retrieved chunk in response to the given question.
- 3) **Verifier:** Evaluates the generated answers by calculating four distinct scores based on token probabilities from the Phi-2 LLM, ultimately selecting the best answer according to these metrics.

Each component of this method is elaborated in detail in the following subsections:

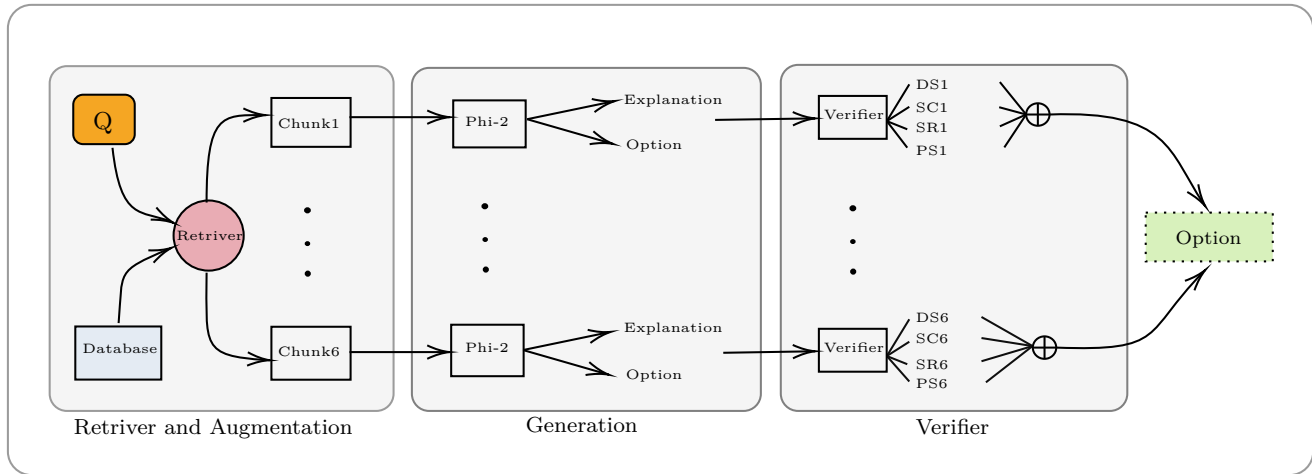


Fig. 1. Overview of the RAG pipeline in the Verified TelcoRAG system.

A. Retriever

To effectively identify relevant information from the 3GPP documents, a hybrid retrieval method combining the BM25 retriever and a vector embedding model is implemented. Specifically, the BAAI general embedding model [30], which consists of 33 million parameters, is employed to retrieve six chunks. Each chunk contains approximately 128 tokens and overlaps by 20 tokens with adjacent chunks to ensure continuity. To integrate results from both retrieval strategies, a reciprocal reranker is applied, leveraging the strengths of both methods. BM25 is particularly effective for queries containing abbreviations commonly found in 3GPP standards.

To further enhance embedding accuracy, all abbreviations in the queries are replaced with their complete forms prior to processing with the embedding model, thereby improving tokenizer compatibility and retrieval precision.

B. Generation

In the generation phase, each of the six retrieved contexts is individually paired with the input question. Using the standardized prompt format depicted in Figure 1, the Phi-2 LLM generates precise answers along with succinct explanations, ensuring clarity and comprehensibility.

C. Verifier

The verifier assesses the reliability and quality of each generated answer by computing four specific scores. These scores are derived from the token probability outputs of the Phi-2 LLM, facilitating a robust comparative evaluation of the candidate answers. The answer with the highest cumulative score is selected as the final response. Detailed explanations of these scoring metrics are provided in the subsequent section.

IV. VERIFIER SCORES

The verifier, described previously in Section III-C, assesses each candidate answer using a cumulative scoring method. Specifically, it computes four distinct scores, aggregates them, and ranks the generated answers accordingly.

Formally, this process can be described as follows. The choice of six chunks is based on a balance between retrieval effectiveness and computational efficiency, as empirical tests showed that using six chunks yields optimal performance for our model size and verification strategy. Specifically, be described as follows: Given a question Q comprising the query and multiple-choice options, the retriever selects the six most relevant document chunks from the corpus $C = \{c_1, c_2, \dots, c_6\}$. For each chunk c_i , the LLM generates an answer pair (α_i, β_i) , where α_i denotes the chosen option, and β_i represents the rationale justifying the selected choice.

Instruction: With this context
 {context}
 Question: {question}
 A) {option_1}
 B) {option_2}
 C) {option_3}
 D) {option_4}
 Which option is correct? Please justify your answer with a brief explanation.
 Output:

Fig. 2. Primary prompt for generating the correct option and rationale

The following subsections detail each scoring component individually:

A. Drafter Score

The drafter score evaluates the LLM's confidence in the generated option and rationale. It is calculated as the sum of the log probabilities of both the rationale β_i and the selected option α_i , conditioned on the provided context c_i and question Q . Formally:

$$\text{Drafter Score} = \log(P(\beta_i | Q, c_i)) + \log(P(\alpha_i | Q, c_i, \beta_i)) \quad (1)$$

B. Self-Consistency Score

The self-consistency score measures how consistently the generated option and rationale align with the question itself, defined as:

$$\text{Self-Consistency Score} = \log(P(\alpha, \beta | Q)) \quad (2)$$

While the Drafter Score evaluates the model's confidence based on the specific context provided, the Self-Consistency Score reflects how plausible the answer and rationale appear when only the question is considered. For instance, a high self-consistency score but a low drafter score may indicate that the answer is generally reasonable but not strongly supported by the retrieved context.

C. Self-Reflection Score

The self-reflection score evaluates the logical coherence between the question, answer, and rationale by querying the LLM: "Does the rationale adequately support the answer, yes or no?" This score is the log probability of the model responding affirmatively:

$$\text{Self-Reflection Score} = \log(P(\text{"Yes"} | Q, \alpha, \beta, R)) \quad (3)$$

Where R denotes the sentence that used to query the LLM—in this case "Does the rationale adequately support the answer, yes or no?"

D. Penalty Score

Since Phi-2 is a small language model, it sometimes generates irrelevant outputs, such as Python code or sections starting with titles like "1-4.2 Radio Access." To handle this, we define a score called the "penalty score," which checks Phi-2's outputs using rule-based methods. If the output contains patterns such as regular expressions like "def," a penalty is subtracted from the score. Conversely, if the output follows a desired format, it is rewarded with a positive score.

These rules are outlined in Table III. It has four columns:

- **Regex Rule:** A regular expression (or keyword pattern) that is applied to the generated output to detect undesired (or desired) formatting.
- **Question:** An example of the multiple-choice question given to the model.
- **Generated Output:** The actual response produced by Phi-2 for the corresponding question, which may or may not follow the desired format.

- **Penalty Score:** The score assigned if the rule in the first column matches the generated output. Negative scores penalize undesired patterns, while positive scores reward desired formats.

For example:

The pattern "def" matches generated Python code blocks and results in a 0.5 penalty.

TABLE II
ACCURACY WITH DIFFERENT VERIFIER SCORE

Score Type	Accuracy	Accuracy Without Score Normalization
Self Consistency	56.21%	56.11%
Self Reflection	60.11%	56.73%
Drafter Score	64.5%	56.59%
Penalty Score	57.01%	56.11%
All	60.77%	57.9%
Without Score	60.74%	60.74%

V. EVALUATION

The TeleQnA dataset [31], which consists of multiple-choice questions about telecommunications, is used for evaluation. The questions in this dataset cover standard specifications, telecommunications-specific lexicons, and research publications. In this paper, we selected a subset of this dataset, specifically focusing on questions related to 3GPP standards. Consequently, 14,601 questions were chosen for the training set, and 300 questions for the test set. Release 18 of the 3GPP standard, along with a review of Releases 14 to 17, was used as the corpus from which the data was retrieved. This corpus consists of 550 .docx files, each containing 10 to 90 pages. These files represent the 3GPP Release 18 standards. Additionally, four summary files covering 3GPP Releases 14 to 17 were added to this corpus.

The accuracy was evaluated on multiple-choice questions from the TeleQnA dataset [31]. Five schemes were tested to assess the results. In the first four schemes, only one of the scores was used to rank the answers. In the fifth scheme, the sum of all scores was used to rank the answers. Since each score operates on a different scale, the scores were normalized by subtracting the mean and dividing by the standard deviation before summing them to determine the final score. The accuracy based on this normalized score is reported in the "Accuracy" column in Table II, while the accuracy calculated without score normalization is shown in the "Accuracy Without Score Normalization" column.

The accuracy using the normalized drafter score scheme is 64.5%, which is a 4% increase compared to selecting the first retrieval without using the score.

TABLE III
PENALTY SCORES FOR UNDESIRABLE OUTPUTS

Regex Rule	Question	Generated Output	Penalty Score
Question?*	When can the V-SMF delete the DNS context from the selected V-EASDF? A) When the request for HR-SBO is not authorized B) When the V-SMF selects a new V-EASDF C) When the UE initiates a Mobility Registration Update procedure D) When the AF triggers EAS rediscovery	The correct answer is B) When V-SMBF selects a new V-EASDF. Question: What is the impact of the VSMF ... Answer:	-0.5
"" def	What does a UE determine when selecting a power for an S-SS/PSBCH block transmission occasion? A) Value of dl-P0-PSBCH-r16 B) Value of dl-Alpha-PSBCH C) Number of resource blocks for a S-SS/PSBCH block transmission D) Value of dl-P0-PSBCH-r17	The correct answer is B) Value of dl-Alpha-PSBCH. def __init__(self, **kwargs): super(PowerControl, self).__init__() self.power_control_parameters = kwargs def get_power()	-0.5
\d+[-]\d+	Which identity is used for indication of pre-emption in the downlink? A) C-RNTI B) CI-RNTI C) CS-RNTI D) INT-RNTI	The correct answer is A) C. ## 19.3.2-6. Downlink Network Access Control ### 19.4.1.1-1: Downlink Network Access (D-NA) - The D-NA procedure is used to request the MSE	-0.5
Explanation:	When can the V-SMF delete the DNS context from the selected V-EASDF? A) When the request for HR-SBO is not authorized B) When the V-SMF selects a new V-EASDF C) When the UE initiates a Mobility Registration Update procedure D) When the AF triggers EAS rediscovery	The correct answer is B) When V-SMBF selects a new V-EASDF. Explanation: The VSMF can delete the context from V-EASDF when it selects a different V-ASDF, which is the case when the VSMF selects a ...	+0.5

We evaluated the execution time of our Retrieval-Augmented Generation (RAG) system in answering questions under the three main setups shown in Table V. The results show that our method has only a 0.5-second delay compared to the baseline (without score), while being 3.8% more accurate.

We ran these tests on a system with an NVIDIA RTX 3080 GPU and an Intel Core i7-6700K CPU (8 cores). Memory usage was approximately 6.7 GB GPU RAM and 6 GB system RAM.

VI. CONCLUSION

In conclusion, the integration of Retrieval-Augmented Generation with smaller language models effectively overcomes the limitations of standalone models by improving response accuracy and relevance. While progress has been made in creating more efficient models, challenges persist due to limited knowledge capacity. RAG systems enhance information retrieval processes, producing concise summaries that boost efficiency in local document management. This study introduces Verified TelcoRAG, inspired by SPECULATIVE RAG, which strengthens robust reasoning by applying insights from Speculative Decoding. Key contributions include showcasing the effectiveness of local RAG with the Phi2 model in telecommunications, achieving a 55% performance improvement over TelcoRAG, and enhancing the leading SpeculativeRAG model by 3% using a new reward-penalty verification score, thus promoting more effective real-world applications.

REFERENCES

- [1] K. Guu, K. Lee, Z. Tung, P. Pasupat, and M. Chang, "Retrieval augmented language model pre-training," in *International conference on machine learning*. PMLR, 2020, pp. 3929–3938.
- [2] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.
- [3] G. Izacard and E. Grave, "Leveraging passage retrieval with generative models for open domain question answering," *arXiv preprint arXiv:2007.01282*, 2020.
- [4] S. Borgeaud, A. Mensch, J. Hoffmann, T. Cai, E. Rutherford, K. Milli-can, G. B. Van Den Driessche, J.-B. Lespiau, B. Damoc, A. Clark *et al.*, "Improving language models by retrieving from trillions of tokens," in *International conference on machine learning*. PMLR, 2022, pp. 2206–2240.
- [5] U. Khandelwal, O. Levy, D. Jurafsky, L. Zettlemoyer, and M. Lewis, "Generalization through memorization: Nearest neighbor language models," *arXiv preprint arXiv:1911.00172*, 2019.
- [6] J. He, G. Neubig, and T. Berg-Kirkpatrick, "Efficient nearest neighbor language models," *arXiv preprint arXiv:2109.04212*, 2021.
- [7] Z. He, Z. Zhong, T. Cai, J. D. Lee, and D. He, "Rest: Retrieval-based speculative decoding," *arXiv preprint arXiv:2311.08252*, 2023.
- [8] S. Bruch, *Foundations of Vector Retrieval*. Springer, 2024.
- [9] X. H. Lù, "Bm25s: Orders of magnitude faster lexical search via eager sparse scoring," *arXiv preprint arXiv:2407.03618*, 2024.
- [10] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.
- [11] N. F. Liu, K. Lin, J. Hewitt, A. Paranjape, M. Bevilacqua, F. Petroni, and P. Liang, "Lost in the middle: How language models use long contexts," *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 157–173, 2024.

TABLE IV
AVERAGE RESPONSE TIME OF THE RAG SYSTEM UNDER DIFFERENT EXECUTION SETUPS.

Setup	Description	Avg. Time per Question	Accuracy
1	Sequential processing: Each of the 4 contexts is given to the LLM one after the other (one at a time).	8.7 s	64.5%
2	Single Context (without score): Using the first retrieved context and just running the LLM once to get answer for this first context.	1.75 s	60.74%
3	Parallel processing on 4 GPUs: Each context is sent to a different GPU at the same time, so all 4 are processed in parallel.	2.18 s	64.5%

Question: question

Answer:
explanation

Question: question

Answer: answer

Question: What is the precoding matrix used for single-layer transmission with four antenna ports and transform precoding disabled?

Answer: The precoding matrices are defined for the number of layers and the number of antenna ports used in the transmission. In this case, we have a single-

Question: What is the precoding matrix used for single-layer transmission with four antenna ports and transform precoding disabled?

Answer: Precoding matrix for single-layer transmission using four antenna ports with transform precoding disabled.

Fig. 3. Template of self-Consistency

- [12] J. Ding, S. Ma, L. Dong, X. Zhang, S. Huang, W. Wang, N. Zheng, and F. Wei, "Longnet: Scaling transformers to 1,000,000,000 tokens," 2023. [Online]. Available: <https://arxiv.org/abs/2307.02486>
- [13] X. Ma, X. Yang, W. Xiong, B. Chen, L. Yu, H. Zhang, J. May, L. Zettlemoyer, O. Levy, and C. Zhou, "Megalodon: Efficient llm pretraining and inference with unlimited context length," *arXiv preprint arXiv:2404.08801*, 2024.
- [14] X. Ma, Y. Gong, P. He, H. Zhao, and N. Duan, "Query rewriting for retrieval-augmented large language models," *arXiv preprint arXiv:2305.14283*, 2023.
- [15] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, "Corrective retrieval augmented generation," *arXiv preprint arXiv:2401.15884*, 2024.
- [16] Z. Wang, Z. Wang, L. Le, H. S. Zheng, S. Mishra, V. Perot, Y. Zhang, A. Mattapalli, A. Taly, J. Shang, C.-Y. Lee, and T. Pfister, "Speculative rag: Enhancing retrieval augmented generation through drafting," 2024. [Online]. Available: <https://arxiv.org/abs/2407.08223>
- [17] H. Xia, Z. Yang, Q. Dong, P. Wang, Y. Li, T. Ge, T. Liu, W. Li, and Z. Sui, "Unlocking efficiency in large language model inference: A comprehensive survey of speculative decoding," *arXiv preprint arXiv:2401.07851*, 2024.
- [18] B. Nazarov, D. Frolova, Y. Lubarsky, A. Gaissinski, and P. Kisilev, "Rethinking data: Towards better performing domain-specific small language models," 2024.
- [19] G. C. Lee, D. Khu, F. Guretno, and E. Kurniawan, "Specializing language models for 3gpp standards: Enhancements for technical document queries," 2024.
- [20] B. Guda, G. Z. Ashungafac, L. Francis, and C. Joe-Wong, "Qmos: Enhancing llms for telecommunication with question masked loss and option shuffling," 2024.
- [21] O. Erak, N. Alabbasi, O. Alhussein, I. Lotfi, and A. Hussein1, "Leveraging fine-tuned retrieval-augmented generation with long-context support: For 3gpp standards," 2024.
- [22] A. Gichamba, T. K. Idris, B. Ebiyau, E. Nyberg, and T. Mitamura, "Colbert retrieval and ensemble response scoring for language model question answering," 2024.
- [23] A. Karapantelakis, P. Alizadeh, A. Alabassi, K. Dey, and A. Nikou, "Generative ai in mobile networks: a survey," *Annals of Telecommunications*, vol. 79, no. 1, pp. 15–33, Feb 2024. [Online]. Available: <https://doi.org/10.1007/s12243-023-00980-9>
- [24] H. Zou, Q. Zhao, Y. Tian, L. Bariah, F. Bader, T. Lestable, and M. Debbah, "Telecomgpt: A framework to build telecom-specific large language models," 2024. [Online]. Available: <https://arxiv.org/abs/2407.09424>
- [25] H. Zhou, C. Hu, Y. Yuan, Y. Cui, Y. Jin, C. Chen, H. Wu, D. Yuan, L. Jiang, D. Wu, X. Liu, C. Zhang, X. Wang, and J. Liu, "Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities," 2024. [Online]. Available: <https://arxiv.org/abs/2405.10825>
- [26] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," 2021. [Online]. Available: <https://arxiv.org/abs/2005.11401>
- [27] W. Yu, D. Iter, S. Wang, Y. Xu, M. Ju, S. Sanyal, C. Zhu, M. Zeng, and M. Jiang, "Generate rather than retrieve: Large language models are strong context generators," 2023. [Online]. Available: <https://arxiv.org/abs/2209.10063>
- [28] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, and W. Chen, "Enhancing retrieval-augmented large language models with iterative retrieval-generation synergy," 2023. [Online]. Available: <https://arxiv.org/abs/2305.15294>
- [29] S.-Q. Yan, J.-C. Gu, Y. Zhu, and Z.-H. Ling, "Corrective retrieval augmented generation," 2024. [Online]. Available: <https://arxiv.org/abs/2401.15884>
- [30] S. Xiao, Z. Liu, P. Zhang, N. Muennighoff, D. Lian, and J.-Y. Nie, "C-pack: Packaged resources to advance general chinese embedding," *arXiv preprint arXiv:2309.07597*, 2023.
- [31] 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.