QMOS: Enhancing LLMs for telecommunications with question-masked loss and option shuffling

Blessed Guda [1]***, Gabrial Zencha Ashungafac*** [1]***, Lawrence Francis*** [1]***, Carlee Joe-Wong*** [1]

[1] *College of Engineering, Carnegie Mellon University*

Corresponding author: Gabrial Zencha Ashungafac, gzenchaa@andrew.cmu.edu

Large Language models (LLMs) have brought about substantial advancements in the field of Question Answering (QA) systems. These models do remarkably well in addressing intricate inquiries in a variety of disciplines. However, because of domain-specific vocabulary, complex technological concepts, and the requirement for exact responses, applying LLMs to specialized sectors like telecommunications presents additional obstacles. GPT-3.5 has been used in recent work, to obtain noteworthy accuracy for telecommunication-related questions in a Retrieval Augmented Generation (RAG) framework. Notwithstanding these developments, the practical use of models such as GPT-3.5 is restricted by their proprietary nature and high computing demands. This paper introduces Question-Masked Option Shuffle (QMOS), an innovative approach which uses a question-masked loss and option shuffling trick to enhance the performance of LLMs in answering multiple-choice questions in the telecommunication domain. We focus on using open-source, smaller language models (Phi-2 and Falcon-7B) within an enhanced RAG framework. Our multi-faceted approach involves several enhancements to the whole LLM-RAG pipeline of fine-tuning, retrieval, prompt engineering and inference. Our approaches significantly outperform existing results, achieving accuracy improvements from baselines of 24.70% to 49.30% with Falcon-7B and from 42.07% to 84.65% with Phi-2. All code developed for this work is publicly available as open-source.[a]

Keywords: Large language models, option batch-shuffle trick, question-masked loss, RAG, telecommunication

[a] https://github.com/ai4africagroup/telcom_llm

1. INTRODUCTION

The field of Question Answering (QA) systems has witnessed significant advancements with the advent of Large Language Models (LLMs) [1]. These models have demonstrated remarkable capabilities in understanding and responding to complex queries across various domains. However, their application to specialized fields, such as telecommunications, presents unique challenges due to domain-specific terminology, intricate technical concepts, and the need for precise, accurate responses [2]. Recent research has shown promising results in applying LLMs to telecommunication-specific QA tasks. A notable study, Telco-RAG achieved accuracies around 75% using GPT-3.5 within a Retrieval-Augmented Generation (RAG) framework for telecommunication-related questions [3]. While this demonstrates the potential of LLMs in domain-specific applications, the computational demands and resource requirements of such models often pose significant challenges for practical, widespread implementation [4]. Also, the use of GPT-3.5 de-emphasizes open-source AI, as the training data details and architecture of GPT-3.5 are not open. This paper explores an innovative approach to addressing Multiple-Choice Questions (MCQs) in the telecommunication domain using *open, small language models within an enhanced RAG framework.* Answering telecommunication standards questions presents a particularly challenging environment for LLMs due to their frequent use of unique abbreviations and rapidly evolving technologies and regulatory considerations [2]. To further complicate the problem, LLMs have shown selection bias when answering MCQs, which significantly degrades performance [5]. In addition, due to rapid changes in the specifications and data sparsity, large language models for telecommunications require frequent fine-tuning to keep them up to date with the latest telecommunication standards. Our research aims to achieve competitive performance of larger models while maintaining efficiency and blackucing computational costs. We focus on Phi-2 [6] and Falcon-7B [7], which have shown promise in achieving competitive performance with significantly fewer parameters than their larger counterparts. Our work addresses these challenges through a multi-faceted approach called Question-Masked Option Shuffle (***QMOS***) (Question Masked Option Shuffle) that combines several novel techniques:

1. We leverage multiple embedding models to diversify and enrich the documents retrieved by the RAG system, potentially capturing a broader range of relevant information.
2. Due to the high number of abbreviations used in telecommunication standards, we enhance the dictionary of abbreviations used in [3] to boost the hit rate of successful abbreviation expansions.
3. We meticulously design our prompts to guide the model to reason over the documents when selecting the answer.
4. We employ Low-Rank Adaptation (LoRA) fine-tuning on the Phi-2 model with *a question-masked loss function* to efficiently adapt it to the telecommunication domain.
5. We implement an innovative optimization technique: inference and train time option batch-shuffling, which enhances the accuracy of the Phi-2 model by eliminating bias in the correct option position.

By focusing on Small Language Models (SLMs) and employing these advanced techniques, our work addresses the growing need for efficient, scalable NLP solutions in specialized domains. This research has significant implications for developing cost-effective QA systems that can be deployed in resource-constrained environments while maintaining high accuracy.

The paper is organized as follows: Section 2 reviews the current state of LLMs and QA in telecommunications. Section 3 outlines our methodology, including the RAG architecture, model fine-tuning and batch shuffling. Section 4 evaluates its effectiveness and Section 5 concludes with future research directions.

2. RELATED WORK

SLMs have emerged as a promising alternative to large models for domain-specific applications. Models like Phi-2, Falcon-7B and TinyLlama-1.1B [8] offer advantages such as resource efficiency, faster inference, and easier fine-tuning for specific domains [9]. In [10], Piovesan et al. (2024) conducted a comprehensive evaluation of the small language model Phi-2 in the telecommunication domain, comparing it to larger models like GPT-3.5 and GPT-4. Their findings demonstrate that Phi-2, despite being significantly smaller, achieved an overall accuracy of 52.30% on the TeleQnA dataset [2], compablack to 67.29% for GPT-3.5 and 74.91% for GPT-4 which have an estimated 175B and 1T parameters respectively . Notably, when enhanced with Retrieval-Augmented Generation (RAG), Phi-2's performance in the challenging "Standards Specifications" category improved from 44.27% to 56.63%, nearly matching GPT-3.5's 56.97%. In another study, [11] conducted a comprehensive evaluation of several small language models, including Falcon 7B, in the telecommunications domain. Their findings demonstrate that Falcon 7B, despite having 7 billion parameters, achieved an overall accuracy of only 15.70% on the TeleQnA dataset, significantly lower than larger models like GPT-3.5 (67.29%) and GPT-4 (74.91%). While there is limited study on improving SLMs for telecommunication application, there has been an exhausive study in improving the performance of large language models. In [12] , it was shown that fine-tuning and RAG are both feasible methods in improving the performance of language models.

These studies show that the SLMs significantly lag begin the larger models in the telecommunication domain. In this research, we show that it is possible to match and even outperform the performance of LLMs with SLMs in the telecommunication domain.

3. PROPOSED METHODOLOGY

We propose enhancing small language models like Phi-2 and Falcon-7B for telecommunication MCQ answering using an improved RAG framework with custom chunking, prompt engineering, and LoRA fine-tuning. We also introduce an option batch-shuffling technique to blackuce selection bias, achieving accuracy comparable to larger models while maintaining efficiency.

3.1 Retrieval-augmented generation architecture

Retrieval-augmented generation is an interesting technique used to enhance the performance of large language models on tasks where the requiblack knowledge was not present in the training data [13]. RAG is achieved by integrating external knowledge sources in the prompt from which the LLM's result is generated. In question answering tasks, questions are augmented with contextually relevant texts from external documents when creating prompts.

3.1.1 Splitting documents into chunks

For RAG, we only need relevant parts of a large number of documents, and this requires a search across these documents. A common approach being employed by commercial tools for RAG such as LlamaIndex [14] and LangChain [15] splits the document so that each chunk contains a certain number of characters with an overlap between chunks. The parameters 'chunk-size' and 'chunk-overlap' are often adjusted so that the split leads to meaningful chunks ('chunk-size' specifies the number of characters in a chunk of text while 'chunk-overlap' specifies the overlap of texts between two contiguous chunks). We employed *a custom document splitting strategy* where we first split the documents into individual sections, excluding the table of contents. Each section is further split into chunks containing 'chunk-size' characters. In our setting, the chunk-size is 1024 characters. Additionally, we prepended the heading of each section to its chunks. This chunking strategy is inspiblack by our analysis of the structure of the 3GPP standard documents. Also, we observe that the first few pages of the documents containing sections like the title page, scope, references and table of contents are not very informative and as such we do not include those sections when creating the chunks. Having obtained the necessary chunks, we proceed to create vector embeddings of these chunks.

3.1.2 Creating chunk embeddings

To enable similarity search, we created associated embeddings for each chunk using an embedding model, as shown in Fig. 1. The choice of embedding model was influenced by models on the Massive Text Embedding Benchmark (MTEB) leaderboard [16]. The MTEB evaluates text embedding models across eight diverse tasks and 58 datasets in over 100 languages, offering a comprehensive leaderboard to guide embedding model selection. We opted for the best performing and easily accessible models, favouring the use "stella_en_400M_v5" and "gte-Qwen2-1.5B-instruct" [17] models with 400M and 1.5B parameters , respectively. We used these text embedding models from the 'Sentence Transformer' library [18] and sped up the embedding process by batching the chunks, using a batch size of 64. The chunks and their corresponding embeddings were saved to disk to be used for context retrieval for the question answering task.

3.1.3 Chunk retrieval

For the chunk retrieval process, we employed a k-Nearest Neighbors (kNN) approach on the similarity scores between a question/query embedding and the chunk embeddings [19]. Our application of the kNN uses the dot product similarity score to find the nearest neigbhours. A similarity score shows a closer neighbour. Hence, we retrieve the top k similar chunks for a given question/query. The number of retrieved chunks is chosen so that the context length of the language model is not exceeded when creating the prompt. We used the top two chunks retrieved with each embedding model. Fig. 1 shows how chunks are retrieved to form a context in the input prompt to an LLM.

Additionally, we employed the use of the BM25 [20] algorithm which is a statistical approach for information retrieval that measures similarity based on the frequency of terms from the query that appear in the chunks. The motivation behind this is to ensure that the retrieval also includes chunks containing specific terms used in the query, that are not necessarily enforced in the neural embedding models. This enabled us to create a context that consists of chunks from two embedding models (stella_en_400M_v5 and gte-Qwen2-1.5B-instruct [17]) and BM25 [20].

3.1.4 Model prompt

Prompt engineering is crucial in RAG-based LLM systems as it significantly enhances their performance and reliability. By carefully crafting prompts, engineers can guide LLMs to produce more accurate, relevant, and contextually appropriate responses without the need for extensive fine-tuning [21]. This is particularly important in RAG systems, where the integration of external knowledge sources in the prompts helps mitigate issues like hallucinations and factual inaccuracies [22]. In designing our prompt we make the following considerations:

- **Question repitition:** We draw from the observation of [3], which showed that in answering Telecommunication questions, repeating the question before and after the contexts helps make the Phi-2 model consider the contexts for the answer.
- **Enhanced abbreviation expansion:** As we noticed that a lot of the questions in the TeleQnA dataset are about abbreviations, we also decide to include the abbreviations in the prompt like [3]. However, we notice that the method in [3] missed a lot of abbreviations because of the insufficient dictionary of abbreviations. This is because the abbreviation dictionary used was generated by only considering the Vocabulary for 3GPP specifications document [23]. We expand this dictionary by searching for other abbreviations in the

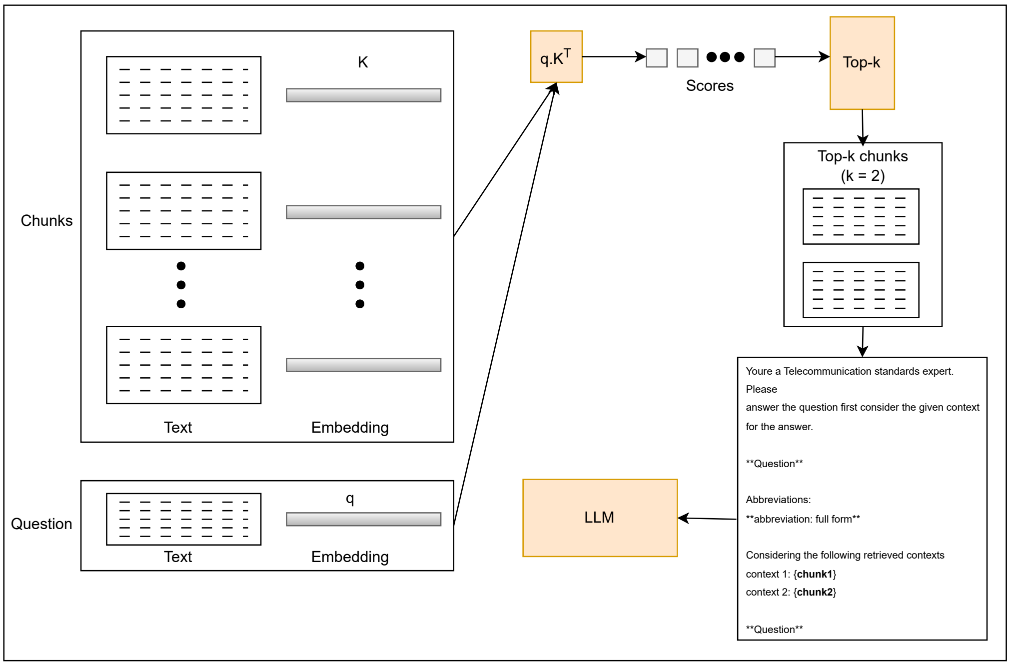

Figure 1 – Chunk retrieval process: The top K chunks are retrieved using the KNN algorithm. Scores for each chunk is obtained from the dot product $q \cdot K^T$. The retrieved chunks are then used to create the input prompt to the LLM

"*Definitions of terms, symbols and abbreviations*" sections of all the documents. With this enhanced abbreviation dictionary, we are able to achieve a hit rate of **95.16%** in the test set which is a significant improvement over the **63.74%** achieved using the approach in [3].

- **Model prompt format**: We consider how the prompts in the training phase of the model were structublack. This is crucial because presenting the model with prompts that are structublack based on how it's trained can help the model perform better.

Prompt for Phi-2

With the above considerations, we design our prompt for Phi-2 as:

Instruct: ***Question***

Abbreviations:

abbreviation: full form

Considering the following retrieved contexts

context 1: context...
context 2: context...

Question

** option 1: ....

** option 2: ....
Output :

For example, consider a question from the TeleQnA dataset given as:

> Question: "According to ETSI EN 301 441, what is the maximum degradation in the receiver signal to noise ratio allowed for the adjacent signal compablack to the wanted signal?"
> Option 1: "1 dB"
> Option 2: "0 dB"
> Option 3: "2 dB"
> Option 4: "3 dB"
> Option 5: "0.5 dB"

And retrieved contexts from RAG:

> "According to ETSI EN 301 441 (version 2.1.1) the maximum degradation in the receiver signal-to-noise ratio (SNR) due to an unwanted in-band (“adjacent”) signal is 1 dB"

The prompt for Phi-2 is designed as

Instruct: According to ETSI EN 301 441, what is the maximum degradation in the receiver signal to noise ratio allowed for the adjacent signal compablack to the wanted signal?

Abbreviations:
ETSI → European Telecommunications Standards Institute
EN → Euronorm

Context 1: According to ETSI EN 301 441 (version 2.1.1) the maximum degradation in the receiver signal-to-noise ratio (SNR) due to an unwanted in-band (“adjacent”) signal is 1 dB

According to ETSI EN 301 441, what is the maximum degradation in the receiver signal to noise ratio allowed for the adjacent signal compablack to the wanted signal?

Option 1: 1 dB
Option 2: 0 dB
Option 3: 2 dB
Option 4: 3 dB
Option 5: 0.5 dB

Output:

3.1.5 Falcon-7B

We make slight modifications to the prompt for the Falcon 7B based on its poor performance on the TeleQnA dataset [11]. We observed that when provided with the options, the model does not do well; therefore, we do not provide the options to the model.

Prompt for Falcon-7B

Youre a telecommunication standards expert. Please answer the question first considering the given context for the answer.

Question

Abbreviations:

abbreviation: full form

Considering the following retrieved contexts

context 1: context...

context 2: context...

Question

From the answer sentence generated by the model, we compute the embeddings of the generated answer sentence and take the answer option that has the highest similarity. We use a simple sentence BERT model [24] for this answer extraction.

3.2 QMOS fine-tuning approach

3.2.1 Phi-2 fine-tuning

To finetune the Phi-2 model, we used LoRA (Low-Rank Adaptation) [25] fine-tuning approaches instead of full fine-tuning due to the size of the model and the small size of the training data. We exploblack both LoRa and Quantized LoRa (QLoRa) [26] fine-tuning. We did not explore fine-tuning the Falcon 7B model, due to its large size, which exceeds our computational budget (NVIDIA L40s with 48GB of GPU memory).

3.2.2 QLoRa

Quantized low-Rank Adaptation (QLoRA) is a parameter-efficient fine-tuning technique for large language models. It combines quantization and low-rank adaptation to significantly blackuce memory requirements and computational costs. In QLoRA, the base model's weights are quantized to 4-bit precision and frozen. Then, small trainable "adapter" layers are added using low-rank decomposition. We envision that these adapters would capture the telecommunication MCQ reasoning ability during fine-tuning while keeping most of the model fixed.

We configure the model for QLoRA fine-tuning with the following parameters.

- Low rank: Set to 64 to balance between performance and computational efficiency.
- Alpha : Set to 16 to scale the low-rank updates. (This controls the magnitude of the LoRA update relative to the original weights.)
- Dropout: Set to 0.05 to prevent overfitting during training.
- Adapter layers: We add adapter layers to the query, key, and value and feedforward weights of the transformer layers of the Phi-2 model.

3.2.3 LoRa

The use of QLoRA fine-tuning blackuces the memory requirement during training. However, the training time is considerably increased due to the quantization and dequantization operations being performed during QLoRA fine-tuning. In order to investigate the influence of the fine-tuning objective on a model's performance on MCQs, we used LoRA since it offers faster training time compablack to QLoRA. We compablack the model's performance when the training objective only considers the answers versus when the entire prompt and answer are consideblack in the objective.

The original cross-entropy loss for next token pblackiction is defined as:

$$\mathcal{L} = -\sum_{t=1}^{T} y_t \log(\hat{y}_t) \tag{1}$$

where:

- T is the total number of tokens in the sequence (including both the question/prompt and the answer).
- y_t is the actual token at position t (one-hot encoded).
- $\hat{y}_t$ is the pblackicted probability distribution over the vocabulary for token t.

To focus only on answer generation, we introduce a masking vector m_t so that:

$$m_t = \begin{cases} 0 & \text{if } t \in \{1, 2, \dots, Q\} \\ 1 & \text{if } t \in \{Q+1, Q+2, \dots, T\} \end{cases} \tag{2}$$

The modified cross-entropy loss, which we call question-masked loss $\mathcal{L}_{\text{masked}}$ is then:

$$\mathcal{L}_{\text{masked}} = -\sum_{t=1}^{T} m_t \cdot y_t \log(\hat{y}_t) \tag{3}$$

Q here is the number of question tokens. This loss function ensures that only the tokens corresponding to the answer part of the sequence contribute to the overall loss, effectively masking out the contributions from the question/prompt part.

3.3 Option batch-shuffle trick

Recent research [2, 5, 27] has unveiled a significant bias in LLMs when answering Multiple-Choice Questions(MCQs). These models exhibit a strong sensitivity to the order of options, often selecting specific answer positions regardless of the content. This phenomenon, termed "selection bias," stems from the models' tendency to assign higher probabilities to certain option labels (like "A" or "B" in options ["A", "B", "C", "D" and "E"]). Consequently, LLMs may prioritize these options even when logically incorrect, undermining the reliability of their performance on MCQ assessments. To avert this we employ a trick where we create multiple prompts for a question, with each prompt having a different option order. The correct answer is thus determined by choosing the most selected answer by the model after observing the answers generated for all created prompts. Given that we have to permute these options to obtain all possible option ordering, the complexity of doing so is $O(n!)$ where n is the number of options present in an MCQ. This complexity significantly increases the inference time for a single question. For example, when an MCQ has four options, we create 4! = 24 prompts instead of 1 prompt. For five options, we create 5! = 120 prompts. To blackuce this complexity, we randomly sample k prompts from the n! prompts, create a batch of k prompts and generate answers for the batch using the Phi-2 model. Using k prompts instead of $n!$ blackuces the complexity from $O(n!)$ to $O(k)$, where $k \ll n!$. The model generates answers for these k prompts in a single batch, thus significantly blackucing inference time while still benefiting from diverse option orderings. The selection of the most frequent answer from these k prompts can be described as:

$$\hat{a} = \arg\max_{a \in A} \sum_{i=1}^{k} \mathbb{I}(a_i = a) \tag{4}$$

where $\mathbb{I}(\cdot)$ is the indicator function at returns 1 if (.) is true and 0 otherwise, a_i is the answer chosen by the model for the i-th prompt, and A is the set of all possible answers.

We call this the *'batch-shuffle trick'*. Using the batch-shuffle trick, we noticed accuracy improvements of over 6% when we apply the trick only at inference time, and this increased to about 10% when we include the trick both during training and inference. In the training phase, we only shuffle the options at the end of each epoch and do not use any explicit sampling. We hypothesize further improvement in performance as we increase k, the number of samples from n! prompts of an MCQ. In our case, we find $k = 20$ to be a good balance between efficiency and accuracy.

4. EVALUATION

We evaluated our approach using a subset of the TeleQnA dataset containing only two categories: standards specifications and standards overview. Matoouk *et al* [2] showed that GPT-3.5 and GPT-4 performed better in other question categories than in these two categories. The dataset, obtained from the "Specializing Large Language Models for Telecom Networks by ITU AI/ML in

5G Challenge" on Zindi [28] contains a trained set of 1461 questions, a public test set of 366 questions, and a private test set of 2000 questions. Evaluation results, as obtained from the leaderboard, are reported for the private test set. The trained set was used for fine-tuning purposes. Additionally, we used the technical documents provided by the challenge as external knowledge sources for RAG.

We compare the performance of the base Phi-2 model and its performance with RAG, with fine-tuning and with the batch-shuffle trick both at inference and at training time. For the Falcon7B, we compare the base model performance and its performance with RAG and with the options excluded in the prompt.

4.1 Phi-2 model

Table 1 shows the performance of the Phi-2 model on the private test sets as obtained from the submissions on Zindi. The accuracy score measures the percentage of correctly answeblack questions. The base Phi-2 model has an accuracy of 42.07%, and this accuracy was increased to 66.39% with the introduction of RAG. The result obtained with RAG was further improved by fine-tuning the model using the training set. With fine-tuning, the accuracy increased to 76.90%. Using the batch-shuffle trick (see Section 3.3) with the fine-tuned Phi-2 model, we obtained an accuracy of 81.65% which is a 6.18% increase over the fine-tuned model and 84.65% when we shuffle the options during training. While fine-tuning the Phi-2 model on the training data, we discoveblack that the model's performance stops improving after certain epochs of training. To investigate this we modified the objective (next token pblackiction) to only account for answer generation by masking the part of the cross-entropy loss associated with question/prompt pblackiction.

Using a train-validation split (20% validation) while fine-tuning, it is expected that the validation accuracy increases as the validation loss decreases. However, Fig. 2 shows no improvement in the validation accuracy even as the validation loss decreases. We suspected that this decrease in validation loss results from the model getting better at pblackicting the question and not the answers. Fig. 3 shows the result obtained when the loss associated with questions is masked out, allowing the training objective to focus only on the answers. Fig. 3 shows that the validation accuracy increases as the validation loss decreases. We, therefore, hypothesize that focusing solely on the answers during fine-tuning may yield better results. Validating this hypothesis is an interesting area of future work to be exploblack with additional experiments.

Table 1 – Performance of Phi-2 model on the test dataset as obtained from submissions to Zindi

Method	**Test Accuracy (%)**
GPT-3.5	56.97**
GPT-4	64.78**
Phi-2	42.07
Phi-2 + RAG	66.39 (Ours)
Phi-2 + RAG + Fine-tuning	76.90
Phi-2 + RAG + Fine-tuning + Inference batch-shuffle	81.65
Phi-2 + RAG + Fine-tuning + Inference & Train batch-shuffle (**QMOS**)	**84.65**

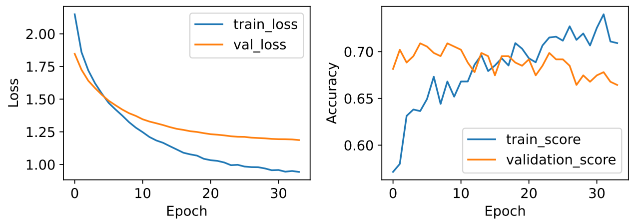

Figure 2 – Loss and accuracy scores when fine-tuning the Phi-2 model with the standard next-token pblackiction objective

In Table 1, ** are results reported from [10] which used part of our test set and other test data.

4.2 Falcon-7B

The performance of the Falcon-7B model is summarized in Table 2.

Table 2 – Performance of Falcon7B model on the test dataset as obtained from submissions to Zindi

Method	**Accuracy (%)**
Baseline Falcon7B	24.51
Falcon7B + RAG	36.61
Falcon7B + RAG + No Options	**49.30**

For the baseline 7B model, when prompted with the options we notice that in some cases the model does

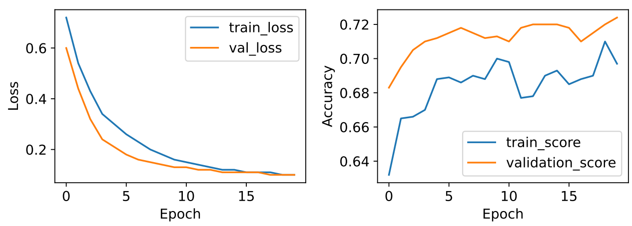

Figure 3 – Loss and accuracy scores when fine-tuning the Phi-2 model and considering only the answers in the next-token pblackiction objective

not output the options but just some unrelated texts.In that case, we randomly choose an option. This yielded an accuracy of 24.51%. Adding the contexts from the RAG further enhanced the score to 36.61%. Finally, by removing the options and allowing the model to generate the answer freely. We then used the embedding of the generated answer with the embedding of the options to select the right option using a cosine similarity metric. With this strategy the model was able to achieve an accuracy of **49.93%**, which is significantly higher than the baseline of **24.51%**. We note that the baseline Falcon-7B model is not fine-tuned considering it has more parameters (7B) than Phi-2 (2.7B). Since fine-tuning will increase the computational cost, we strictly rely on the effectiveness of prompting and RAG systems in a bid to improve its baseline performance on question answering in the telecommunication domain.

5. CONCLUSION

In this research, we have presented a comprehensive approach to addressing Multiple-Choice Questions (MCQs) in the telecommunications domain using small, open-source language models within an enhanced Retrieval-Augmented Generation (RAG) framework. Our study demonstrates that small models, such as Phi-2 and Falcon-7B, when combined with advanced techniques like LoRA fine-tuning, diversified embedding models for RAG, innovative prompt engineering, and batch-shuffle trick can achieve competitive performance compablack to larger, proprietary models like GPT-3.5, while significantly blackucing computational costs. Future work will involve fine-tuning the embedding models (used for RAG) for the telecommunication domain and also further investigate the performance of the proposed QMOS framework on the other language models and MCQs datasets.

6. BIBLIOGRAPHY AND CITATIONS

REFERENCES

[1] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. "Language models are few-shot learners". In: *Advances in Neural Information Processing Systems*. Vol. 33. 2020, pp. 1877–1901.

[2] Adam Maatouk, Fares Ayed, Nicolas Piovesan, Antonio De Domenico, Merouane Debbah, and Zhi-Quan Luo. "TeleQnA: A Benchmark Dataset to Assess Large Language Models Telecommunications Knowledge". In: *arXiv preprint arXiv:2308.06013* (2024).

[3] Andrei-Laurentiu Bornea, Fadhel Ayed, Antonio De Domenico, Nicola Piovesan, and Ali Maatouk. "Telco-RAG: Navigating the challenges of retrieval-augmented language models for telecommunications". In: *arXiv preprint arXiv:2404.15939* (2024).

[4] David Patterson, Joseph Gonzalez, Quoc Le, Percy Liang, Vinod Madhavan, Jeff Dean, et al. "Carbon Emissions and Large Neural Network Training". In: *arXiv preprint arXiv:2104.10350* (2021).

[5] Pouya Pezeshkpour and Estevam Hruschka. "Large language models sensitivity to the order of options in multiple-choice questions". In: *arXiv preprint arXiv:2308.11483* (2023).

[6] Microsoft Research. *Phi-2: The Surprising Power of Small Language Models*. 2023. URL: https://www.microsoft.com/en-us/research/blog/phi-2-the-surprising-power-of-small-language-models/.

[7] Ebtesam Almazrouei, Hamza Alobeidli, Abdulaziz Alshamsi, Alessandro Cappelli, Ruxandra Cojocaru, Mérouane Debbah, Étienne Goffinet, Daniel Hesslow, Julien Launay, Quentin Malartic, Daniele Mazzotta, Badreddine Noune, Baptiste Pannier, and Guilherme Penedo. *The Falcon Series of Open Language Models*. 2023. arXiv: 2311.16867 [cs.CL]. URL: https://arxiv.org/abs/2311.16867.

[8] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. *TinyLlama: An Open-Source Small Language Model*. 2024. arXiv: 2401.02385 [cs.CL]. URL: https://arxiv.org/abs/2401.02385.

[9] Shengding Hu, Yuge Tu, Xu Han, Chaoqun He, Ganqu Cui, Xiang Long, Zhi Zheng, Yewei Fang, Yuxiang Huang, Weilin Zhao, Xinrong Zhang, Zheng Leng Thai, Kaihuo Zhang, Chongyi Wang, Yuan Yao, Chenyang Zhao, Jie Zhou, Jie Cai, Zhongwu Zhai, Ning Ding, Chao Jia, Guoyang Zeng, Dahai Li, Zhiyuan Liu, and Maosong Sun. *MiniCPM: Unveiling the Potential of Small Language Models with Scalable Training Strategies*. 2024. arXiv: 2404.06395 [cs.CL]. URL: https://arxiv.org/abs/2404.06395.

[10] Nicola Piovesan, Antonio De Domenico, and Fadhel Ayed. *Telecom Language Models: Must They Be Large?* 2024. arXiv: 2403.04666 [cs.CL]. URL: https://arxiv.org/abs/2403.04666.

[11] Tasnim Ahmed, Nicola Piovesan, Antonio De Domenico, and Salimur Choudhury. *Linguistic Intelligence in Large Language Models for Telecommunications*. 2024. arXiv: 2402.15818 [cs.CL]. URL: https://arxiv.org/abs/2402.15818.

[12] Heydar Soudani, Evangelos Kanoulas, and Faegheh Hasibi. *Fine Tuning vs. Retrieval Augmented Generation for Less Popular Knowledge*. 2024. arXiv: 2403.01432 [cs.CL]. URL: https://arxiv.org/abs/2403.01432.

[13] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks". In: *Advances in Neural Information Processing Systems*. Ed. by H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin. Vol. 33. Curran Associates, Inc., 2020, pp. 9459–9474. URL: https://proceedings.neurips.cc/paper%5C_files/paper/2020/file/6b493230205f%5C%5C780e1bc26945df7481e5-Paper.pdf.

[14] Llamaindex. *Token text splitter*. https://docs.llamaindex.ai/en/stable/api_reference/node_parsers/token_text_splitter/.

[15] LangChain. *Langchain Text Splitters*. `https://python.langchain.com/v0.1/docs/modules/data_connection/document_transformers/`.

[16] Niklas Muennighoff, Nouamane Tazi, Loïc Magne, and Nils Reimers. *MTEB: Massive Text Embedding Benchmark*. 2023. arXiv: 2210.07316 [cs.CL]. URL: https://arxiv.org/abs/2210.07316.

[17] Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. "Towards general text embeddings with multi-stage contrastive learning". In: *arXiv preprint arXiv:2308.03281* (2023).

[18] Nils Reimers and Iryna Gurevych. "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks". In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Nov. 2019. URL: https://arxiv.org/abs/1908.10084.

[19] Kashvi Taunk, Sanjukta De, Srishti Verma, and Aleena Swetapadma. “A Brief Review of Nearest Neighbor Algorithm for Learning and Classification”. In: *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*. 2019, pp. 1255–1260. DOI: 10.1109/ICCS45141.2019.9065747.

[20] Stephen Robertson and Hugo Zaragoza. “The Probabilistic Relevance Framework: BM25 and Beyond”. In: *Found. Trends Inf. Retr.* 3.4 (Apr. 2009), pp. 333–389. ISSN: 1554-0669. DOI: 10.1561/1500000019. URL: https://doi.org/10.1561/1500000019.

[21] Ggaliwango Marvin, Nakayiza Hellen, Daudi Jjingo, and Joyce Nakatumba-Nabende. “Prompt engineering in large language models”. In: *International Conference on Data Intelligence and Cognitive Informatics*. Springer. 2023, pp. 387–402.

[22] Shubham Vatsal and Harsh Dubey. *A Survey of Prompt Engineering Methods in Large Language Models for Different NLP Tasks*. 2024. arXiv: 2407.12994 [cs.CL]. URL: https://arxiv.org/abs/2407.12994.

[23] 3GPP-TR-21.905. “3rd Generation Partnership Project; Technical Specification Group Services and System Aspects; Vocabulary for 3GPP Specifications (Release 17)”. In: *3GPP Release 17* (2024).

[24] Nils Reimers and Iryna Gurevych. “Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks”. In: *CoRR* abs/1908.10084 (2019). arXiv: 1908.10084. URL: http://arxiv.org/abs/1908.10084.

[25] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. “Lora: Low-rank adaptation of large language models”. In: *arXiv preprint arXiv:2106.09685* (2021).

[26] Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. “Qlora: Efficient finetuning of quantized llms”. In: *Advances in Neural Information Processing Systems* 36 (2024).

[27] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. “Large language models are not robust multiple choice selectors”. In: *The Twelfth International Conference on Learning Representations*. 2023.

[28] Zindi Africa. *Specializing Large Language Models for Telecom Networks*. https://zindi.africa/competitions/specializing-large-language-models-for-telecom-networks/data. 2024.

AUTHORS

Blessed Guda is a dynamic computer engineer with a profound passion for intelligent systems and AI-driven innovations. He holds a Master of Engineering in artificial intelligence from Carnegie Mellon University, where he graduated with distinction, and a first-class honours - Bachelor of Computer Engineering from the Federal University of Technology, Minna. As a passionate researcher and problem-solver, Blessed has consistently demonstrated his ability to mentor and guide students to success, leading teams to winning positions in several high-profile International Telecommunication Union (ITU) competitions, including TinyML, AI in 5G and autonomous networks.

Gabrial Zencha Ashungafac is a research associate at Carnegie Mellon University Africa and a member of the Vision and Language Intelligence Lab, working under Prof. Moise Busogi and Assane Gueye. His research focuses on computer vision, speech technologies, and bias quantification and mitigation in large language models, with particular emphasis on robustness, fairness, and generalization. He holds a B.Sc. in Computer Engineering from TED University (Ankara, Turkey) and an M.Sc. in Engineering Artificial Intelligence from Carnegie Mellon University Africa.

Lawrence Francis is currently a research associate at Carnegie Mellon University Africa. He holds an MSc in Electrical and Computer Engineering from Carnegie Mellon University and a BSc in Electrical and Electronics Engineering from University of Lagos . His research focuses on efficient deep learning computing, with a particular emphasis on improving the accessibility and deployment of deep learning models in resource-constrained environments. He also has practical experience developing AI-poweblack products driven by reinforcement learning.

Carlee Joe-Wong is the Robert E. Doherty Associate Professor of Electrical and Computer Engineering at Carnegie Mellon University. She received her A.B. degree (magna cum laude) in Mathematics, and M.A. and Ph.D. degrees in Applied and Computational Mathematics, from Princeton University in 2011, 2013, and 2016, respectively. Her research interests lie in optimizing various types of networked systems, in particular applications of machine learning and economics to computing and communication networks. From 2013 to 2014, Carlee was the Director of Advanced Research at DataMi, a startup she co-founded from her research on mobile data pricing. Her work has received best paper and poster awards at several conferences, and she received the NSF CAREER award in 2018, the Army Young Investigator award in 2019, and the Department of Energy Early Career Research Program Award in 2024.