Optimizing Retrieval-Augmented Generation for Small Language Models via Output Alignment

Runkai Dong[1*]

[1]School of Computer Science, University of Birmingham, Birmingham, United Kingdom

Abstract. Traditional Visual Question Answering (VQA) models have a significant limitation: they cannot remember the latest external world knowledge, nor use it. Although Retrieval-Augmented Generation (RAG) can solve the problems of hallucination, current researches are focuses on the massive, computation-heavy models, which prevents us from knowing whether RAG is effective in resource constrained environment. This paper presents a comprehensive study of building a light multi modal RAG (MM-RAG) pipeline on consumer-grade hardware. Specifically, this study made a comparative study between two small language models: TinyLlama (1.1B) and Qwen 2.5(3B). The results of research demonstrate that, first, with the access of RAG, the accuracy of outcome results has a significant increase (of about 13%-16%) compared to zero-shot baselines, despite the scale of models. Second, which is the most important, this study identifies a verbosity failure of those instruction-tuned small language models (SLMs), which are more likely to generate output noise and leads to low evaluation marks. To address this, this research developed a post processing protocol, recovering the accuracy of Qwen from 8% to 52.6%. These findings illustrate that for edge-deployed VQA systems, rigorous output alignment is as critical as the retrieval mechanism itself.

1 Introduction

Visual Question Answering (VQA) systems have a main duty: they need to generate answers based on images. But in real life, many questions cannot be answered only given the images, such as figuring out a specific historical person or a rare object. This needs external knowledge. Although Large Language Models (LLMs) are good at this, they still have problems, especially hallucination and lacking up-to-date new world knowledge [1]. To solve these problems, Retrieval-Augmented Generation (RAG) was created. It connects external databases to the model, bridging the gap between what people see and read.

Recently, this area shows a significant development. Lewis et al. built the first RAG framework for NLP [1]. They showed that using outside memory is better than just using a big frozen model. After that, Fan et al. and Zhao et al. wrote surveys about how RAG is moving towards multimodal tasks [2, 3]. Big models like REVEAL and LLaVA got great results using visual instruction tuning [4, 5]. But these models have a big problem: they need massive computational resources and expensive GPUs to run. This makes it very hard for

* Corresponding author's email: dongrunkai@gmail.com

people with normal computers or edge devices to use them. While some works like MobileVLM tried to fix this for phones, the paper still don't know enough about how Small Language Models (SLMs) work with RAG [6].

This paper tries to fill this gap. This study built a light system on consumer-grade hardware (T5 GPUs). The paper followed the baseline method from Xenos et al. [7], but the paper made a big change in model selection. The paper chose two specific models: TinyLlama (1.1B) [8], which is ultra-lightweight for edge devices, and Qwen 2.5 (3B) [9], which represents the "sweet spot" where models start to become smart.

This study has some interesting findings. First, this study found that although small models are not good at reasoning, they can have significant improvement with the access of RAG. Additionally, just like Zheng et al. found, the paper noticed a "verbosity bias" (talking too much) in these small models [10]. The paper proposed a cleaning method to fix this, showing that alignment is very important for getting the right score.

2 Related Work

The foundation of this research lies in the intersection of VQA and RAG [11, 12]. Traditional VQA approaches store all the knowledge within their model parameters. This is useful for closed-domain tasks, but it fails when facing some open tasks which require broad external knowledge.

To address this, Lewis et al. introduced RAG in the NLP domain, proposing a hybrid model that retrieves relevant documents to condition the generation process [1]. Afterwards this method was used to solve multimodal tasks. Early multimodal RAG systems used complex structures. More recent approaches, like REVEAL and LLaVA, employ end-to-end visual instruction tuning [4, 5]. While they are smart, they require massive computational resource for training and reasoning, which is not suitable in this study's edge deployment scenarios.

At the same time, people started to show an interest towards effective small models. Works such as MobileVLM prove it is feasible to run visual language models (VLMs) on mobile devices [6]. But, applying RAG to these lightweight architectures still remains under-explored. To fill this gap, this study evaluated how modular RAG pipeline works with models under 3B parameters, such as TinyLlama and Qwen, and focus on the challenges of output alignment and instruction following in smaller architectures [8, 9].

3 Methodology

3.1 Pipeline Architecture

The MM-RAG architecture presented in this study shows a three-stage pipeline, the difference between this pipeline and some monolithic end-to-end models is that this system adopts a modular design. (see Fig. 1)

First, this system uses the BLIP model, converting images into text description. This step is crucial because it is important to convert pixels into text that can be searched from database. Second, this system uses cosine similarity to find the 9 most related Question-Answer pairs in the knowledge base. Third, this system inputs the retrieved results in the prompt and allows the Large Language Model (LLM) do the final synthesis.

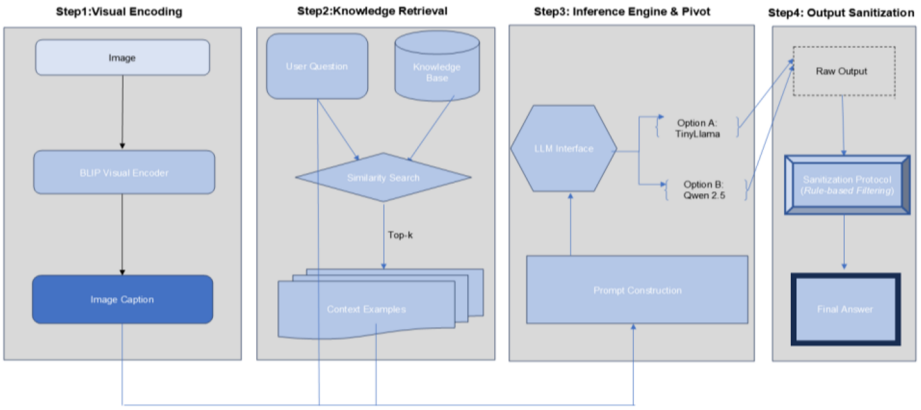

Fig. 1. The schematic overview of the proposed lightweight MM-RAG pipeline. (Picture credit: Original)

3.2 The Strategic Pivot

Table 1. Comparison of language models evaluated

Model	Parameters	Role in Study	Hardware Status
Llama-2	13B	Ideal reproduction model	Access Gated
Mistral	7B	Initial Pivot	OOM (Failed)
Qwen 2.5	3B	Main Subject	Functional
TinyLlama	1.1B	Main Subject	Functional

This study faced a big problem with the use of language models. Initially, this study wanted to use Llama-2, but it is a gated model and all access was rejected by community keepers. Then this study had to change the plan. This study rewrote the code to use the Automodel For CausalLM interface instead of Llama For CausalLM [7]. This was a strategic pivot. This change allowed us to easily switch between different models. But this study still had a hardware constraint, as the Mistral 7B model caused "Out-Of-Memory" (OOM) errors because it was too big for the T5 GPU, so this study chose two smaller models: TinyLlama (1.1B) and Qwen 2.5 (3B) (see Table 1). The pivot turned the hardware limitation into a chance to compare different lightweight models.

3.3 Output Sanitization

When this study tested Qwen, a huge problem was found: The model was too "chatty". Although it could output right answers like "skating", it added some useless words to the initial answer, like "skating\n\Context", which leads to marking failure and low output accuracy. To solve this problem, this study developed a rule-based sanitization protocol. It cuts off any text after a newline character and removes redundant words. This step was crucial. It separated the real reasoning from the "noise", allowing us to see the model's true performance. As a result, this study observed a significant improvement of accuracy, from 8% to 52.6%

4 Experiments and results

4.1 Experimental setup

This study used the OK-VQA validation set (1000 samples) for our tests, and designed a 2x2 experiment matrix: this study compared two models (TinyLlama vs Qwen) in two situations (no RAG vs with RAG). This study only used Exact Match Accuracy to measure success, after applying our cleaning script.

4.2 Quantitative Analysis

The experimental data shows that RAG is crucial for small language models. With access to RAG, the accuracy of two models shows a significant increase between 13% and 16%. Model scale also matters, when this study tested the baseline, Qwen (3B) outperformed TinyLlama (1.1B) by 0.8%, which was a small margin. But with the access of RAG, the increase of Qwen(+15.2%) was much higher than the increase of TinyLlama(+13.5%) (see Fig. 2). This shows that Qwen has a better "brain" to use the outside help (context) effectively.

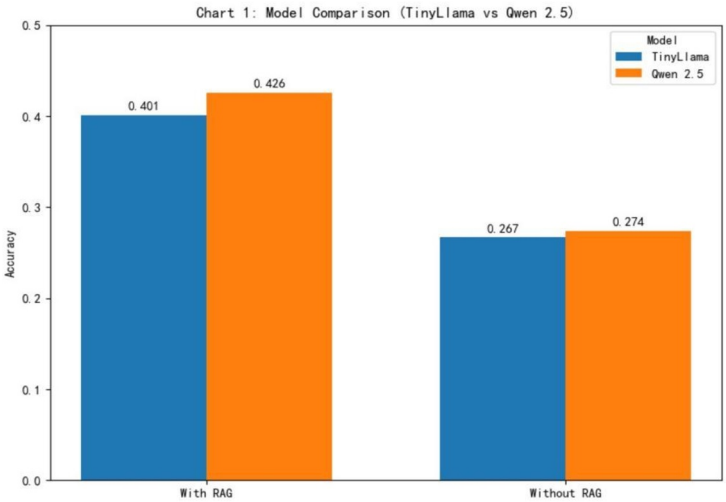

Fig. 2. Performance comparison (Exact Match Accuracy) across model scales (Picture credit: Original)

4.3 Ablation Study

To prove that this study's cleaning script was necessary, this study conducted a test. Without the script, Qwen got a terrible score of only 8%. But when this study used the script, the score jumped to 52.6%. This huge difference tells us something important: the model was not stupid, but it just had a format problem. The model tried too hard to follow the chat format and disrupted the answer structure. So, for chat-based models, alignment of the output format is even more important than the reasoning power itself.

4.4 Qualitative Analysis

The study also looked at specific examples to see why models failed or succeeded. For visual ability, TinyLlama often makes simple mistakes. For example, it called a "gazebo" a "garage" because it couldn't see well. On the other hand, Qwen was smarter. In a picture of a person in snow, Qwen correctly guessed "alpine skiing" because it understood the tools (see Fig. 3(a)). However, the paper also saw a "negative RAG" problem. Sometimes, the LLM could give a wrong answer after given the access of RAG. In one case, the model saw a "napkin"

correctly. But the retrieval system brought back text about “forks” and “tableware”. The model got confused and trusted the text more than its eyes, giving the wrong answer “fork”. This shows small models can be easily tricked by bad retrieval results.

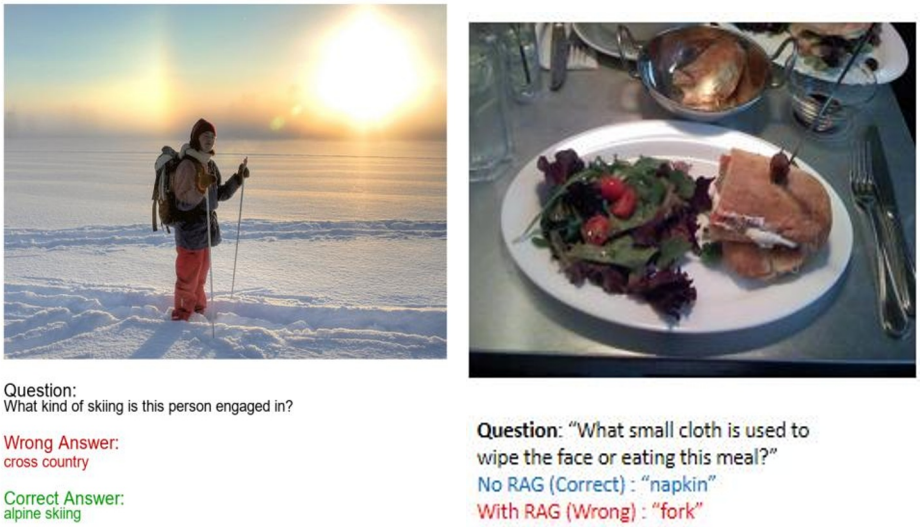

Fig. 3. Qualitative examples of VQA predictions. (a) Qwen successfully reasons about "alpine skiing". (b) An instance of "Negative RAG" where noisy retrieval ("tableware") misleads the model (Picture credit: Original)

5 Conclusion

This research successfully validates that implementing MultiModal Retrieval-Augmented Generation on consumer-grade hardware is accessible. Previously, it was believed that this study must use large models, but through the pivot of changing monolithic model to modular small models (1-3B), this study proves that small model combined with RAG can also achieve significant performance improvements. Specifically, for the Qwen 2.5(3B) model, when combined with the sanitization protocol developed in this study, it reached an accuracy of 52.6%, which is competitive among small models.

This paper also found some challenges. First, small models are more likely to have verbosity bias. This means that good post-processing scripts are always needed; also, the “Exact Match “evaluation metric is too rigid for chat models. Second, through the “Negative RAG” phenomenon, this study shows that it is necessary to find ways to filter out the retrieved noise in future work.

For future work, this study aims to develop a “LLM-as-a-Judge” framework to use Large Language Models as judges. Unlike the rigid exact match system, this approach is more flexible for format changes. Also, 7B parameter model would be implemented to see if the results would be better if access to better GPUs is available in the future.

References

1. P. Lewis et al., Retrieval-augmented generation for knowledge-intensive NLP tasks, in Proc. NeurIPS (2020)
2. W. Fan et al., A survey on RAG meeting LLMs: Towards retrieval-augmented large language models, arXiv preprint arXiv:2405.06211 (2024)

3. P. Zhao et al., Retrieval-augmented generation for AI-generated content: A survey, arXiv preprint arXiv:2402.19473 (2024)
4. Z. Hu et al., REVEAL: Retrieval-augmented visual-language pre-training with multi-source multimodal knowledge memory, in Proc. CVPR, 23369–23379 (2023)
5. H. Liu et al., Visual instruction tuning, in Proc. NeurIPS (2023)
6. X. Chu et al., MobileVLM: A fast, strong and open vision language assistant for mobile devices, arXiv preprint arXiv:2312.16886 (2023)
7. A. Xenos et al., A simple baseline for knowledge-based visual question answering, in Proc. EMNLP (2023)
8. P. Zhang et al., "TinyLlama: An open-source small language model," arXiv:2401.02385, (2024)
9. Qwen Team, "Qwen2.5: A comprehensive series of large language models," arXiv:2409.12191, (2024)
10. L. Zheng et al., Judging LLM-as-a-judge with MT-bench and chatbot arena, in Proc. NeurIPS (2023)
11. S. Maekawa et al., Retrieval helps or hurts? A deeper dive into the efficacy of retrieval augmentation to language models, arXiv preprint arXiv:2402.13492 (2024)
12. A. Asai et al., Self-RAG: Learning to retrieve, generate, and critique through self-reflection, in Proc. ICLR (2024)