Comparative Analysis of RAG vs. Fine-Tuning for Unanswerable Question Detection

Victoria Do and Alejandro Franza and Eric Tsang
{victoriavdo,afranza,ectsang}@berkeley.edu

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Cras ullamcorper elementum metus, eget dignissim felis posuere vitae. Nullam bibendum tellus sed bibendum gravida. Fusce posuere laoreet tortor, at facilisis magna imperdiet vel. Donec finibus, ipsum a finibus ornare, leo risus dictum felis, sit amet porta diam diam sit amet lectus. Aenean aliquam condimentum velit. Quisque sodales volutpat ligula, non gravida metus tincidunt in. Sed fermentum eros ac libero gravida, eu hendrerit enim porttitor. Fusce vehicula dignissim vehicula. Aenean tincidunt posuere lobortis. Nam sodales nunc id ipsum ullamcorper elementum. Nam molestie velit quis elit blandit placerat.

1 Introduction

Question answering (QA) represents a primary use of large language models (LLMs), especially in domains that require accurate information. However, QA systems face a fundamental challenge: they often hallucinate answers to unanswerable questions instead of abstaining (Ji et al., 2023).

This challenge becomes critical in high-stakes domains. Healthcare systems for clinical decision support must never fabricate medical advice, as hallucinations can have life-threatening consequences (Pal et al., 2023). Legal research systems that invent non-existence cases have resulted in attorney sanctions (Dahl et al., 2024). These domains particularly need robust unanswerable question detection because their knowledge bases have clear boundaries - medical guidelines cannot address every rare disease combination; legal databases cannot cover every novel situation.

Two paradigms address hallucination through different knowledge integration approaches. Retrieval-Augmented Generation (RAG) retrieves relevant passages before generation, providing external, non-parametric knowledge access (Lewis et al., 2020). Fine-tuning encodes knowledge directly into model parameters through training on domain-specific data, representing internal, parametric integration (Roberts et al., 2020).

Organizations deploying QA systems must choose between these paradigms yet lack empirical guidance on which better handles unanswerable questions. We address: *Do models more reliably abstain when knowledge is provided externally (RAG) or encoded internally (fine-tuning)? How does each paradigm balance answering correctly while recognizing knowledge boundaries?*

Gap in existing work. While prior work compares RAG and fine-tuning on standard QA metrics (Soudani et al., 2024), this focuses on accuracy when answers exist, not on abstention behavior. Hallucination detection work addresses fabrications after generation (Sadat et al., 2023; Farquhar et al., 2024), rather than comparing prevention through different knowledge integration approaches. SQuAD 2.0 studies concentrate on architectural improvements to extractive models (Rajpurkar et al., 2018), not paradigm comparisons for generative LLMs.

Our contribution. We present a systematic comparison of RAG versus fine-tuning focused on unanswerable question detection. Using SQuAD 2.0, we evaluate three systems: (1) zero-shot baseline with no retrieval or fine-tuning, (2) RAG system with dense retrieval from the SQuAD 2.0 corpus, and (3) Llama fine-tuned on SQuAD 2.0 using QLoRA. Our experimental design isolates the effect of knowledge integration paradigm on abstention behavior while maintaining comparable model capacity and training data. Beyond standard metrics (EM, F1), we measure answer grounding using BERTScore (Zhang et al., 2020) and Semantic Textual Similarity (STS) (Cer et al., 2017) to assess whether answers derive from provided/retrieved context versus parametric memory.

Results preview. Our evaluation reveals distinct tradeoffs between RAG and fine-tuning for

unanswerable question detection...

2 Background

SQuAD datasets. Question answering systems have evolved significantly with large-scale benchmarks. Rajpurkar et al. (2016) introduced SQuAD 1.0 with 100,000+ answerable questions from Wikipedia. However, SQuAD 1.0 had a critical limitation: every question was guaranteed to be answerable from the given context, meaning systems never needed to recognize when they lacked sufficient information. To address this, Rajpurkar et al. (2018) released SQuAD 2.0, adding 50,000 unanswerable questions. This established evaluating both answer accuracy and the ability to recognize knowledge boundaries - the capability we use to assess how different knowledge integration methods handle unanswerable questions.

LLM hallucination. The tendency of LLMs to hallucinate—generate plausible yet unfactual content—is a fundamental challenge in deployment (Ji et al., 2023). While post-generation detection methods exist (Farquhar et al., 2024; Sadat et al., 2023), our work examines prevention through different knowledge integration paradigms—specifically comparing how RAG versus fine-tuning affect hallucination rates when questions are unanswerable.

Retrieval-Augmented Generation. Lewis et al. (2020) introduced RAG, which retrieves relevant documents using dense passage retrieval, then conditions generation on query and retrieved passages. This provides non-parametric knowledge access—information stored externally and provided at inference rather than encoded in weights. Shuster et al. (2021) found RAG reduces fabricated information in dialogue, suggesting external knowledge helps models distinguish accessible versus lacking information.

Fine-tuning. Fine-tuning represents an alternative paradigm where domain knowledge is encoded directly into model parameters through continued training on task-specific data. Roberts et al. (2020) demonstrated fine-tuning effectively injects factual knowledge into parameters for closed-book QA. This parametric approach internalizes knowledge within the model weights themselves.

PEFT. Parameter-efficient methods like LoRA (Hu et al., 2022) and QLoRA (Dettmers et al., 2023) made this practical for large models by fine-tuning only small matrices while freezing pretrained weights. These enable training on domain data including unanswerable examples, potentially teaching boundary recognition. We adopt QLoRA in our fine-tuning experiments to ensure computational feasibility while maintaining model quality.

RAG-FT comparisons. Soudani et al. (2024) compared paradigms across knowledge-intensive tasks, finding strengths depend on task structure and that RAG excels with less popular knowledge while fine-tuning benefits from abundant training data. However, their evaluation emphasized answerable question performance using exact match and F1 scores, not abstention on unanswerable questions.

Our positioning. We address a gap in existing comparisons: systematic evaluation of how RAG versus fine-tuning handle unanswerable question detection in generative LLMs. While prior comparisons (Soudani et al., 2024) focus on accuracy when answers exist, we examine abstention behavior using SQuAD 2.0's unanswerable questions. Our controlled experimental setup compares three systems—zero-shot baseline, RAG with dense retrieval, and QLoRA fine-tuning—using the same base model and training data to isolate the effect of the knowledge integration paradigm. Beyond standard metrics (EM, F1), we use BERTScore (Zhang et al., 2020) and STS (Cer et al., 2017) to measure whether generated answers are grounded in provided/retrieved context versus hallucinated from parametric memory, directly quantifying the distinction between external and internal knowledge access.

3 Methods

3.1 Dataset Description

We evaluate all systems using SQuAD 2.0 (Rajpurkar et al., 2018), a widely-adopted benchmark for question answering that extends SQuAD 1.0 (Rajpurkar et al., 2016) by incorporating 50,000 unanswerable questions alongside 100,000+ answerable questions. This combination makes SQuAD 2.0 suitable for evaluating abstention behavior, as systems must determine both when they can answer and when they should decline.

The dataset consists of question-answer pairs derived from Wikipedia articles across 477 distinct topics. Each example includes a context paragraph, a question, and either an extracted answer span or empty array representing that the question is unanswerable. We use the official training split (130,319 examples) for model training and fine-tuning, and

Table 1: SQuAD 2.0 dataset composition by answerability.

Split	Type	Count
Training	Answerable	86,821 (66.6%)
	Unanswerable	43,498 (33.4%)
	Total	*130,319*
Validation	Answerable	5,928 (49.9%)
	Unanswerable	5,945 (50.1%)
	Total	*11,873*

Table 2: SQuAD 2.0 token count statistics using Llama-3.2-3B tokenizer. Mean values are shown with median in parentheses.

Component	Split	Mean (Med.)	Range
Question	Train	12.32 (12)	1–61
	Val	12.37 (12)	4–37
Context	Train	158.61 (146)	26–907
	Val	166.90 (149)	30–780
Answer	Train	4.69 (3)	1–75
	Val	4.40 (3)	1–42

reserve the official validation split (11,873 examples) as our held-out test set.

Data Characteristics. Table 1 shows the dataset composition by answerability. The training set contains 130,319 examples with 66.6% answerable questions, reflecting the original SQuAD distribution augmented with unanswerable examples. In contrast, the validation set is evenly balanced with 50.1% unanswerable questions across 11,873 examples—a distribution that challenges models to recognize knowledge boundaries without relying on dataset bias.

We evaluate our models on the SQuAD 2.0 dataset, which extends the original SQuAD with unanswerable questions. Table 2 presents token count statistics for questions, contexts, and answers. Questions are consistently brief, with a median length of 12 tokens across both splits. Context paragraphs are substantially longer, with median lengths of 146 tokens (training) and 149 tokens (validation). Answer spans for answerable questions are short, with a median of 3 tokens, though the distribution has a long tail extending to 75 tokens in the training set.

3.2 Baseline Model

Our baseline system answers questions from the SQuAD 2.0 validation dataset without retrieval or fine-tuning, using Llama-3.2-3B-Instruct (**?**) in a zero-shot configuration.

Model Configuration. Each validation example is formatted with three-components: (1) a task instruction defining expected behavior, (2) the context paragraph from SQuAD 2.0, and (3) the question. The model must either extract an answer span from the context or output "unanswerable" when the context provides insufficient information. We employ a low-temperature, top-p-constrained decoding strategy (`temperature=0.1`, `top_p=0.9`, `do_sample=True`) to enforce extractive behavior, as the task requires selecting answer spans rather than generating free-form text. We set `max_new_tokens=35` based on the distribution of answer token counts in the training dataset. Generated outputs are parsed by extracting text following the "Answer:" delimiter; responses containing keywords such as "unanswerable" or "cannot answer" are classified as abstentions.

Instruction Design. We evaluate three instruction variants on a stratified sample of 2,000 training examples to identify the optimal prompt formulation. Based on BERTScore F1 performance, we select the instruction variant that was designed to be more "explicit", which emphasizes extractive behavior and provides clear abstention guidelines (see Appendix A for full methodology and selected instruction).

3.3 RAG Model

3.4 Fine-Tuned Model

To enable direct comparison with the baseline, we fine-tune the same Llama-3.2-3B-Instruct model using QLoRA (Dettmers et al., 2023), a parameter-efficient method that enables fine-tuning of quantized models while preserving performance. We load the base model in 4-bit precision using NormalFloat4 (NF4) quantization with double quantization enabled, reducing memory footprint while maintaining model quality. Computation is performed in bfloat16 precision.

We apply low-rank adapters to both attention layers (`q_proj`, `k_proj`, `v_proj`, `o_proj`) and MLP layers (`gate_proj`, `up_proj`, `down_proj`), as the QLoRA paper demonstrates that targeting all linear layers improves adaptation quality compared to attention-only configurations. We use rank $r = 16$ with scaling factor $\alpha = 32$ (following the common heuristic of $\alpha = 2r$), and dropout of 0.05. This configuration yields 24.3M trainable param-

eters (0.75% of the 3.2B total), enabling efficient fine-tuning on a single GPU.

We construct training examples from the SQuAD 2.0 training split by pairing each question-context input with a target: the extracted answer span for answerable questions, or the literal string "unanswerable" for unanswerable questions. This formulation trains the model to both extract answers and recognize knowledge boundaries within a single generation objective. Due to computational constraints, we randomly subsample 8,000 examples from the training split (seed=42) rather than using the full 130K examples. We reserve 5% of the original training data (6,516 examples) for validation during training.

We train for one epoch to match the baseline's zero-shot setting in terms of minimal task exposure. Training uses a batch size of 1 with gradient accumulation over 8 steps (effective batch size of 8), learning rate of 2×10^{-4} with linear decay, and maximum sequence length of 384 tokens. Following the QLoRA paper's recommendation, we mask prompt tokens during loss computation (setting labels to -100), training only on the target completion. This focuses learning on answer generation rather than prompt reconstruction.

We evaluate on a randomly sampled subset of 2,000 examples from the SQuAD 2.0 validation split, balanced to include both answerable and unanswerable questions. The same subset is used to evaluate the baseline model, enabling controlled comparison between the two approaches.

3.5 Evaluation Strategy

4 Results and Discussion

Table 3 presents the performance comparison between the baseline Llama-3.2-3B-Instruct model and the QLoRA fine-tuned variant across all evaluation metrics.[1]

The QLoRA fine-tuned model reveals a fundamental tradeoff in teaching LLMs to recognize knowledge boundaries. Fine-tuning dramatically improved unanswerable question detection—Exact Match rose from 52.01% to 74.53%, with BERTScore F1 increasing from 44.26% to 67.50%—demonstrating that the model successfully learned to abstain when context is insufficient.

[1]Baseline metrics are computed on the full SQuAD 2.0 validation set (11,873 examples), while QLoRA metrics are from a 2,000-example subset. Future work will evaluate both models on identical subsets for stricter comparison.

Table 3: Baseline vs. QLoRA fine-tuned model performance on SQuAD 2.0.

Metric	**Baseline 3B**		**QLoRA 3B**	
	Ans.	Unans.	Ans.	Unans.
Exact Match	38.36%	52.01%	2.99%	74.53%
F1 Score	59.58%	52.01%	18.93%	74.53%
BERTScore F1	46.45%	44.26%	10.08%	67.50%
STS Score	85.83%	78.41%	71.14%	88.38%

However, this capability came at the expense of answer extraction: answerable Exact Match fell from 38.36% to 2.99%, and F1 from 59.58% to 18.93%. The STS scores reinforce this pattern: high semantic similarity on unanswerable questions (88.38%) confirms reliable abstention behavior, while lower answerable STS (71.14%) reflects the model's tendency to abstain even when answers exist. This asymmetry suggests that within our constrained training regime (8,000 randomly sampled examples, single epoch), the model learned abstention as a dominant strategy rather than developing balanced judgment. The finding highlights a key challenge for fine-tuning approaches: optimizing for unanswerable question detection may inadvertently suppress the extractive capabilities that make QA systems useful, a tradeoff that retrieval-augmented approaches may handle differently by externalizing knowledge boundaries.

5 Conclusion

Cras cursus condimentum semper. Donec rhoncus ante ac mi aliquam rutrum. Nullam pulvinar felis vitae tellus porttitor, non vehicula ante consectetur. Fusce accumsan tortor at nunc finibus, non malesuada felis cursus. Nulla malesuada mattis mollis. Donec finibus consequat tincidunt. Quisque metus risus, blandit dictum mollis ut, molestie eget ex. Suspendisse mauris erat, ultrices vitae euismod ac, varius tempor nulla. Vivamus pellentesque nisl eu venenatis aliquam. Donec tincidunt ut lectus nec interdum. Nam eu libero luctus, dapibus odio ac, gravida quam. Quisque blandit quis velit vel consequat.

6 Authors' Contributions

Victoria Do contributed with the data preprocessing, exploratory data analysis (EDA), evaluation functions, and baseline model/results. Victoria was the primary author for the Introduction and Background sections. Victoria was also the primary au-

thor for the Data Description and Baseline Model Implementation subsections of the Methods section.

Alejandro Fernandez contributed...

Eric Tsang contributed...

References

Daniel Cer, Mona Diab, Eneko Agirre, Iñigo Lopez-Gazpio, and Lucia Specia. 2017. SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 1–14, Vancouver, Canada. Association for Computational Linguistics.

Matthew Dahl, Varun Magesh, Mirac Suzgun, and Daniel E Ho. 2024. Large legal fictions: Profiling legal hallucinations in large language models. *Journal of Legal Analysis*, 16(1):64–93.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. In *Advances in Neural Information Processing Systems*, volume 36, pages 10088–10115. Curran Associates, Inc.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Comput. Surv.*, 55(12).

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. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474. Curran Associates, Inc.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2023. Med-HALT: Medical domain hallucination test for large language models. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pages 314–334, Singapore. Association for Computational Linguistics.

Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don't know: Unanswerable questions for SQuAD. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Adam Roberts, Colin Raffel, and Noam Shazeer. 2020. How much knowledge can you pack into the parameters of a language model? In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5418–5426, Online. Association for Computational Linguistics.

Mobashir Sadat, Zhengyu Zhou, Lukas Lange, Jun Araki, Arsalan Gundroo, Bingqing Wang, Rakesh Menon, Md Parvez, and Zhe Feng. 2023. DelucionQA: Detecting hallucinations in domain-specific question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 822–835, Singapore. Association for Computational Linguistics.

Kurt Shuster, Spencer Poff, Moya Chen, Douwe Kiela, and Jason Weston. 2021. Retrieval augmentation reduces hallucination in conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3784–3803, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Heydar Soudani, Evangelos Kanoulas, and Faegheh Hasibi. 2024. Fine tuning vs. retrieval augmented generation for less popular knowledge. In *Proceedings of the 2024 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region*, SIGIR-AP 2024, pages 12–22, New York, NY, USA. Association for Computing Machinery.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. 2020. Bertscore: Evaluating text generation with bert. In *International Conference on Learning Representations*.

A Baseline Prompt

A.1 Test Data Construction

.

We sampled 2,000 examples from the SQuAD 2.0 training set, stratified by:

Answerability: Answerable vs. unanswerable questions (equal representation).

Complexity: Defined as the average of standardized token counts for context, question, and answer. We partition into three levels (low, medium, high)

using quantile binning, resulting in six strata. Proportional sampling ensures balanced representation across all combinations.

A.2 Instruction Variants

V0 (Simple):

> *Answer the question based solely on the given context. If the answer is not in the context, respond with 'unanswerable'.*

V1 (Explicit):

> *You are an extractive question answering system. Extract the shortest possible answer span from the context that fully answers the question. Extract the answer word-for-word from the context. Do not paraphrase or generate new text. Do not use outside knowledge. Do not provide an explanation of your response. Just the answer span. If the question cannot be answered from the context, output exactly: 'unanswerable'.*

V2 (Chain-of-Thought):

> *You are an extractive question answering system. Answer the question based on the context given. Think step by step: 1. First, determine if the context contains information to answer the question. 2. If yes, extract the exact text span that answers it. 3. If no, respond with just 'unanswerable'.*

A.3 Results