

A Passage-Based Question Answering (PBQA) System on Domain-Specific Data, and a Comparative Evaluation with a Retrieval-Augmented Generation (RAG) System Using an LLM-as-a-Judge

Vishal Saxena, Stephanie Owyang

August 3, 2025

Abstract

We present a Passage-Based Question Answering (PBQA) system and a Retrieval-Augmented Generation (RAG) system, both applied to domain-specific spiritual content from ancient Vedic texts. Using a quantized, instruction-tuned LLaMA 8B model fine-tuned with QLoRA, we built an affordable and replicable pipeline that generates, retrieves, and evaluates answers from Vedic discourse. Due to limited compute, we worked with a few thousand examples and evaluated system quality using a combination of human scores, and an LLM-as-a-Judge model. Our LLM-as-a-Judge was prompt-tuned using gold scored QA sets and demonstrated strong correlation with human evaluation. RAG outperformed PBQA in terms of relevance, faithfulness, and completeness. Our pipeline is fully automated and generalizable to other domains such as finance, healthcare and legal.

1. Introduction

LLM-based QA systems often struggle in low-resource domains like ancient Vedic Sanskrit, where key terms (e.g., *Parabrahmam*, *Rasa Aaswaadanam*) lack English equivalents and are poorly represented in modern corpora. This leads to hallucinations and reduced reliability for spiritually inclined users.

To address this, we developed a low-cost, high-fidelity pipeline that processes Vedic texts and generates relevant, grounded answers using QLoRA-based fine-tuning. We also introduce an LLM-as-a-Judge model to evaluate outputs at scale with human alignment.

Research Question: *How does a retrieval-augmented generation (RAG) system compare to a passage-based QA system in answering domain-specific questions, when evaluated using a human-aligned LLM-as-a-Judge?*

We compare three systems: a base LLaMA-3 model (Control), a passage-based QA system using a fine-tuned model (Treatment Group 1), and a RAG system using the same generator with top-3 passage retrieval (Treatment Group 2). All systems were evaluated on identical inputs using Likert-scale scores for relevance, faithfulness, and completeness (see Appendix-F for criteria details). Our approach combines efficient tuning, retrieval, and automated evaluation into a generalizable pipeline.

2. Background

2.1 Prior Work on Domain-Specific RAG: Sharma et al. [1] fine-tuned a RAG system on proprietary Adobe product data. They chunked documents, encoded them with a transformer, and retrieved top-k documents via cosine similarity. Their generator (GPT-4) performed best when coupled with a fine-tuned retriever. Evaluation was both qualitative and quantitative using a 0–1 relevance score. Their architecture mirrors ours in retrieval flow but differs in retriever fine-tuning (we use a pre-trained retriever).

2.2 Evaluation Metrics and Reference-Free Scoring: Oro et al. [2] analyzed both reference-based and reference-free RAG evaluation metrics. They compared BEM, Exact Match, and AC-RAGAS against human judgment and found that reference-based scores had moderate correlation (e.g., BEM at 0.531 with GPT-4-turbo), while AR-RAGAS had weaker correlation (as low as 0.23). This highlights the challenge of high-fidelity automatic evaluation without ground truth — particularly relevant for our setup.

2.3 LLM-as-a-Judge as Evaluator: Ho et al. [3] demonstrated that LLM-as-a-Judge can better replicate human QA evaluation compared to traditional metrics like EM and F1. Their experiments showed Spearman correlations of 0.85 for a large judge model (Qwen 72B), outperforming EM (0.22) and F1 (0.40). Importantly, they found no bias when the same LLM is used both for generating and judging answers. We compare our Spearman correlation scores to Ho et. al.’s work. While an earlier iteration of our full work did use the same LLM for both question generation and answer generation, that iteration is not included in this report. In this report we only report results from an iteration that used one LLM for question generation, and another LLM for generating answers.

2.4 Efficient LLM Fine-Tuning: LoRA and QLoRA enable fine-tuning of large models with dramatically fewer parameters. Hu et al. [4] introduced LoRA by applying low-rank matrices (rank r) to attention and FFN layers, reducing trainable parameters from billions to millions. Dettmers et al. [5] extended this with QLoRA: quantizing weights to 4-bit, freezing them, and updating only adapters. We apply this method to fine-tune LLaMA-8B on limited hardware.

2.5 Approach for Evaluating LLM Outputs in Expert Knowledge Tasks: LLM evaluators often align more closely with lay users than with domain experts, revealing a gap in domain-specific reasoning. For our experiment, we modified our approach to account for the potential for a low level of agreement between a domain expert and the LLM evaluator. Drawing on insights from Szymanski et al. [6], we enhanced our evaluation setup by instructing the LLM to adopt an expert persona and incorporating human feedback.

Appendix-A has additional details on sections 2.1-2.4.

3. Methods

3.1 Overview

We developed and evaluated three QA system variants on a common domain-specific question set. Each system was evaluated on the same questions using three criteria: Relevance, Faithfulness, and Completeness (RFC). Evaluation was performed using (i) a human domain subject matter expert (SME), and (ii) a custom-trained LLM-as-a-Judge. We excluded RAGAS for our analysis since a high-fidelity use of RAGAS requires ground truths, and reference free RAGAS metrics while they exist have low correlations with human judgments [2].

Group	Description
Control Group (CG)	Base LLaMA-8B model with no fine-tuning, given a single passage per question.
Treatment Group 1 (TG-1)	Fine-tuned LLaMA-8B model using QLoRA, same single-passage input per question.
Treatment Group 2 (TG-2)	RAG system using a retriever (BAAI/bge-small-en-v1.5) to select top-3 relevant passages. Uses TG-1 fine-tuned LLM as generator.

Table 1: Our experiment design

We hypothesized that (1) TG-1 would outperform CG due to fine-tuning, and (2) TG-2 would outperform TG-1 due to multi-passag e semantic retrieval.

Each system’s answers were evaluated using a 5-point Likert scale along three axes—Faithfulness, Completeness, and Relevance—as defined in Appendix-F. Faithfulness, which penalizes hallucinations and measures grounding in the passage, was given highest importance, followed by Completeness (coverage of all aspects of the question), and Relevance (directness of the answer). To compute a single aggregate score, we applied a weighted scheme of 60% Faithfulness, 30% Completeness, and 10% Relevance, used consistently for both model evaluation and development of the LLM-as-a-Judge (see Evaluation section for details).

3.2 Data Preparation:

We used two source PDFs: **Geetha Vahini**: Served as the corpus for passage-question-answer triplet generation and fine-tuning; **Jnana Vahini**: Provided inference-only question set for evaluation.

From **Geetha Vahini**, we generated 4,733 QA triplets using Mistral-7B-Instruct v0.2. The passage-to-question generation used a single-question-per-passage strategy to manage memory constraints. Context length was capped at 1000 characters with 200-character overlaps.

The **evaluation set** contained 50 inference-time questions generated from Jnana Vahini. All three system variants produced answers for these same 50 questions. See Appendix-B for additional details on document to question pipeline.

3.3 Implementation framework for Control Group

Following picture shows the workflow for using inference against the base LLM model to generate answers from the Control Group to the same set of questions as fed to the two treatment groups. Appendix-C.1 has the prompt used for the LLM model.

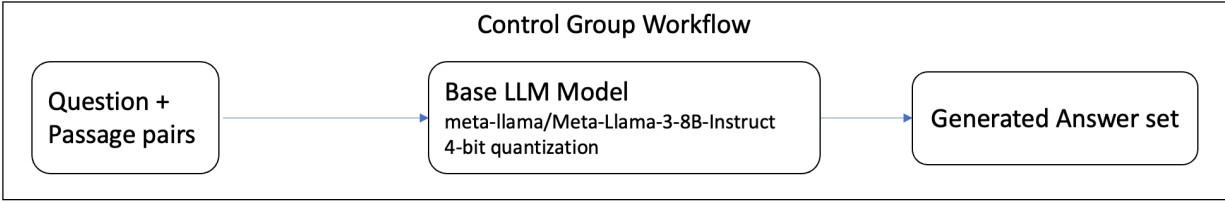


Figure 1: Workflow for the control group

3.4 PBQA framework for Treatment Group 1

This implementation fine-tuned a large LLM for Treatment Group 1 as shown in Figure 2.

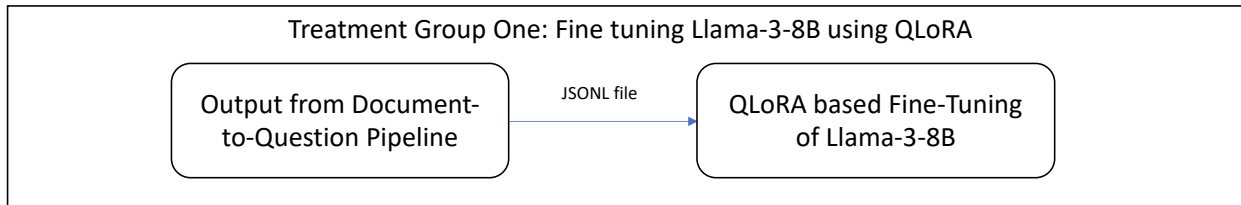


Figure 2: Workflow for fine tuning Llama-3-8B

Appendix-C.2 has details about our design, approach, and the used training hyperparameters. We set the LoRA rank to 16 and dropout to 0.05, and applied low-rank adaptation to both attention sublayers (q_proj, k_proj, v_proj, o_proj) and feed-forward layers (gate_proj, up_proj, down_proj), following architectural insights from models like LLaMA and Mistral. Inspired by the original LoRA paper [4], which focused on attention layers, we extended adaptation to FFN layers; future ablations may assess the impact of excluding them. This reduced trainable parameters from 8.072 billion to 41.943 million (a 99.48% reduction). Training over 3 epochs on a single A100 GPU (≥ 40 GB VRAM) using QLoRA (4-bit quantization) took 81 minutes and reduced loss from 1.244 to 0.126, indicating mild overfitting. Future work could explore: Fewer training epochs; increased corpus size and diversity; lower adapter rank (e.g., 8 or 4); removing FFN layers from adaptation to reduce overfitting.

Post-training, we merged adapters into the base model to minimize inference latency and also retained unmerged adapters for reuse in downstream experiments, including TG-2.

3.5 RAG framework for Treatment Group 2

We used Langchain and Chroma vector store to create an embedding framework for RAG, as shown below:

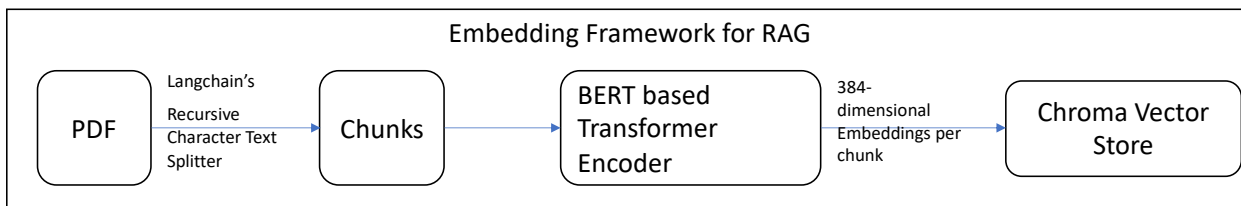


Figure 3: Phase 1 (Ingestion) workflow for RAG used for Treatment Group 2

Appendix-C.3 details the RAG pipeline: Ingestion (Figure 3), and Inference. During ingestion, we split the 146-page training PDF (*Geetha Vahini*) into 508 text chunks using a 1000-character chunk size and 200-character overlap to maintain context. For inference, each question was embedded using the BERT-based encoder BAAI/bge-small-en-v1.5 (same as indexing), and the top 3 semantically similar chunks were retrieved from the vector store via cosine similarity. These chunks were concatenated with the query to form a prompt, which was fed into the fine-tuned LLaMA-3-8B-Instruct model (same generator as Treatment Group 1) to generate a context-aware answer.

4. Evaluation Framework

4.1 Evaluation Criteria

Our used criteria for evaluation were introduced in Section 3.1.

4.2 LLM-as-a-Judge (LJ)

We developed a custom LLM-as-a-Judge using GPT-4o Mini via few-shot prompt tuning. GPT-4o Mini was chosen for its strong performance, cost-efficiency, and low latency [9]. Following Szymanski et al. [6], we used 20 unit-test question-answer pairs, each scored by a human on relevance, faithfulness, and completeness, to calibrate the LLM’s judgment to reference scores.

To evaluate alignment between the LLM and human judgment, we used Spearman’s ρ , with strong alignment defined as $\rho \geq 0.8$ [9]. To improve correlation, we selected high-delta samples for few-shot examples and conducted prompt optimization using ChatGPT. The process involved three refinement iterations, testing various few-shot combinations, scoring strategies, and n-completions with aggregation. See Appendix-D for full implementation details and evaluation results of LLM-as-a-Judge.

5. Results

5.1 Selecting the Best LLM-as-a-Judge

Across multiple iterations, we selected the best-performing judge (v1.1) based on Spearman correlation with human scores on gold QA set containing 20 golden RFC scored QA pairs (see Appendix E). This set was manually created to provide a distribution for Relevance (R), Faithfulness (F), and Completeness (C) scores. Judge version 1.1 showed highest correlation (Table 2) with human scores for R, F, and C and was selected as the best judge for downstream evaluations. Appendix-D has details on how each version was developed.

Spearman correlation with Human Scores (UT-15)			
Judge Version	Relevance	Faithfulness	Completeness
LLM-as-a-judge v0	0.749	0.733	0.835
LLM-as-a-judge v1	0.756	0.733	0.835
LLM-as-judge v1.1	0.759	0.799	0.840
LLM-as-a-judge v2	0.756	0.774	0.835

Table 2: Spearman Correlation Between Human Scores and LLM-as-a-Judge Versions

5.2 Fidelity of LLM-as-a-Judge

To further validate our best judge (v1.1) judge, we compared its scores to human scores on the 46 QA examples used for system evaluation (after removing prompt-tuning examples; Table 3). Spearman correlations were found to be on par with those reported by Oro et al. [2] for reference-free metrics.

Spearman correlation with Human Scores (46 Evaluation Questions for v1.1)						
Experiment Group	Relevance		Faithfulness		Completeness	
	HJ	LJ	HJ	LJ	HJ	LJ
CG: Base Llama	1	0.671	1	-0.182	1	0.581
TG1: PBQA Fine-tuned Llama	1	0.612	1	0.205	1	0.448
TG2: RAG	1	0.391	1	0.360	1	0.438

Table 3: Assessing fidelity of LLM-as-a-Judge (LJ) by comparing its scores on the evaluation QA set with human judgment (HJ) scores for the same set

5.3 System Performance by Judge

We now present the **actual Relevance, Faithfulness, and Completeness scores** assigned to each system variant by both human (HJ) and LLM-as-a-Judge (LJ). Both HJ and LJ scored RAG the highest on relevance; none rated CG highest on Faithfulness (which aligns with our observations of hallucination from CG); the groups that used fine-tuned Llama (TG-1 and TG-2) were rated highest by HJ or LJ; HJ rated RAG highest on completeness and we note that LJ’s completeness score for RAG is close to its highest score.

Actual scores of each evaluation model (46 Evaluation Questions using v1.1)						
Experiment Group	Relevance		Faithfulness		Completeness	
	HJ	LJ	HJ	LJ	HJ	LJ
CG: Base Llama	3.14	3.245	3.34	3.551	2.76	4.143
TG1: PBQA Fine-tuned Llama	3.74	3.531	3.32	4.490	3.56	3.816
TG2: RAG	3.94	3.878	3.96	4.347	3.92	4.122

Table 4: Actual scores of each evaluation model

6. Weighted Scoring and Hypothesis Testing

We applied the weighted RFC formula (0.6 Faithfulness, 0.3 Completeness, 0.1 Relevance) to compute overall scores for each system (Table 5). Both human judgment (HJ) and LLM-as-a-Judge (LJ) ranked **TG-2 (RAG)** highest, reflecting its advantage of using three semantically relevant passages, unlike **TG-1 (PBQA)**, which received only one. Despite using the same fine-tuned LLaMA model for generation, TG-2’s richer context led to superior performance. Results support both hypotheses: (1) TG-1 outperforms CG, confirming the benefit of domain-specific fine-tuning; (2) TG-2 outperforms TG-1, validating the added value of retrieval-augmented input.

Weighting Scheme		Weighted scores of each group		
Criterion	Weight	Experimental Group	HJ	LJ
Relevance	0.1	CG: Base Llama	3.146	3.698
Faithfulness	0.6	TG1: PBQA Fine-tuned Llama	3.434	4.192
Completeness	0.3	TG2: RAG	3.946	4.233

Table 5: Weighting scheme and weighted scores for each experimental group

7. Conclusion

We developed a fully automated QA pipeline for low-resource domain texts using open-source models and efficient training techniques. Our system demonstrated moderate-to-high fidelity on ancient Vedic texts, with strong alignment between LLM-as-a-Judge and human evaluation. The RAG-based system (TG-2) outperformed both PBQA (TG-1) and the baseline (CG), validating our hypotheses. This pipeline—combining PDF ingestion, QA pair generation, fine-tuned LLMs, and LLM-based evaluation—is scalable and generalizable to other domains such as finance and healthcare. These domains may additionally employ named entity removal to provide privacy safeguards for sensitive data [1]. By integrating low-cost tuning (QLoRA) with automated evaluation (LLM-as-a-Judge), our work presents a practical blueprint for domain-specific NLP under compute constraints.

APPENDIX CONTENTS

- REFERENCES
- SALIENT GIT HUB FILES
- DIVISION OF WORK / CONTRIBUTION STATEMENTS
- APPENDIX A: Detailed Literature Review
- APPENDIX B: Document to Question Generation Pipeline
- APPENDIX C.1: Implementation Details for Meta-LLaMA-3-8B-Instruct Control Group
- APPENDIX C.2 Implementation Details for Fine-Tuning Meta-LLaMA-3-8B-Instruct with QLoRA
- APPENDIX C3: Treatment Group 2's RAG Pipeline Details
- APPENDIX D: Building an LLM-as-a-Judge
- APPENDIX E: Unit Test (Gold QA) Set used for Prompt Training LLM-as-a-Judge
- APPENDIX F: Evaluation criteria definitions and their Likert Scales
- APPENDIX G: Base-Llama Negative Correlation Exploration

Note: Figures and tables in the Appendix follow a separate sequential numbering scheme from those in the main body of the report.

REFERENCES

1. Sharma, S., Sultania, D., Yoon, D. S., Bagga, K., Derroncourt, F., Zhang, M., Bui, T., & Kotte, V. (2024, May). Retrieval-augmented generation for domain-specific question answering [Preprint]. arXiv. <https://arxiv.org/abs/2305.06983>
2. Oro, E., Granata, F. M., Lanza, A., Bachir, A., De Grandis, L., & Ruffolo, M. (2023). Evaluating retrieval-augmented generation for question answering with large language models [Preprint]. arXiv. <https://arxiv.org/abs/2307.08301>
3. Ho, X., Boudin, F., Huang, J., & Aizawa, A. (2025, April). LLM-as-a-judge: Reassessing the performance of LLMs in extractive QA [Preprint]. arXiv. <https://arxiv.org/abs/2305.14620>
4. Hu, E., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2022). LoRA: Low-rank adaptation of large language models (Version 2) [Preprint]. arXiv. <https://arxiv.org/abs/2106.09685>
5. Dettmers, T., Pagnoni, A., Zettlemoyer, L., & Lewis, M. (2023). QLoRA: Efficient finetuning of quantized LLMs [Preprint]. arXiv. <https://arxiv.org/abs/2305.14314>
6. Szymanski, A., Ziems, N., Eicher-Miller, H. A., Li, T. J.-J., Jiang, M., & Metoyer, R. A. (2025). Limitations of the LLM-as-a-judge approach for evaluating LLM outputs in expert knowledge tasks. In Proceedings of the 30th International Conference on Intelligent User Interfaces (IUI 2025). ACM. <https://arxiv.org/abs/2410.20266>
7. Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., Wang, S., Zhang, K., Wang, Y., Gao, W., Ni, L., & Guo, J. (2024). A survey on LLM-as-a-judge [Preprint]. arXiv. <https://arxiv.org/abs/2411.15594>

8. Gao, M., Liu, Y., Hu, X., Wan, X., Bragg, J., & Cohan, A. (2024). Re-evaluating automatic LLM system ranking for alignment with human preference [Preprint]. arXiv. <https://arxiv.org/abs/2501.00560>
9. Croxford, E., Gao, Y., First, E., Pellegrino, N., Schnier, M., Caskey, J., Oguss, M., Wills, G., & Patterson, B. W. (2025). Automating evaluation of AI text generation in healthcare with a large language model (LLM-as-a-judge) [Preprint]. medRxiv. <https://doi.org/10.1101/2025.04.22.25326219>

Salient Git Hub FILES TABLE

File	Description
266_FP_VS_VZ1_QExtract_0724_new_largellm.ipynb	Executes PDF to Question Extraction pipeline
266_FP_VS_VZ2_FineTuning_llama_8B.ipynb	Code for Fine Tuning Llama 8B using QLoRA
266_FP_VS_VZ5_RAG_question_answer.ipynb	Code for creating a RAG and executing it for inference
266_FP_VS_VZ6_Unit_Test_OnFineTuned_llama_8B.ipynb	Code for executing Unit Test cases on fine tuned LLama
266_FP_VS_V4_Control_Group_llama_8B.ipynb	Code for executing inference against base LLama
266_FP_SO_QExtract_gpt4.ipynb	GPT-4 question bank generation
266_FP_SO_llama8B_control_group_generate_answers.ipynb	Answer generation of control group. Based on LLaMA8B
266_FP_SO_gpt4o_mini_QA_evaluator_v1.ipynb	LLM evaluator v1 – selected for final results
266_FP_SO_gpt4o_mini_QA_evaluator_v2.ipynb	LLM evaluator v2 - prototyping

DIVISION OF WORK / CONTRIBUTION STATEMENTS

Stephanie – My main focus was the LLM-as-a-judge implementing the evaluator following the best practices in the industry, and iterating on the architecture and design in an attempt to meet the latest benchmarks for LLM evaluators correlation with human evaluators. I also was responsible for generating the answers using Meta-Llama-3-8B-Instruct and question bank generation using GPT-4.

Vishal - Concept, Idea Owner; Solution architect; Ownership of: development (design and coding) of Control and both the Treatment groups; unit test and evaluation cases; human judgment scoring and SME facilitation; solution design inputs to LLM-as-a-Judge work; core authorship of report/overall narrative; collaboration on results development; ownership of research items for control, treatment groups, LoRA, QLoRA, and some items for LLM-as-a-judge e.g. evaluation.

APPENDIX A: Detailed Literature Review

Sharma et. al. fine-tuned a retriever for a domain specific LLM that generated answers to domain questions. The model that used user click data for questions around Adobe products, used an approach similar to ours: it converted documents to chunks; vectorized the chunks using an LM encoder; stored embeddings in a vector database for later retrieval; and when given a query, grabbed its embedding using the same LM encoder (BERT), and passed top ranked documents to a fine-tuned LLM, based on cosine similarity search, that then produced the final answer. They learned query and document embeddings using contrastive learning (similar samples modeled closely in the representation space) and by sharing the same weights between the two encoding instances (query, and document) of LM transformer instance. Prior to vector storage, they applied Named Entity Recognition (NER) to filter out obnoxious or non-relevant textual patterns (e.g. email formats or signature lines). They evaluated the retriever performance using nDCG scores, and found their fine-tuned retriever to perform superior to not-fine-tuned models. They evaluated generated answers using a Relevance score (0-1) both quantitatively and qualitatively, and found the performance of a fine-tuned retriever + GPT4 (for generator) to be the most superior. In contrast, for our RAG, we use a fine-tuned LLM model for generation, but do not fine tune the retriever.

Oro et. al. assessed evaluation metrics for RAG outputs. They evaluated ground-truth based metrics as well as reference free metrics for assessing the quality of RAG outputs and by examining the spearman correlation between metrics' score vis-a-vis human judgments. They found that reference free metrics (more relevant for our case) still struggle to capture nuances in answer quality without predefined correct responses. For answer generation evaluation, they used Exact Match (EM), F1 for syntactic metrics (formal response aspects), and semantic metrics e.g. BEM that better capture meaning of the response. For ground truth free evaluation (more relevant for our case) they used RAGAS. They found that ground truth-based metrics like BEM and AC RAGAS showed moderate to strong correlation with human judgment (e.g. 0.531 in the case of GPT 4 turbo) while their reference free counterparts such as AR RAGAS struggled to correlate well with human judgments. Their reference free evaluation AR RAGAS' correlation score with human judgment was 0.23 and 0.287 for Financial and all datasets (that additionally included Adobe product user information) respectively (Tables 4-5).

Ho et. Al. evaluated the performance of LLM-as-a-judge used as evaluators for answers from extractive QA. They found that using LLM-as-a-judge correlates better with human judgments as compared to traditional metrics such as Exact Match (EM) and F1 scores. Specifically, for correlation with human judgment they found the respective correlation scores to be 0.22 for EM, 0.40 for F1-score, and 0.85 for LLM-as-a-judge. In addition, and perhaps more interestingly, they found no bias issues when the same model was used for both QA extraction and the judgment tasks. They also found that generally EM and F1-scores underestimate the model's performance. All models in their experiments were instruction tuned (similar to our work). To establish a judge-model they prompt tuned a large LLM using a few-shot prompting approach (similar to our approach). During evaluation they present the model with a question, a gold answer, a predicted answer, matching context and instructed the model to predict one of two labels: Correct and Incorrect. As perhaps expected, larger models (e.g. Qwen 2.5 72B) show a higher correlation with human judgments relative to smaller large models (e.g. Mistral 7B) — the mean person correlation coefficient scores between human judgments and LLM-as-a-judge evaluations were 0.847 and 0.653 for the Qwen 2.5 72B and Mistral 7B as LLM-as-a-judge respectively. They also found that in a few instances LLM-as-a-judge is less strict when judging the correctness of the predicted answer.

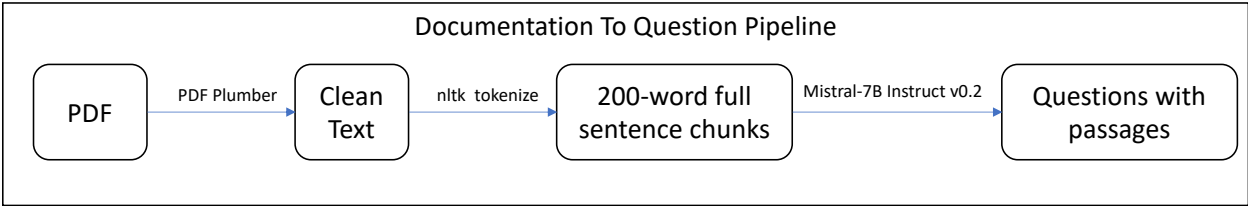
Hu et al. (2021) propose a method for Low-Rank Adaptation (LoRA) of large language models. Specifically, when LoRA of rank r is applied to a transformer layer—such as the query, key, or value projection in self-attention—and: k is the size of the hidden vector entering the layer, and, d is the size of the output vector from the layer, then, instead of updating the full weight matrix with $d \times k$ trainable parameters (as in traditional fine-tuning), LoRA introduces a low-rank decomposition of the weight update: $\Delta W = B \times A$, where B is of dimensions $(d \times r)$ and A is of dimensions $(r \times k)$.

This reduces the number of trainable parameters to $r \times (d + k)$, significantly lowering compute and memory requirements while maintaining strong downstream task performance. Further, the learned adapter weights can be merged with main weights to reduce latency in inference (an approach we also use). In addition, use of really low rank such as 2, on all of W_q, W_k, W_v, W_v of attention layer produces best performance (an approach also used by us).

Dettmers et. al. (2023) introduced a novel way to fine tune large LLMs on reduced GPU memory without degrading fine tuning performance. In the approach called QLoRA: they quantize pre-trained model weights to 4-bit, add a small set of low-rank adapter weights, then train these low-rank adapter weights by back propagating gradients through the quantized weights. The quantized weights remain frozen, and only the LoRA adapters are updated. We use QLoRA to fine tune Llama 8B for this work.

APPENDIX B: Document to Question Generation Pipeline

1. Data processing: Document to question pipeline



We used this pipeline two times. First, on a pdf file called Geetha Vahini.pdf, that served as the main training corpus for our work. This deployment produced 4,733 triplets: of passage text, question extracted from the passage, and a matching answer. This generation was from Mistral-7B-Instruct v0.2, a large LLM that is instruction-fine-tuned and designed to follow prompt. The prompt we used for generating this triplet is provided below. We human spot checked the generated triplets for relevance and fidelity using random samples. We changed our prompt to produce only one question from the passage versus three, to manage throughput on our memory constrained compute. We set maximum new tokens to 128, and did not find the need to reduce this since our questions weren’t consistently short, use sampling to manage diversity in generated questions, used 4-bit quantization and a batch size of eight for efficiency.

A second deployment of this pipeline was for inference questions, which were generated from a separate PDF file called Jnana Vahini. These inference questions were used to test performance of control and all our treatment groups.

File: 266_FP_VS_VZ1_QExtract_0724_new_largellm.ipynb

```

question_prompt = (
    f"[INST] Given the following text passage, generate 1 factual question "
    f"that can be answered directly and explicitly from the passage. "
    f"Ensure the question is clear and concise." # Adjusted for singular
    f"\n\nPassage:\n{passage_text}\n[/INST]\n"
)

```

APPENDIX C.1: Implementation Details for Meta-LLaMA-3-8B-Instruct Control Group

Following figure shows the workflow for using inference against the base LLM model to generate answers from the Control Group to the same set of questions as fed to the two treatment groups. Appendix A has the prompt used for the LLM model. We use a very basic prompt only instructing the model to answer the question given the passage.

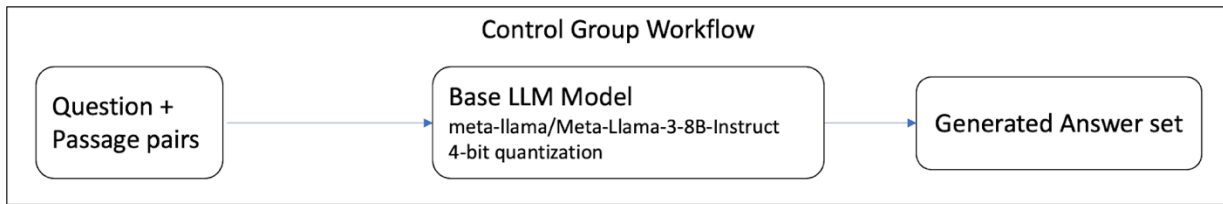


Figure 4:

Fine-tuning workflow

Prompt Format

Use the following context to answer the question. Only use the information in the context.

Context: {context}

Question: {question}

APPENDIX C.2 Implementation Details for Fine-Tuning Meta-LLaMA-3-8B-Instruct with QLoRA for Passage-Based Question Answering for Treatment Group 1

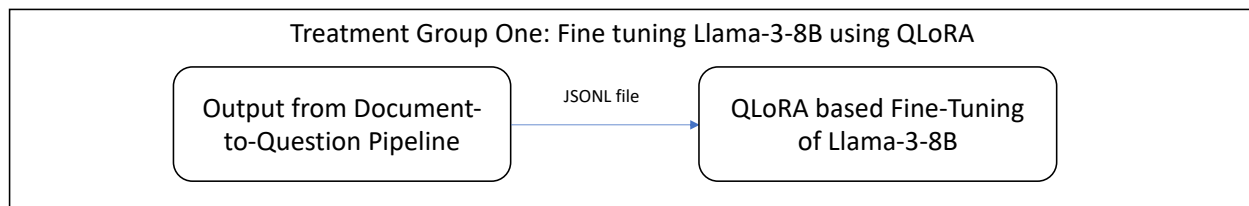


Figure 5: *Fine-tuning workflow*

We fine-tuned the 8B parameter Meta-LLaMA-3-Instruct model using QLoRA (quantized low-rank adaptation) on a custom dataset of passage-question-answer examples. The goal was to enable the model to answer questions using only the information in the associated passage, ensuring concise, grounded, and faithful responses.

Prompt Format

Each training instance was formatted using a structured instruction-following prompt that aligns with LLaMA-3's conversational tuning format:

System:

You are a helpful assistant that answers questions based on the provided passage. Ensure your answer is concise and directly addresses the question using information only from the passage.

User:

Passage: {passage}

Question: {question}

Assistant:

{answer}

This format was tokenized in full, with special masking to ensure loss was only computed on the assistant's response.

Loss Computation

To ensure the model learned to generate only the assistant's answer:

- The entire prompt (system, user, and assistant) was tokenized.
- The `labels` tensor was created by copying `input_ids` but masking all tokens before the assistant's response using the value `-100`.
- During training, Hugging Face's `CrossEntropyLoss(ignore_index=-100)` computed loss only on the tokens in the assistant's answer.

This technique allows the model to learn appropriate completions without being penalized for not reproducing the prompt scaffolding.

Model & Training Configuration

We used QLoRA to fine-tune Llama-3-8B to act as the answer generator for the treatment group (see workflow figure above). Key design and parameter details along with the git hub code file name are provided in the table below.

For LoRA hyperparameters, we used rank = 16 and dropout = 0.05. We applied low-rank adaptation to the core linear layers of the transformer, targeting both the attention sublayers (q_proj, k_proj, v_proj, o_proj) and the feed-forward layers (gate_proj, up_proj, down_proj) — the latter based on architectural analysis of models like LLaMA and Mistral.

This design was inspired by the original LoRA paper, where the best performance was achieved by adapting all attention projections while still using a low rank adaptation. We extended this idea to FFNN layers, though a future ablation may examine the effect of removing FFNN layers from LoRA injection.

With LoRA, the number of trainable parameters was reduced from 8.072 billion to 41.943 million, a 99.48% reduction. Training was performed over 3 epochs, during which the training loss dropped from 1.244 to 0.126, suggesting some overfitting. Future work could explore: Fewer training epochs; increased corpus size and diversity; lower adapter rank (e.g., 8 or 4); removing FFNN layers from adaptation to reduce overfitting.

Our training run completed in 81 minutes on a single NVIDIA A100 GPU with high VRAM (≥40 GB), using QLoRA (4-bit quantization) for memory efficiency. After fine-tuning, we merged the LoRA adapters into the base model weights to produce a consolidated model for inference. This step helps reduce inference latency by eliminating the need to maintain separate adapter paths during runtime.

Additionally, we persisted the trained LoRA adapters separately (i.e., without merging) to enable reuse by other experiments or teams, such as Treatment Group 2. This provides flexibility to apply the adapters on other compatible base models or to continue further fine-tuning.

Aspect / Parameter	Value
Code file	266_FP_VS_VZ2_FineTuning_llama_8B.ipynb
LLM	meta-llama/Meta-Llama-3-8B-Instruct
Fine-Tuning Technique	QLoRA (Quantized Low-Rank Adaptation), 4-bit quantization
Input Format	JSONL file with id, question, passage, answer triplets. 4,733 triplets for training.

Table 6: Key design and parameter details for fine tuning Llama-3-8B

Tokenizer parameters

- pad_token set to eos_token
- padding direction: right-aligned (standard for causal models)

Training Hyperparameters

Parameter	Value
Epochs	3
Batch Size (per device)	1

Gradient Accumulation	8
Learning Rate	2e-4
Optimizer	paged_adamw_8bit
Max Sequence Length	2048
Scheduler	Linear
Weight Decay	0.01
Logging Steps	10
BF16 Enabled	Yes (if supported)

Output

The fine-tuned LoRA adapters and tokenizer were saved so that these adapters could be reused with the original base model for inference with minimal additional memory requirements.

Cross-Entropy Loss and Training Loss Computation

During fine-tuning, we used the standard cross-entropy loss function to train the model to generate the correct assistant response. This loss was computed token by token over the assistant's answer (the part of the input not masked as -100). The model outputs a probability distribution over the full vocabulary at each time step, and the cross-entropy loss compares the predicted distribution to the actual target token (the next token in the answer). During training, the total loss was the average of the cross-entropy losses across all unmasked tokens in the batch. Masked tokens (with label -100) were ignored in the computation due to the use of `ignore_index=-100` in the loss function implementation.

APPENDIX C3: Treatment Group 2's RAG Pipeline Details

Phase 1: Knowledge Ingestion

In this phase, we ingested and indexed domain-specific knowledge from a structured source document to enable semantic retrieval during inference

Steps:

1. PDF Loading

Source: geetha_vahini.pdf

Loaded using PyPDFLoader into LangChain Document objects (one per page).

2. Chunking

Text was split into overlapping semantic chunks using RecursiveCharacterTextSplitter with:

chunk_size = 1000 characters

chunk_overlap = 200

This ensures contextual continuity for downstream QA tasks.

3. Embedding Creation:

Each chunk was embedded using BAAI/bge-small-en-v1.5, a high-quality sentence transformer.

Embeddings are 384-dimensional vectors suitable for semantic retrieval.

4. Vector Store Construction:

All embedded chunks were stored using Chroma, a fast persistent vector database.

- The resulting vector store was saved.

Phase 2: Inference and Answer Generation

This phase generated faithful answers to natural language questions using a fine-tuned LLaMA-3 model (same model as was fine-tuned for TG-1) augmented with retrieval.

Model Setup:

Fine-tuned LLaMA-3-8B QLoRA (4-bit) adapters from Treatment Group were loaded. These adapters were merged into the base model for efficient inference

Retriever:

Top-3 most semantically relevant chunks were retrieved using the precomputed Chroma vector store and BGE embeddings

Prompt Template:

The model was conditioned using the following chat-style RAG prompt:

System:

You are a helpful assistant that answers questions based ONLY on the provided context.

If the answer is not in the context, state that you don't know.

Context: {retrieved_context}

User:

{question}

Assistant:

{generated_answer}

Generation Settings:

- max_new_tokens = 150
- top_p = 0.9
- temperature = 0.7
- padding_side = "left" for causal generation

Inference Process:

- Each unit test question was processed individually
- The RAG chain retrieved context, generated an answer, and stored results as JSONL file.

Output Format Example:

```
{  
  "id": 27,  
  "question": "What is the purpose of the Geetha?",  
  "generated_answer": "The Geetha provides spiritual guidance through a dialogue between Krishna and Arjuna.",  
  "retrieved_chunks": 3  
}
```


APPENDIX D: Building an LLM-as-a-Judge

Given that our dataset included only a context passage, question, and generated answer, we used pointwise evaluation to assign a score to each system response. Over the course of three main development cycles, we calibrated the LLM evaluator using a multi-prompt evaluation pipeline. Each evaluation metric (relevance, faithfulness, completeness) was assessed independently using its own dedicated prompt. This architecture aligns with prompt-chaining frameworks and minimizes cognitive load per prompt, resulting in consistently higher model scores and more stable scoring behavior [2]. Spearman ρ and Kendall τ are common measurements to evaluate alignment between an LLM evaluator and human judgment where the target for a strong LLM evaluation pipeline is a Spearman $\rho \geq 0.8$ and Kendall $\tau \geq 0.65$ [8]. Additionally, we evaluated the accuracy of the LLM-based scorer by comparing its ratings to those of human annotators. To account for common score inflation observed in LLM evaluations, we also calculated an accuracy with a threshold metric that considered predictions correct if they deviated by no more than one point from the human-assigned scores [4]. While these values are reflected in Spearman ρ and Kendall τ , it gives us a quick, tangible reference point between model iterations during development.

To efficiently process the large volume of data, we leveraged OpenAI’s batch API to reduce latency and improve both consistency and ranking calibration across the dataset. The LLM evaluator’s outputs were then compared to human judgments, with Spearman’s ρ and Kendall’s τ computed to assess alignment and track improvements over time.

We experimented with various prompt phrasings and few-shot examples, selecting the configuration that yielded the highest correlation with human judgments. Once the correlation plateaued, further gains required architectural changes rather than prompt-level tuning. The temperature for all iterations was set to 0 to be more deterministic and reduce randomness.

The initial version of the LLM-as-a-judge (v0) adhered to established prompting practices: assigning a clear role to the model, enforcing a strict output format, and using a well-defined rubric. Few-shot examples were drawn from a balanced mix of excellent, moderate, and poor ratings across relevance, faithfulness, and completeness. While this approach produced solid results, it fell just short of our performance targets (see Table 1).

In the next iteration (v1), we refined the prompt more deliberately, tuning the system message and individual rubrics for relevance, faithfulness, and completeness. Each few-shot example was paired with an explanation from a human judge, clearly articulating the rationale behind each score. To further enhance variability and reduce overfitting, we drew examples from both control and treatment groups, capturing a broader diversity in language and style.

For version v2, we introduced an aggregation step to the evaluation process. This involved generating three completions per judgment and testing several aggregation strategies, ultimately finding that averaging the scores (mean) yielded the most consistency.

Evaluation Summary								
Metric	Spearman ρ			Kendall τ			Accuracy	Acc. w/ threshold
Revision	Relevance	Faithfulness	Completeness	Relevance	Faithfulness	Completeness	+/- 0	+/- 1
v0	0.749400	0.732500	0.835200	0.705100	0.679000	0.767200	0.533300	0.900000
v1	0.759300	0.798700	0.839700	0.718000	0.765400	0.797900	0.533300	0.883300
v2	0.756400	0.774300	0.835200	0.724200	0.711200	0.767200	0.516700	0.883300

Table 1: Spearman’s ρ and Kendall’s τ evaluation for each version of the LLM evaluator

Ultimately, we selected v1 since it had the highest Spearman ρ and Kendall τ correlation in the most categories while maintaining accuracy. While the Spearman ρ correlation only met the target benchmark in completeness, the Kendall τ correlation is strong in relevance, faithfulness and completeness for all versions of the LLM evaluator.

APPENDIX E: Unit Test (Gold QA) Set used for Prompt Training LLM-as-a-Judge

Here are some of the context passage, questions and answers used to calibrate the LLM-as-a-judge. The answers are scored by a human, and the LLM-as-a-judge will aim to match it.

Example 1: Excellent answer – relevance (5), faithfulness (5), completeness (5)

Context Passage: If it is stated that women are not entitled to it, why is it mentioned that Siva taught Vedantha to Parvathi? Or, how did Kapilacharya, the great Yogi, teach the Sankhya system to his mother, Devahoothi? Or, Yajnavalkya the great Rishi impart the essential principles of Vedanthic philosophy to his wife, Maithreyi, as mentioned in the Brihadaranyaka Upanishad ? The Upanishad cannot be false. The Scriptures where these facts are mentioned speak only the truth. There is no doubt that sage Mathanga is a great ascetic.

Question: How does the denial of scriptural access to women reflect deeper misconceptions about consciousness and embodiment in Vedantic thought?

Answer: The exclusion of women from Vedic study reflects a misunderstanding of Vedantic principles, which regard the Self as beyond bodily distinctions; since Atma is universal and genderless, any restriction based on gender pertains only to the body and not to the spiritual essence, which is equally entitled to liberation.

Example 2: Fair answer – relevance (2), faithfulness (2), completeness (2)

Context Passage: Without a clear understanding of the play in which they are acting their roles, people cling the error that they are Jivis or Sadhakas and waste away their lives, buffeted by the waves of joy and sorrow. When the mystery is cleared, and the play is discovered as mere play, the conviction dawns that you are He and He is you. Therefore, try to know the Truth behind Life, search for the Fundamental, bravely pursue the underlying Reality. Seekers of Jnana must always be conscious of this.

Question: What does the metaphor of life as a 'play in which one is cast' reveal about the human struggle with identity and detachment?

Answer: Somethings move, other things stay stationary, while trucks and cars are both useful as modes of transportation.

APPENDIX F: Evaluation criteria definitions and their Likert Scales

Metric	Definition
Relevance	<p>You are a grading assistant for a subject matter expert. Your job is to apply the rubric carefully and consistently using only the context provided. Do not use external knowledge or reward fluency alone. Relevance is about how well the answer directly addresses the question, not just the general topic.</p> <p>Rate each answer from 1 to 5: 5 – Fully answers the question, highly focused and on-topic. 4 – Mostly answers the question, but includes minor unrelated information. 3 – Partially addresses the question, some digression or vagueness. 2 – Weakly related to the question; mostly off-topic. 1 – Irrelevant or unrelated to the question.</p>
Faithfulness	<p>You are a grading assistant for a subject matter expert. Evaluate how factually consistent the answer is with the provided context. Do not rely on outside knowledge. Do not assume — only accept content that is explicitly or clearly supported.</p> <p>Faithfulness scores: 5 – Fully supported by the context; no hallucination or unsupported claims. 4 – Mostly supported; minor inference or vague phrasing not directly grounded. 3 – Partially supported; contains some unsupported or ambiguous statements. 2 – Minimally supported; includes clear inaccuracies or weak grounding. 1 – Contradicts or fabricates information; no meaningful support from context.</p>

Completeness	<p>You are a grading assistant for a subject matter expert.</p> <p>Assess how completely the answer addresses all parts of the question using only the provided context. Do not assume or reward unnecessary elaboration. A brief but complete answer may receive a 5.</p> <p>Completeness scores:</p> <p>5 – Fully answers all aspects of the question; nothing important is missing.</p> <p>4 – Mostly complete; one minor detail or part may be missing.</p> <p>3 – Partially complete; some important information is missing.</p> <p>2 – Incomplete; only covers a small portion of the necessary content.</p> <p>1 – Severely incomplete; fails to address the question meaningfully.</p>
--------------	--

APPENDIX G: Base-Llama Negative Correlation Exploration

A negative correlation indicates that one variable is increasing, while the other is decreasing. In this case, the LLM evaluator scored the answer high where the human evaluator scored the answer low. Below are examples of varying levels of hallucinations from the control group that could contribute to a negative correlation between the LLM-as-a-judge and human evaluator

Example of Strong Hallucination

Context Passage: Ego consciousness leads to self-aggrandizement; and when self holds sway over the heart, no deed worthy to be styled Service can emanate from him. It is sheer selfish greed that makes him label his deed as Service. Ignorance will never vanish, until this discrimination dawns. "This world is but God and nothing else. Everything, every being is but His Manifestation, bearing withal a new name and a new form". Love this Truth, believe in it, and then you have the right to speak of Seva, JNANA VAHINI 42 Bhakthi and Dharma and the authority to preach those paths.

Question: What is the ultimate goal of separating the mind from external influences, as mentioned in the passage?

Generated Answer: The ultimate goal of separating the mind from external influences is to attain JNANA, which is the truth that this world is but God and nothing else. This realization allows one to love this truth, believe in it, and then have the right to speak of Seva, Bhakthi, and Dharma, and the authority to preach those paths.t.

Explanation of generated answer (Human evaluation 1, LLM evaluator 5): This is an example of strong hallucination. The answer is not grounded in the passage at all and constructs a response based on unrelated philosophical ideas.

Example of Mild Hallucination

Context Passage It is only when full knowledge is won that one can get liberated, or, in other words, attain Moksha (Liberation). After the attainment of the above said Atmic knowledge, one has to follow the path of Brahman (The absolute consciousness, God) and act according to the New Wisdom. All doubts that afflict the mind have to be solved by consulting those who know, or the Sadgurus JNANA VAHINI 2 (Philosophers and guides) one has the chance to meet.

Question: What is the result of attaining Moksha as described in the passage?

Generated Answer: The result of attaining Moksha (Liberation) is that one gets liberated. In other words, one attains liberation. (According to the passage, it is only when full knowledge is won that one can get liberated, or attain Moksha.) Note: Moksha is the ultimate goal of the spiritual path in Hinduism, and it is considered to be the state of liberation from the cycle of birth and death. It is often described as a state of ultimate bliss and self-realization. Source: JNANA VAHINI 2 (Philosophers and guides) - This is a book written by SRI SATHYA SAI BABA, a spiritual leader and guru. The passage is taken

Explanation of generated answer (Human evaluation 1, LLM evaluator 5): "It is often described as a state of ultimate bliss and self-realization" Although true, this is an external insertion that is not mentioned in the context passage.

Example of Lack of Grounding in Text

Context Passage: When one takes in an intoxicant, one is not aware of pain, is it not? How does this happen? The mind is then detached from the body and so, it is not bothered by physical pain or discomfort. Similarly, the Jnani too has immersed his mind in the Atma. He can establish mental peace and quiet, by disciplining the mind. The Jnani gets full Bliss from his own Atma. He does not seek it anywhere outside himself. In fact, he will have no desire or plan to find joy in anything external. He is satisfied with the inner joy he gets.

Question: How does the mind contribute to the cycles of bondage and liberation?

Generated Answer: The mind contributes to the cycles of bondage and liberation by being detached from the body and being immersed in the Atma. When the mind is detached from the body, it is not bothered by physical pain or discomfort. Similarly, when the Jnani immerses his mind in the Atma, he can establish mental peace and quiet, and get full Bliss from his own Atma. The mind also helps in disciplining itself, which is essential for achieving liberation. In essence, the mind plays a crucial role in both bondage and liberation. It is the key to achieving inner joy and satisfaction, and it is also the source of suffering when it is not disciplined. Therefore, the mind must be controlled and directed towards the Atma in order to

Explanation of generated answer (Human evaluation 1, LLM evaluator 5): 'The mind contributes to the cycles of bondage...' The passage never mentions bondage, suffering, or samsara.

Not directly grounded the text:

"The mind must be controlled and directed toward the Atma"

"the mind is the source of suffering when undisciplined"