Real world RAG System

Project Report - Group 3

### Prakash Adidam, Venkat Yadavalli, Shiv Tripathi, Jayavardhan February 23, 2024

#### Mentors:

* Mr. Gopichand Kanumolu
* Mr. Lokesh Madasu

**Contents**

[**1**](#_heading=h.30j0zll) 2

[**2**](#_heading=h.3znysh7) 2

[2.1](#_heading=h.2et92p0) 2

[2.2](#_heading=h.tyjcwt) 2

[2.3](#_heading=h.3dy6vkm) 3

[2.4](#_heading=h.4d34og8) 4

[2.5](#_heading=h.17dp8vu) 5

[2.6](#_heading=h.lnxbz9) 5

[2.7](#_heading=h.1ksv4uv) 9

[**3**](#_heading=h.2jxsxqh) 9

[3.1](#_heading=h.3j2qqm3) 9

[3.2](#_heading=h.4i7ojhp) 9

[3.3](#_heading=h.1ci93xb) 9

[3.4](#_heading=h.2bn6wsx) 10

[3.5](#_heading=h.3as4poj) 10

[3.6](#_heading=h.49x2ik5) 11

[3.7](#_heading=h.147n2zr) 11

[3.8](#_heading=h.23ckvvd) 11

[3.9](#_heading=h.32hioqz) 11

[**4**](#_heading=h.41mghml) 13

# Introduction

In this project, we explored the capabilities of Retrieval-Augmented Generation (RAG) systems to enhance information retrieval and document ranking using advanced embedding models and search techniques. The project was structured into multiple stages, focusing on:

* Task 1: Setting up the RAG system pipeline and evaluation on RAGBench dataset.
* Task 2: RAG System Evaluation on RGB Dataset.

These tasks provided hands-on experience in dataset curation, model evaluation, and the deployment of a scalable and efficient RAG system. The project was executed in a phased manner, with each phase concentrating on specific retrieval and ranking improvements.

# Task 1: RAG System pipeline setup

## Problem Description

The evolution of large language models (LLMs) has significantly advanced natural language understanding and generation. However, limitations like hallucination, inability to retrieve updated or domain-specific information, and inefficiencies in handling diverse information needs present critical challenges. RAG systems aim to bridge this gap by integrating information retrieval mechanisms with LLMs to enhance reliability, accuracy, and adaptability. The goal is to build a scalable, efficient RAG framework that performs well across dynamic, high-impact domains such as healthcare, legal, finance, and customer support.

## Dataset

This proposal leverages datasets highlighted in RAGBench and RGB, designed to evaluate RAG systems across diverse challenges and domains.

**RAGBench Dataset**

Dataset: <https://huggingface.co/datasets/rungalileo/ragbench>

RAG Bench is a large-scale benchmark dataset specifically designed for training and evaluating RAG systems. It contains 100k examples spanning five industry-specific domains which includes textual data like Questions, documents, responses, explanations, and sentences, numerical data like Evaluation scores (e.g., relevance, utilization, completeness) and categorical data like Model names, dataset names, and Boolean scores (e.g., adherence).

* Biomedical Research
* General Knowledge
* Legal
* Customer Support
* Finance

**Description**: The dataset contains entries related to questions, responses, and their evaluations in the context of retrieval-augmented generation (RAG) systems, with a focus on models such as GPT-3.5 and GPT-4. It includes metadata, evaluation metrics, and annotations

**Biomedical Domain**:

* *PubMedQA*: A dataset of biomedical questions based on PubMed abstracts requiring accurate retrieval and domain-aware responses.
* *CovidQA*: Focuses on literature related to COVID-19, ideal for testing retrieval in dynamic and evolving knowledge domains.

**General Knowledge**:

* *HotpotQA*: Multi-hop reasoning questions where answers depend on retrieving and synthesizing information from multiple documents.
* *MS MARCO*: Real-world search engine queries that assess the system's ability to handle diverse and ambiguous user inputs.

**Legal Domain**:

* *CUAD*: A dataset of annotated legal contracts challenging models to work with highly structured and formal language.

**Customer Support**:

* *TechQA*: Queries from technical documentation, requiring precision in retrieving specific troubleshooting information.
* *EManual*: Troubleshooting questions derived from user manuals.

**Finance Domain**:

* *FinBench*: Focuses on financial documents, assessing the system's ability to jointly interpret tabular and textual data for QA tasks.

## Methodology

Following is an overview of the methodology used for building this RAG system. This approach takes a modular pipeline optimized for efficiency and scalability.

Key steps in RAG

1. ***Sentence Level Chunking with BAAI/LLM-Embedder:***

We implemented sentence-level chunking using BAAI/LLM-Embedder to ensure efficient text segmentation. This method helped in preserving contextual integrity while reducing noise. Each sentence was embedded separately, allowing better retrieval performance in downstream tasks.

1. ***Sliding window overlap with token limit:***

To enhance context preservation, we applied a sliding window approach with a predefined token limit. This method ensured that overlapping segments retained crucial information while avoiding excessive redundancy. We experimented with different overlap sizes to find the optimal balance between coherence and performance.

1. ***Small to big chunking:***

We experimented with a progressive chunking approach, where smaller text segments were initially processed, and larger chunks were formed based on relevance scores. This technique improved retrieval efficiency and ensured that essential information was not lost.

1. ***Evaluation of Chroma DB vs Milvus DB and DB selection:***

We conducted a comparative evaluation between Chroma DB and Milvus DB based on retrieval speed, scalability, and integration flexibility. Milvus demonstrated superior performance in handling high-dimensional vector searches and scalability, leading us to finalize it as our primary vector database.

1. ***Evaluation and implementation of Query Classification:***

To refine search performance, we implemented query classification to differentiate between fact-based, exploratory, and contextual queries. This classification improved the precision of retrieval by directing LLM or the RAG.

1. ***Dense, Sparse, Hybrid Search:***

We implemented and evaluated different search techniques:

* Sparse Search: Utilized BM25 for keyword-based retrieval.
* Dense Search: Used embeddings from various LLMs to match semantically similar documents.
* Hybrid Search: Combined both sparse and dense techniques to improve recall and precision.

Hybrid search provided the best balance between accuracy and computational efficiency.

1. ***Pseudo document generation using HyDE:***

We explored Hypothetical Document Embeddings (HyDE) to generate pseudo documents, aiding in query expansion and retrieval enhancement. This method significantly improved retrieval results, especially for underrepresented query types.

1. ***Doc Summarization (Recomp), adherence score comparison with and without Summarization:***

We implemented document summarization using Recomp to condense lengthy text while preserving key information. Adherence scores were computed to compare search effectiveness with and without summarization. Results indicated that summarization improved retrieval precision by reducing noise and redundancy.

1. ***Document re-ranking with monot5 and cross-encoder:***

To enhance retrieval accuracy, we compared MonoT5-based ranking with cross-encoder models. Cross-encoder-based re-ranking yielded better precision, leading us to proceed with it for final implementation.

1. ***Computed RMSE and AUCROC metrics for samples in all 12 datasets:***

To measure the effectiveness of our approach, we calculated Root Mean Square Error (RMSE) and Area Under the ROC Curve (AUCROC) across 12 diverse datasets. These metrics helped in evaluating the reliability and robustness of our retrieval pipeline.

1. ***Compare metrics with deepseek-r1-distill-llama-70b & llama3-8b-8192:***

We compared retrieval performance using different LLM embedders, specifically DeepSeek-R1-Distill-LLaMA-70B and LLaMA3-8B-8192. Our evaluation indicated that LLaMA3-8B-8192 offered better efficiency for our use case while maintaining competitive accuracy.

1. ***Evaluate and implement Noise Robustness, Negative Rejection, Information Integration, Counterfactual Robustness on RGB dataset:***

We tested the system for robustness by introducing noise, evaluating its rejection of irrelevant information, and measuring its performance in integrating useful knowledge. Additionally, counterfactual robustness was assessed by modifying inputs to ensure consistent and reliable retrieval.

1. ***Generate comparative metrics using different LLM embedders:***

To select the best LLM embedder, we compared multiple models based on retrieval accuracy, inference time, and scalability. This analysis helped in refining our final embedding model selection.

1. ***Generate comparative metrics for cross-encoder vs. monot5 re-tanking:***

Cross-encoder-based re-ranking consistently outperformed MonoT5 in terms of precision and retrieval relevance. This analysis reinforced our decision to proceed with cross-encoder for final implementation.

1. ***Generate comparative metrics for hybrid search vs. dense search:***

Hybrid search demonstrated superior performance in retrieving both keyword-matching and semantically relevant documents. Dense search alone, while effective in semantic matching, lacked the precision of hybrid search, reinforcing our decision to integrate both approaches.

## Models Used

We fine-tuned and evaluated several models for the task. The following table provides an overview of the models used:

|  |  |
| --- | --- |
| **Model** | **Training Notebook** |
| llama3-8b-8192 | [Real\_World\_RAG\_Impl.ipynb](https://github.com/adidam/rag-impl/blob/main/Real_World_RAG_Impl.ipynb) |
| llama3-70b-8192 | [Real\_World\_RAG\_Impl.ipynb](https://github.com/adidam/rag-impl/blob/main/Real_World_RAG_Impl.ipynb) |
| deepseek-r1-distill-llama-70b | [Real\_World\_RAG\_Impl.ipynb](https://github.com/adidam/rag-impl/blob/main/Real_World_RAG_Impl.ipynb) |
| cross-encoder/ms-marco-MiniLM-L-6-v2 | [Real\_World\_RAG\_Impl.ipynb](https://github.com/adidam/rag-impl/blob/main/Real_World_RAG_Impl.ipynb) |
| BAAI/LLM-Embedder | [Real\_World\_RAG\_Impl.ipynb](https://github.com/adidam/rag-impl/blob/main/Real_World_RAG_Impl.ipynb) |
|  |  |
|  |  |

Table 1: Models used for training and testing

## Evaluation Metrics for RAG Pipeline

We conducted RMSE and AUC-ROC on each of the models.

* + - Root Mean Square Error (RMSE) Evaluation
    - Area Under the ROC Curve (AUC-ROC)

Our evaluation framework leverages RMSE and AUC-ROC to measure retrieval and response quality. The stable RMSE scores indicate that retrieval variations had minimal impact on final outputs, while AUC-ROC provided additional insights into adherence quality.

## 

## Performance Metrics

We evaluated the RMSE, AUC-ROC for each search and with combination of models. The following tables provides a detail scores:

* Sparse search

|  |  |  |  |
| --- | --- | --- | --- |
| **DataSet** | **RMSE relevance** | **RMSE Utilization** | **AUCROC** |
| covidqa | [0.6842105263157895, 0.4117647058823529, 0.8260869565217391] | [0.6842105263157895, 0.3382352941176471, 0.8695652173913043] | 0.5 |
| cuad | [0.0, 0.16666666666666663, 0.9966996699669967] | [0.0, 0.9642857142857143, 0.9966996699669967] | nan |
| delucionqa | [0.6785714285714286, 0.9047619047619048, 0.6923076923076923] | [0.29285714285714287, 0.1285714285714286, 0.8846153846153846] | nan |
| emanual | [0.16666666666666666, 0.7916666666666666, 0.5294117647058824] | [0.16666666666666666, 0.875, 0.5294117647058824] | 0.25 |
| expertqa | [0.4375, 0.8787878787878788] | [0.1875, 0.5681818181818181] | 0.5 |
| finqa | [0.96, 0.030303030303030304, 0.9411764705882353] | [0.04, 0.030303030303030304, 0.9411764705882353] | nan |
| hagrid | [0.75, 1.0] | [0.75, 1.0] | nan |
| hotpotqa | [0.8, 0.10526315789473684, 0.8666666666666667, 0.4] | [0.8, 0.10526315789473684, 0.13333333333333333, 0.3] | nan |
| msmarco | [0.5483870967741935, 0.46153846153846156, 0.41025641025641024] | [0.2043010752688172, 0.3435897435897436, 0.8717948717948718] | nan |
| pubmedqa | [0.8571428571428572, 0.5454545454545454, 0.5384615384615384, 0.18181818181818177] | [0.8571428571428572, 0.8181818181818181, 0.09230769230769231, 0.18181818181818182] | 0.5 |
| tatqa | [0.6666666666666667, 0.8888888888888888, 0.1111111111111111] | [0.6666666666666667, 0.8888888888888888, 0.1111111111111111] | nan |
| techqa | [0.9858823529411764, 1.0, 0.8736842105263158] | [0.19058823529411767, 0.5, 0.9789473684210527] | 0.5 |

* Dense search

|  |  |  |  |
| --- | --- | --- | --- |
| **DataSet** | **RMSE relevance** | **RMSE Utilization** | **AUCROC** |
| covidqa | [0.6842105263157895, 0.4117647058823529, 0.8260869565217391] | [0.6842105263157895, 0.3382352941176471, 0.8695652173913043] | 0.5 |
| cuad | [0.0, 0.16666666666666663, 0.9966996699669967] | [0.0, 0.2142857142857143, 0.9966996699669967] | nan |
| delucionqa | [0.6785714285714286, 0.9047619047619048, 0.6923076923076923, 0.9047619047619048] | [0.39285714285714285, 0.4285714285714286, 0.8846153846153846, 0.8571428571428572] | nan |
| emanual | [0.16666666666666666, 0.7916666666666666, 0.5294117647058824, 0.9130434782608696] | [0.16666666666666666, 0.875, 0.5294117647058824, 3.9130434782608696] | 0.5 |
| expertqa | [0.4375, 0.8787878787878788, 0.75] | [0.4375, 0.8181818181818181, 0.75] | 0.25 |
| finqa | [0.96, 0.9696969696969697, 0.058823529411764705, 0.8888888888888888] | [0.96, 0.9696969696969697, 0.058823529411764705, 0.8888888888888888] | nan |
| hagrid | [0.25, 1.0, 0.75] | [0.25, 1.0, 1.75] | nan |
| hotpotqa | [0.8, 0.10526315789473684, 0.8666666666666667, 0.4] | [0.8, 0.10526315789473684, 0.13333333333333333, 0.3] | nan |
| msmarco | [0.5483870967741935, 0.46153846153846156, 0.41025641025641024, 0.23529411764705888, 0.75] | [0.8709677419354839, 0.2435897435897436, 0.8717948717948718, 0.36470588235294116, 0.8] | nan |
| pubmedqa | [0.8571428571428572, 0.5454545454545454, 0.5384615384615384] | [0.35714285714285715, 0.18181818181818182, 0.49230769230769234] | 0.5 |
| tatqa | [0.6666666666666667, 0.8888888888888888, 0.1111111111111111] | [0.6666666666666667, 0.8888888888888888, 0.1111111111111111] | nan |
| techqa | [0.9858823529411764, 1.0] | [0.9905882352941177, 1.0] | 0.5 |

* Hybrid search

|  |  |  |  |
| --- | --- | --- | --- |
| **DataSet** | **RMSE relevance** | **RMSE Utilization** | **AUCROC** |
| covidqa | [0.6842105263157895, 0.5882352941176471, 0.8260869565217391, 0.08333333333333333, 0.05555555555555555] | [0.6842105263157895, 0.5882352941176471, 0.8695652173913043, 0.041666666666666664, 0.05555555555555555] | 0.5 |
| cuad | [0.0, 0.16666666666666663, 0.9966996699669967, 0.9616724738675958] | [0.0, 0.6309523809523809, 0.9966996699669967, 0.9965156794425087] | 0.5 |
| delucionqa | [0.6785714285714286, 0.9047619047619048, 0.6923076923076923, 0.9047619047619048] | [0.8928571428571429, 0.17857142857142858, 0.8846153846153846, 0.6071428571428572] | nan |
| emanual | [0.16666666666666666, 0.7916666666666666, 0.5294117647058824, 0.9130434782608696] | [0.16666666666666666, 0.625, 0.5294117647058824, 0.9130434782608696] | 0.333333 |
| expertqa | [0.4375, 0.8787878787878788, 0.75, 0.7678571428571428] | [0.35416666666666663, 0.5681818181818181, 0.08333333333333331, 0.475] | 0.833333 |
| finqa | [0.96, 0.9696969696969697, 0.058823529411764705, 0.8888888888888888] | [0.96, 0.9696969696969697, 0.058823529411764705, 0.8888888888888888] | nan |
| hagrid | [0.25, 1.0] | [0.25, 0.5] | nan |
| hotpotqa | [0.8, 0.10526315789473684, 0.8666666666666667, 0.4] | [0.8, 0.10526315789473684, 0.8666666666666667, 0.3] | nan |
| msmarco | [0.5483870967741935, 0.46153846153846156, 0.41025641025641024] | [0.8709677419354839, 0.4935897435897436, 0.8717948717948718] | nan |
| pubmedqa | [0.8571428571428572, 0.5454545454545454, 0.5384615384615384, 0.18181818181818177, 0.36363636363636365] | [0.8571428571428572, 0.2181818181818182, 0.3076923076923077, 0.06818181818181818, 0.12121212121212122] | 0.25 |
| tatqa | [0.6666666666666667, 0.8888888888888888, 0.1111111111111111] | [0.6666666666666667, 0.8888888888888888, 0.1111111111111111] | nan |
| techqa | [0.9858823529411764, 1.0, 0.8736842105263158, 0.8571428571428572] | [0.19058823529411767, 0.5, 0.6456140350877193, 0.3660714285714286] | 0.5 |

* RMSE & AUCROC with llama3-8b-8192 and deepseek-r1-distill-llama-70b

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **RMSE-Relevance** | | **RMSE-Utilization** | | **AUCROC** | |
|  | **llama3-8b** | **deepseek-r1** | **llama3-8b** | **deepseek-r1** | **llama3-8b** | **deepseek-r1** |
| covidqa | [0.68421, 0.41176, 0.8208, 0.9166, 0.05555] | [0.68421, 0.411, 0.82681, 0.9166, 0.9444] | [0.115785, 0.0882, 0.53623, | [0.6845, 0.5882, 0.86956, 0.95833, 0.94444] | 0.375 | 0.5 |
| 0.9583, 0.05555] |
| cuad | [0.0, 0.83333, 0.0033 | [0.31575, 0.411, 0.8261, 0.9666] | [0.0, 0.03571, 0.00330, 0.003] | [0.315705, 0.07855, 0.86953, 0.95834] | 0.5 | 0.83 |
| 0033, 0.03832] |
| delucionqa | [0.6785, 0.09523, 0.30769] | [0.684, 0.4117, 0.826, 0.9166] | [0.89285, 0.071428, 0.1153] | [0.6842, 0.41179, 0.86943, 0.95834] | N/A | 0.83 |
| tatqa | [0.666, 0.8888, 0.11111] | [0.667, 0.7214,0.11111] | [0.6666, 0.88888, 0.1111] | [0.667, 0.7214,0.11111] | N/A | N/A |
| Emanual | [0.83333, 0.7916, 0.5294, 0.08695] | [0.8334, 0.79166, 0.5294, 0.08695] | [0.16666, 0.875, 0.1960, 0.086956] | [0.8334, 0.6750.1960, 0.086956] | 0.83 | 0.83 |
| Expertqa | [0.4375, 0.8787, 0.75, 0.23215] | [0.4375, 0.878, 0.75, 0.2345] | [0.35416, 0.56818, 0.41666, 0.125] | [0.3543, 0.818181, 0.75,0.256] | 0.5 | 0.25 |
| Finqa | [0.5625, 0.121212, 0.25, 0.23215] | [0.96, 0.0303, 0.9413, 0.8888, 0.10584] | [0.3125, 0.18181, 0.25, 0.125] | [0.96, 0.0304, 1.60782, 0.88888, 0.056842] | 0.5 | N/A |
| hagrid | [0.25, 0.0, 0.25] | [0.96, 0.96969] | [0.25, 0.0, 0.25] | [0.293333, 0.96997] | N/A | N/A |
| hotpotqa | [0.8, 0.8947, 0.86667, 0.4] | [0.96, 0.034, 0.9413, 0.111] | [0.8, 0.5614, 0.866666, 0.3] | [0.96, 0.03304, 0.9453, 0.111] | N/A | N/A |
| msmarco | [0.4516, 0.5384, 0.58974, 0.7647] | [0.54838, 0.4615, 0.41024] | [0.12903, 0.25641, 0.1282, 0.2352] | [0.8709, 0.34336, 0.87178] | N/A | N/A |
| pubmedqa | [0.85714, 0.54545, 0.53846, 0.18181] | [0.85712, 0.55454, 0.5384, 0.18187] | [0.8571, 0.48484, 0.49230, 0.0681] | [0.852, 0.41815, 0.6076, 0.4115] | 0.33333 | 0.16 |
| techqa | [0.98588, 1.0, 0.87368, | [0.985884, 1.0, 0.1263151] | [0.1905, 0.5, 0.6453, 0.8660, 0.13285] | [0.990, 1.0, 0.02368] | 0.75 | 0.5 |
| 0.85714, 0.88537] |

* RMSE & AUCROC with BAAI/LLM-Embedder vs. BAAI/bge-large-en

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dataset** | **RMSE-Relevance** | | **RMSE-Utilization** | | **AUCROC** | |
|  | **LLM-Embedder** | **bge-large-en** | **LLM-Embedder** | **bge-large-en** | **LLM-Embedder** | **bge-large-en** |
| covidqa | [0.68421, 0.41176, 0.8208, 0.9166, 0.05555] | [0.68423, 0.58855,0.722941, 0.82608391] | [0.115785, 0.0882, 0.53623, | [0.684215, 0.588471, 0.6583, 0.13065] | 0.375 | 0.5 |
| 0.9583, 0.05555] |
| cuad | [0.0, 0.83333, 0.0033 | [0.0, 0.83, 0.9967] | [0.0, 0.03571, 0.00330, 0.003] | [0.0, 0.0357, 0.0034] | 0.5 | N/A |
| , 0.03832] |
| delucionqa | [0.6785, 0.09523, 0.30769] | [0.6786, 0.90448, 0.6963] | [0.89285, 0.071428, 0.1153] | [0.3985, 0.592, 0.38446] | N/A | N/A |
| tatqa | [0.666, 0.8888, 0.11111] | [0.6667, 0.888888, 0.1111] | [0.6666, 0.88888, 0.1111] | [0.6667, 0.888888, 0.1111] | N/A | N/A |
| Emanual | [0.83333, 0.7916, 0.5294, 0.08695] | [0.8334, 0.791666, 0.5294, 0.91696] | [0.16666, 0.875, 0.1960, 0.086956] | [0.166666, 0.875, 0.0843, 0.9644] | 0.83 | 0.875 |
| Expertqa | [0.4375, 0.8787, 0.75, 0.23215] | [0.4375, 0.8787, 0.75, 0.76728] | [0.35416, 0.56818, 0.41666, 0.125] | [0.35663, 0.31882, 0.25, 0.675] | 0.5 | 0.83 |
| Finqa | [0.5625, 0.121212, 0.25, 0.23215] | [0.96, 0.969697, 0.94113] | [0.3125, 0.18181, 0.25, 0.125] | [0.96, 0.967, 0.058] | 0.5 | N/A |
| hagrid | [0.25, 0.0, 0.25] | [0.75, 0.0,1.0] | [0.25, 0.0, 0.25] | [0.75,0.0, 1.0] | N/A | N/A |
| hotpotqa | [0.8, 0.8947, 0.86667, 0.4] | [0.8, 0.1054, 0.1333, 0.4] | [0.8, 0.5614, 0.866666, 0.3] | [0.3, 0.13684, 0.13333, 0.3] | N/A | N/A |
| msmarco | [0.4516, 0.5384, 0.58974, 0.7647] | [0.4516, 0.4615, 0.41024, 0.2388] | [0.12903, 0.25641, 0.1282, 0.2352] | [0.12903, 0.2564, 0.47, 0.7641] | N/A | 0.5 |
| pubmedqa | [0.85714, 0.54545, 0.53846, 0.18181] | [0.857, 0.54554, 0.53884, 0.18181] | [0.8571, 0.48484, 0.49230, 0.0681] | [0.8572, 0.81811, 0.69223, 0.068118] | 0.33333 | 0.5 |
| techqa | [0.98588, 1.0, 0.87368, | [0.9864, 1.0, 0.8738, 0.8572, 0.8852] | [0.1905, 0.5, 0.6453, 0.8660, 0.13285] | [0.323451, 0.666666, 0.9787, 0.19919, 0.596] | 0.75 | 0.5 |
| 0.85714, 0.88537] |

## 

## Gradio Integration for Deployment

To facilitate seamless interaction with our RAG pipeline, we utilized Gradio, an open-source Python library designed for deploying machine learning models through an intuitive web interface. Gradio enables rapid prototyping and provides an easy-to-use frontend for users to query the model and visualize responses in real time. By integrating Gradio, we ensured accessibility without requiring extensive frontend development, allowing non-technical stakeholders to interact with the model effortlessly. This deployment approach also allowed us to iterate quickly, test various retrieval strategies, and refine user interactions based on feedback. Gradio’s API flexibility and support for real-time inference made it an ideal choice for showcasing our model’s capabilities.

# Task 2: RAG System Evaluation on RGB Dataset

## Problem Description

The goal of task 2 is to evaluate the performance of a RAG system on the RGB dataset, specifically testing its robustness to noise, negative rejection, information integration, and counterfactual robustness. The system should be able to retrieve relevant knowledge accurately while filtering misleading or irrelevant information.

## Dataset

We The RGB dataset consists of queries and documents categorized into different robustness challenges, including noise, negative samples, information integration, and counterfactual scenarios.

It is preprocessed to normalize text, generate embeddings for retrieval, and apply ranking mechanisms.

## Data Preprocessing

The preprocessing for this task involved formatting the training dataset into an appropriate prompt format. We experimented with two prompting styles:

#### GPT-style prompting

#### Alpaca-style prompting

The following figures show the visualization of the training and test datasets.

A graph of a graph of a graph

AI-generated content may be incorrect.

Figure 1: Visualization of the training dataset

A graph of a number of blue bars

AI-generated content may be incorrect.

Figure 2: Visualization of the test dataset

## Models

We fine-tuned the following models for answering AIML-related questions:

|  |  |
| --- | --- |
| **Model** | **Training Notebook** |
| GPT2 | [QnA finetuning gpt2 V4.ipynb](https://github.com/ch22group16/capstone-GenAI/blob/main/Question_Answering_Training_and_Gradio_Noteboooks/QnA_finetuning_gpt2_V4.ipynb) |
| Llama 2 | [Capstone Group16 QnA finetuning LLAMA2 V1.ipynb](https://github.com/ch22group16/capstone-GenAI/blob/main/Question_Answering_Training_and_Gradio_Noteboooks/Capstone_Group16_QnA_finetuning_LLAMA2_V1.ipynb) |
| Gemma 2 | [QnA gemma 2 2b v5.ipynb](https://github.com/ch22group16/capstone-GenAI/blob/main/Question_Answering_Training_and_Gradio_Noteboooks/QnA_gemma_2_2b_v5.ipynb) |
| Llama 3 | [Capstone Group16 QnA finetuning LLAMA3 V1.ipynb](https://github.com/ch22group16/capstone-GenAI/blob/main/Question_Answering_Training_and_Gradio_Noteboooks/Capstone_Group16_QnA_finetuning_LLAMA3_V1.ipynb) |

Table 4: Models used for training and testing

## Zero Shot Testing

We performed zero-shot testing on each of the models before fine-tuning, and several challenges were identified:

* + - **Hallucinations**: Models generated irrelevant and repetitive information, such as the following:

**Question:** What is AlexNet?

**Generated Answer:** AlexNet is a web application that allows you to create and manage your own websites. It is a web application that allows you to create and manage your own websites.

* + - **Repetitive Answers**: The models often repeated parts of the answer. See the visualization below:

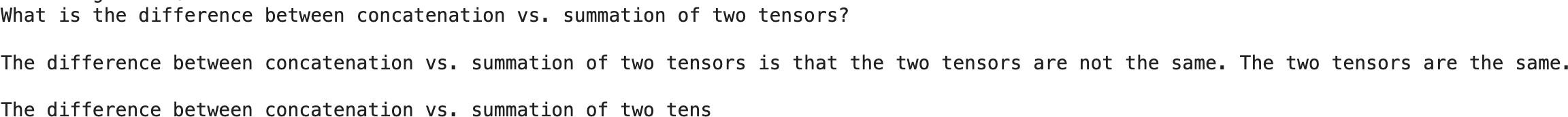


Figure 3: Repetitive answers in zero-shot testing

* + - We focused on hyperparameters such as:
      * Prompt Structure
      * Max New Tokens
      * Max Length and Min Length

## Model Fine-Tuning and Training

We fine-tuned and tested four models: GPT2, Llama 2, Gemma 2, and Llama 3. Initially, we tried encoder-decoder models, but we shifted to decoder-only models to improve performance on unseen topics.

A graph with a line

AI-generated content may be incorrect.

Figure 4: Llama 2 - 20 Epochs Training Loss

A graph with a line

AI-generated content may be incorrect.

Figure 5: Gemma 2 - 20 Epochs Training Loss

A graph showing a graph

AI-generated content may be incorrect.

Figure 6: Llama 3 - 20 Epochs Training Loss

## Model Weights

We saved all the fine-tuned model weights in Hugging Face Spaces:

* + - **GPT2**: [GPT2 for Q&A](https://huggingface.co/paramasivan27/gpt2_for_q_and_a)
    - **Llama 2**: [Llama 2 for Q&A](https://huggingface.co/paramasivan27/Llama-2-7b-for_q_and_a)
    - **Gemma 2**: [Gemma 2 for Q&A](https://huggingface.co/paramasivan27/Gemma_2b_it_q_and_a)
    - **Llama 3**: [Llama 3 for Q&A](https://huggingface.co/paramasivan27/llama-3-8b-bnb-4bit)

## ROUGE Score Comparison

The following table presents the ROUGE score comparison for each model:

## ROUGE Score Inference

#### Model Size and Capacity:

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **ROUGE 1** | **ROUGE 2** | **ROUGE L** |
| GPT2 | 0.3696 | 0.1639 | 0.3128 |
| Llama 2 | 0.3912 | 0.1807 | 0.3158 |
| Gemma 2 | 0.4551 | 0.2290 | 0.3845 |
| Llama 3 | 0.4834 | 0.2632 | 0.4149 |

Table 5: ROUGE score comparison for the Q&A task

* + Larger models, such as Llama 3 (8 billion parameters) and Gemma 2 (2 billion parameters), per- formed better due to their ability to capture complex language patterns and generalize to topics not covered in the training set.
  + GPT2 Small (124 million parameters) struggled with generalization, particularly in multi-word sequences (ROUGE-2) and sentence-level coherence (ROUGE-L).

#### Architecture Advancements:

* + Llama 3, with its advanced architecture and optimizations, generated more coherent responses.
  + Gemma 2 showed improvement over GPT2 due to its more modern architecture, but Llama 3 outperformed in most metrics.
  + Llama 2 also performed well, but it did not reach the fine-tuning efficiency of Llama 3.

A diagram of a cell phone

AI-generated content may be incorrect.

Figure 7: Architecture Advancement

#### Generalization on unseen topics:

* + The ROUGE score differences can also be attributed to how well each model generalizes to topics not covered in the training data.
  + Llama 3 shows the best ability to generalize beyond the AI/ML domain, likely due to its large size, architectural sophistication, and fine-tuning efficiency.
  + Gemma 2 and Llama 2 also generalize well but not as effectively as Llama 3.
  + GPT-2 Small struggles the most with out-of-domain generalization, leading to significantly lower scores.
  + The following sample question is an interesting example, the training data does not contain anything related to RAG

A screenshot of a chat

AI-generated content may be incorrect.

Figure 8: Performance on Unseen topics

# Conclusion

It was a great learning experience fine tuning these transformer models to perform a specific task. The nuances of encoder-decoder vs decoder-only models and how it could influence the performance was a great learning experience.

There were engineering issues that these large models caused and we had to figure out ways to deal with them along the way. This task has increased our understanding multifold.