Retrieval-Augmented Generation (RAG) for Question Answering

Meghana Maringanti

Abstract

This paper explores the development and evaluation of a RAG system that retrieves context from PDF documents and generates answers using a transformer-based language model. It outlines the data sources, pre-processing steps, retrieval and generation methodologies, and evaluation procedures to understand the modelperformance

1 Introduction

The latest developments in Natural Language Processing (NLP) helped to create more robust question-answering (QA) systems. Classical models relied on retrieval or generation. However, the introduction of a hybrid approach such as Retrieval-Augmented Generation (RAG) combines both the components of Retrieval and Generation to get more accurate and efficient responses.

RAG contains a retriever component that finds relevant documents or context and a generator component that generates an answer using this retrieved information. This approach helps improve the accuracy of the responses and helps work with new data on which the model wasn't trained. It also helps reduce hallucinations.

In this paper, we will design a RAG-based question and answering system to understand and respond to questions, specifically using *The Adventures of Pinocchio* and *Tenali Ramakrishna stories* as data sources. I am storing both the Pure RAG - retrieved answers and LLM-generated responses and comparing them to my reference answers to assess the system's performance using metrics like BERT Score, ROUGE-L, and semantic similarity. And I am also doing manual evaluations, to understand the strengths and areas for improvement.

2 Methodology

2.1 Data Source

Data: The dataset consists of "Adventures of Pinocchio" and "Tenali Ramakrishna stories" files in the PDF format as data sources.

I utilized a list containing questions and another list containing reference answers to test and evaluate the effectiveness of the RAG and LLM systems. The dataset includes a variety of question types, like straightforward questions and complex, inference-based questions

Sample Questions and reference answers:

Question 1: Who is Tenali RamaKrishna? Reference answer: Tenali Ramakrishna was a Telugu poet, scholar, and advisor in the court of Sri Krishnadevaraya of the Vijayanagara Empire.

Question 2: How did Tenali RamaKrishna Died? Reference answer: Died in 1528 reportedly due to a snakebite

Question 3: What is TenaliRamaKrishna Famous for? Reference answer: Tenal RamaKrishna is famous for his wit and humour.

Question 4: What was Pinocchio made of? Reference answer: Wood

Question 5: What happens each time Pinocchio tells a lie? Reference answer: His nose grows

2.2 Pre-Processing:

I extracted the text from these documents, processed the text into chunks and embedded them using the *all-MiniLM-L6-v2* model from the SentenceTransformers library. Then the embeddings were indexed using the FAISS library, for fast and accurate search and retrieval.

2.2.1 Model:

This section provides an overview of the model utilized in the Pure RAG-based question-answering system.

all-MiniLM-L6-v2:

This model is a part of the Sentence-Transformers library and is specifically designed for creating sentence embeddings. This model was used to create vector representations of the text chunks extracted from PDF documents.

2.3 Retrieval-Augmented Generation (RAG)

Retrieval: For a set of questions I retrieved the top-k relevant documents and formatted them to get the answer

Generation: For the same set of questions based on the above-retrieved chunks genrated responses using the FLAN-T5-Base generation model.

2.3.1 Model:

This section provides an overview of the LLM model utilized in the generation of responses.

FLAN-T5-Base:

This model is a variant of the T5 (Text-to-Text Transfer Transformer) model. This model is used to generate well-structured answers based on the retrieved text chunks. It processes a question along with relevant context and generates an answer by predicting the next word in a sequence.

2.4 Evaluation Metrics

The following metrics were used to evaluate the generated answers:

1. BERT Score:

It measures semantic similarity using contextual embeddings. Precision, recall, and F1 scores were calculated for generated and reference answers.

2.ROUGE-L Score:

It measures the longest common subsequence overlap between generated and reference text.

3.Semantic Similarity:

Calculated using cosine similarity between embeddings of the generated and reference text.

Metrics were calculated for both RAG-generated answers and LLM-generated answers with retrieved context.

2.5 Tables and figures

Tables shows the evaluation metrics for the retrieval answers using the RAG method and the LLM-generated answers. Each cell represents the average scores observed for each metric.

Table 1: Semantic Similarity and ROUGE-L F1 Score Comparison for Pure RAG and RAG + LLM

Method	Semantic Similarity	ROUGE-L F1 Score
RAG	0.4051	0.1107
LLM	0.3913	0.229

Table 2: BERT Score Comparison for Pure RAG and RAG + LLM

Method	BERT Precision	BERT Recall	BERT F1 Score
RAG	0.3332	0.5382	0.406
LLM	0.5309	0.4639	0.4858

Analysis:

The results show that the Pure-RAG has a higher average score in the recall, which suggests the system retrieves more relevant content. However, LLM has higher scores in precision, ROUGE-L, and overall F1, which indicates that the integration of LLM helped in generating more accurate and appropriate responses.

However, the observed low average scores in all metrics for both sustems, suggest the need for further exploration through error analysis to understand the differences in performance.

2.6 Bootstrap Sampling for Robust Evaluation

Bootstrap sampling helps in estimating the variability and confidence intervals of evaluation metrics. For my analysis, I applied bootstrap sampling to a dataset of 27 questions, and 1,000 samples were generated to ensure a robust estimation of results using BERT scores. This was done to both retrieved answers(Pure RAG) and RAG + LLM-generated answers.

2.7 Tables and figures

Analysis: From the bootstrap analysis for the Pure RAG system, we observed the following points:

The mean precision score of 0.33 (CI: 0.29–0.38) shows that while the system retrieves relevant data, it often includes irrelevant details.

Metric	Mean Score	Confidence Interval Lower	Confidence Interval Upper
Precision	0.3319	0.2872	0.3789
Recall	0.5366	0.4776	0.5993
F1 Score	0.4046	0.3558	0.4567

Table 3: Bootstrap values for Pure RAG system's Precision, Recall, and F1 Score with confidence intervals.

The mean recall score of 0.54 (CI: 0.48–0.60) shows that though we can retrieve relevant some key information might be missing for complex queries.

The mean F1 score of 0.40 (CI: 0.36–0.46) reflects a balance between precision and recall, indicating a need for improvement.

Metric	Mean Score	Confidence Interval Lower	Confidence Interval Upper
Precision	0.5292	0.4561	0.6076
Recall	0.4637	0.3921	0.5389
F1 Score	0.4849	0.4184	0.5595

Table 4: Bootstrap values for RAG + LLM system's Precision, Recall, and F1 Score with confidence intervals.

Analysis:

From the bootstrap analysis for the RAG + LLM system we observed the following points:

The mean precision score of 0.53 (CI: 0.46–0.61) shows that the LLM system generally delivers relevant responses, with fewer unrelated details compared to RAG.

The mean recall score of 0.46 (CI: 0.39–0.54) shows that though the LLM system can generate relevant answers some key information might be missing for complex queries.

The mean F1 score of 0.48 (CI: 0.42–0.56) reflects a balance between precision and recall, indicating a need for improvement.

3 Error Analysis - Manual Evaluation

I compared the reference answers to retrieved answers and LLM-generated answers and used the labels below to categorize them.

3.1 Explanation of Labels:

1.Exact Match: The response is a perfect match or a paraphrase of the reference answer.

2.Partial Match: The response is partially correct but is missing key details or has some inaccuracies.

3.Match with Irrelevant Data: The answer includes correct information but is along with unnecessary details.

4.Related but Wrong: The response shows a connection to the question but fails to provide the correct answer.

5.Unrelated/Wrong: The response is completely incorrect or not related to the question.

3.2 Summary of labels:

Label	RAG Count	LLM Count
Unrelated/Wrong	13	12
Related but Wrong	7	7
Match with Irrelevant Data	6	0
Exact Match	1	5
Partial Match	0	3

Table 5: Label distribution for RAG and LLM systems.

3.3 Analysis

Based on the analysis, both pure RAG and RAG + LLM systems show similar trends.

For straightforward questions, such as "Who is <name>?" or "Who is the author of <story-name>?", the Pure RAG system had a higher count of Match with Irrelevant Data while LLM-generated responses were Exact or Partial matches. This indicates that the integration of LLMs enhances response accuracy by filtering out irrelevant data.

However, when faced with more complex, inference-based questions, both systems struggled to provide accurate answers, resulting in an increase in Related but Wrong and Unrelated/Wrong responses. It is mainly due to the inaccurate retrieval of documents. This highlights the need for efficient retrieval process.

One Question and answer pair we need to take into account is Q:"How many chapters are there in Adventures of Pinocchio"? Reference: 36 chapters LLM response: 24. I labelled it as Related but wrong. Because it looks like a mistake in counting. But when we analyse the context it is taking into account we can see "Chapter 24" in the context so it might mean that LLM simply extracted the 24 if that is the case then it should be categorised under Unrelated/Wrong.

By comparing the answers generated by pure RAG and RAG + LLM systems, we can observe

that these issues primarily arise from the retrieval process. This highlights the need to fine-tune the retrieval process to better capture relevant documents. Furthermore, in cases where the system successfully retrieved relevant documents, the LLM-generated answers were still incorrect. This suggests that even with the right context, sometimes the systems failed to draw the necessary conclusions to answer the questions. Therefore, there is a clear need to train the model to improve its ability to answer complex and inference-based questions.

In conclusion, we must enhance the retrieval process and fine-tune the LLM model to generate accurate responses.

4 Conclusion

This study demonstrates that RAG can effectively combine retrieval and generation to enhance question-answering systems. The metrics and the manual evaluation suggest that we need to adjust and improve the retrieval process to enhance the relevance, accuracy and quality of the responses.

Limitations

The system's performance depends heavily on pre-trained models like all-MiniLM-L6-v2 and FLAN-T5-Base so it might not work efficiently on the tasks they are not trained on. I observed that FAISS indexing does not always retrieve the most relevant information so it will impact the accuracy and quality of the generated answers. We are also specifying a limit on the output length which can truncate essential input, which can lead to less comprehensive answers.

References

- SentenceTransformers documentation: https://www.sbert.net/
- PyPDF2 documentation: https://pypi.org/project/PyPDF2/