

# Enhancing Large Language Model Performance Using Retrieval-Augmented Generation: A Comprehensive Evaluation with RAGAS and GEval

Omar Alexander Hoang  
The University of Texas at El Paso  
[oahoang@miners.utep.edu](mailto:oahoang@miners.utep.edu)

## Abstract

In this research project, I explored the fine-tuning of a large language model (LLM) using Retrieval-Augmented Generation (RAG), with evaluations using RAGAS and GEval. Our objective is to enhance the performance of the LLM by optimizing its parameters through a series of five fine-tuning runs. RAG, a technique that integrates retrieved documents into the generation process, is used to improve the correctness and relevancy of the model's outputs. To assess the impact of parameter adjustments of the models being used, I used RAGAS and G-Eval, which have frameworks for evaluation. RAGAS focuses on the retrieval aspect, ensuring that the documents retrieved are beneficial for the generation of output. GEval on the other hand, offers comprehensive metrics to evaluate the generative performance of the model. By conducting five iterative fine-tuning runs and evaluating each iteration with these tools, I aim to achieve significant improvements in the model's ability to generate appropriate and accurate responses. The findings from this study are expected to contribute valuable insights into the optimization of LLMs using RAG and how RAGAS and GEval metrics have an impact in the fine-tuning process.

## Background

The rapid advancement of large language models (LLMs) has transformed the field of natural language processing (NLP), developing sophisticated applications in text generation, summarization, and conversational agents. However, the performance of these models can vary significantly depending on the quality and relevance of the data they are trained on. As stated in Kumar's, Gattani's, and Singh's medical research using a LLM, the usage of inaccurate and incomplete datasets can hurt the LLM's intended objectives[1]. To address this, fine-tuning LLMs using advanced techniques like Retrieval-Augmented Generation (RAG) has become an essential area of research.

RAG integrates external knowledge retrieval into the text generation process, enhancing the model's ability to produce contextually relevant and accurate responses. This approach mitigates the limitations of traditional LLMs, which solely rely on their pre-trained knowledge. By

incorporating retrieved documents, RAG allows the model to access and utilize up-to-date information, leading to more precise and aware outputs.

In the context of fine-tuning LLMs, evaluating the impact of parameter changes is crucial for optimizing model performance. This research uses RAGAS and G-Eval as the primary evaluation frameworks. RAGAS is designed to assess the retrieval component of RAG, ensuring that the retrieved documents are beneficial for the generation phase. G-Eval aims to evaluate models in a way that closely mirrors real-world usage, by simulating human-like tasks and queries and providing a better understanding of how the model performs in practical applications. Compared to RAGAS, G-Eval aims to simulate real-world usage scenarios dynamically to understand how models perform in practical applications.

The iterative fine-tuning process involves adjusting the model's parameters and evaluating its performance over five runs. Each iteration is assessed using RAGAS and GEval to identify the most effective parameter configurations. This approach aims to achieve significant improvements in the model's ability to generate high-quality appropriate responses.

By exploring the fine-tuning of LLMs with RAG and utilizing RAGAS and GEval for evaluation, this research seeks to contribute to the optimization methodologies for LLMs and demonstrate how these evaluation tools enhance model performance. The knowledge gained from this study is expected to advance the understanding of fine-tuning techniques and their practical applications in NLP.

## **Methods**

For this research, I utilized the “Llama\_2\_7b\_chroma” model as the base large language model (LLM) for fine-tuning. This model is from Meta (formerly Facebook). ‘7B’ stands for 7 billion parameters, making it one of the larger variants of the LLaMA model series. The dataset utilized for fine-tuning was a text file containing the complete first Game of Thrones book (GOT1.txt). This file served as the primary resource for training, providing diverse structures representative of natural dialogue styles.

For the retrieval component of RAG, it was essential to divide the .txt file into manageable chunks. The “Mini LM L6 v2” model was used for this task, which efficiently chunked the text into chunks suitable for retrieval and different process pool batch sizes for computation. It maps sentences & paragraphs to a 384 dimensional dense vector space and can be used for tasks like clustering or semantic search. Larger batch sizes can lead to more efficient use of the GPU, reducing the overhead associated with each batch and speeding up the overall computation. Smaller batch sizes can help with convergence during training, as they introduce more noise into the gradient updates, which can act as a form of regularization.

The parameters that were changed for each run were temperature, max new tokens, and repetition penalty for the “Llama\_2\_7b\_chroma” model. Temperature 'randomness' of outputs, 0.0 is the min (more deterministic) and 1.0 the max (more random). Max new tokens is the

number of tokens to generate in the output. Repetition penalty helps to reduce repetition in the generated text. For the “Mini LM L6 v2” model the batch size and text chunk size were modified.

The inputs that were used for the Llama models was a list of questions that pertained to the .txt file. These questions were a set of easy and hard questions to determine if the model can handle any types of questions that it would receive. The easy set of questions were very basic that entailed mostly who, what, and where inquiries. The harder questions were more in depth in what was going on in the story, more of what certain things represent within the book.

Once the model gave the answers as outputs from the different types of questions, the data was stored in .csv files in order to be evaluated. RAGAS was used via OpenAI API and the metric utilized was “answer correctness”, a high score (close to 1) signifies that the generated answers are highly accurate compared to the ground truth, which was a list of correct answers that correlated with the questions given to the Llama model. Lower scores suggest that the answers may contain inaccuracies. G-Eval also was implemented with a OpenAI API call, the metrics used were “answer correctness” and “answer relevancy”. “Answer relevancy” evaluates how pertinent and appropriate the responses are, reflecting the model's ability to generate responses that are not only correct but also meaningful. Both metrics use a scoring range from 0 to 1, with higher scores indicating better performance. With G-Eval there was no need for a ground truth set as the model itself would get the correct answer for the questions.

This framework was used during a test and experimental run. Where the test runs acted as a way to get familiar with the models and tools being used. The experimental runs were for the actual evaluating and modifying of parameters of the models that came into play. During the experimental run part of the research, the model was given a different data set of cyber security questions and observations were made to see how the model would do given no cyber security documentation to go off.

## **Test Run Procedure and Results**

Before starting on any of the experiments the model, the tools to parse the data, and the evaluation models were to be fully understood in order to have concise experimental runs. These four ‘test’ runs served as a way to structure how the five ‘experimental’ runs would be fine tuning the LLM using RAG.

Since the data set was going to be a .txt file of a book, the book's text needed to be tokenized into chunks in a vector database using ChromaDB. This database would be used for managing and querying vectors. To prepare the text for the RAG process, the Mini LM L6 v2 model was used for text chunking, which effectively divided the book into smaller, coherent segments. The runs had various chunk sizes from the first to the last respectively: 50, 100, 50, 100. This approach was used in a way another research team implemented RAG for their LLM on a Brazilian Harry Potter book. They used much larger chunks that ranged in the 1,000s[2], so I wanted to see how the retrieval would be if I chunked the book in smaller consistent sizes.

The thought process of keeping the chunks small was so that I can mainly see the changes when adjusting the parameters of the Llama model. Once the database was in order I then formulated a set of easy and hard questions for each run as seen in figure 1 & 2.

Easy Questions:

Who is the youngest Lannister sibling?  
What is the name of the ancestral home of House Tully?  
Who is the father of Daenerys Targaryen?  
What is the name of Jon Snow's direwolf?  
Who is known as the 'Queen of Thorns'?  
What is the name of the Wall made of ice?  
Who is the king that Daenerys Targaryen marries in the first season?  
What is the name of the wolf pup that Arya Stark adopts?  
Who is the youngest Stark son?  
What is the name of the continent where the Dothraki live?  
Who is the ruler of the Vale of Arryn?  
What is the name of Tyrion Lannister's lover from Essos?  
Who trains Arya Stark in Braavos?

Fig. 1 Easy Questions for GoT dataset

Hard Questions:

How does the political alliance between House Tyrell and House Lannister impact the War of the Five Kings?  
What are the implications of Tyrion Lannister's marriage to Sansa Stark?  
How does the relationship between Theon Greyjoy and Ramsay Bolton affect Theon's character development?  
What role do the Faceless Men play in Arya Stark's transformation throughout the series?  
How does Cersei Lannister's Walk of atonement influence her subsequent actions and the power dynamics in King's Landing?  
What is the significance of the pact made between the Children of the Forest and the First Men?  
How does the discovery of dragonglass impact the fight against the White Walkers?  
What are the consequences of the Battle of the Blackwater for the key characters involved?  
How does the storyline of the Greyjoy Rebellion affect the larger narrative of 'A Game of Thrones'?  
What is the importance of the prophecy about Cersei's children and how does it shape her actions?

Fig 2. Hard Questions for GoT dataset

A different set of questions were asked for each run with the parameters of the Llama model being modified. I noticed that greater the temperature (more random) with less of a penalty gave better results as opposed to more concise and penalized model setup. In the first run the temperature was set to 1, being the most random, the max tokens at 500 and the repetition penalty at 1.1. Compared to the other three runs, the first run's RAGAS score was far better as seen in figures 3 through 6.

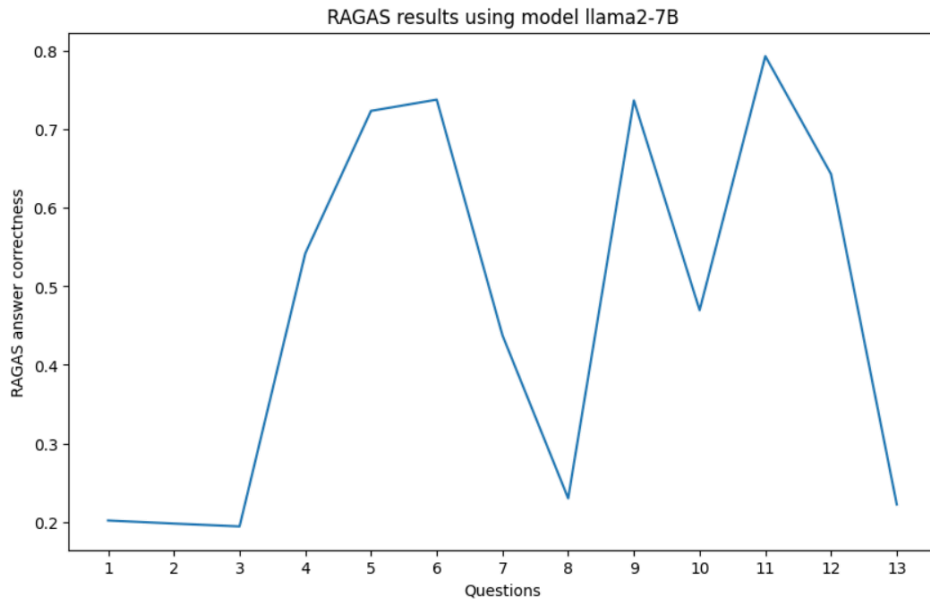


Fig 3. Run1 easy question scores for GoT dataset

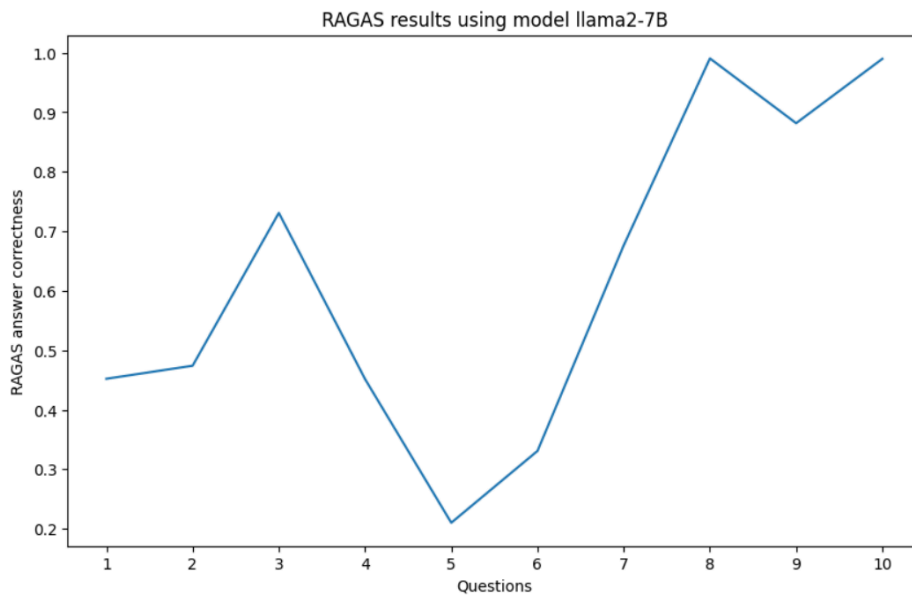


Fig 4. Run1 hard questions for GoT dataset

In Figure 3 & Figure 4, Run 1 had its models configured to these hyper-parameters for the set easy and hard questions:

- MiniLM L6:
  - Batch size = 32
- Llama 7B:
  - Temperature = 1.0
  - Max new tokens = 500
  - Repetition penalty = 1.1
- Text splitter:
  - Chunk size: 50

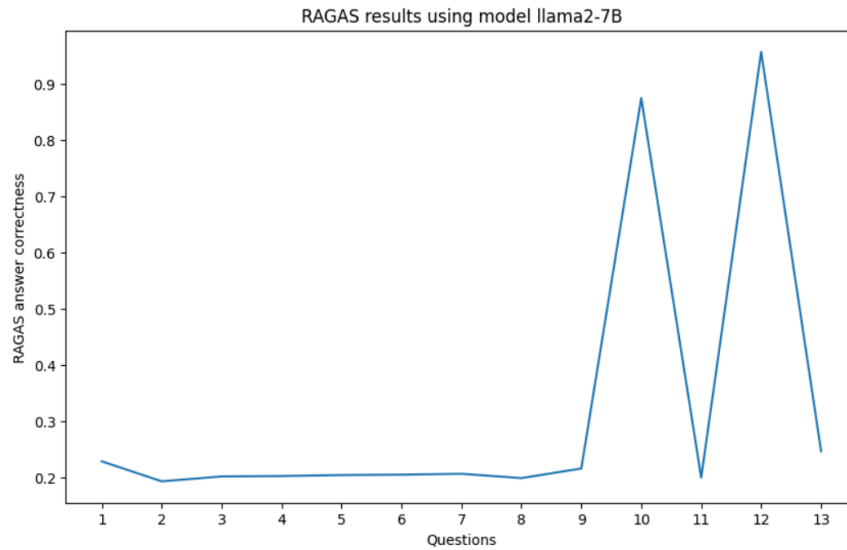


Fig 5. Run4 easy questions for GoT dataset

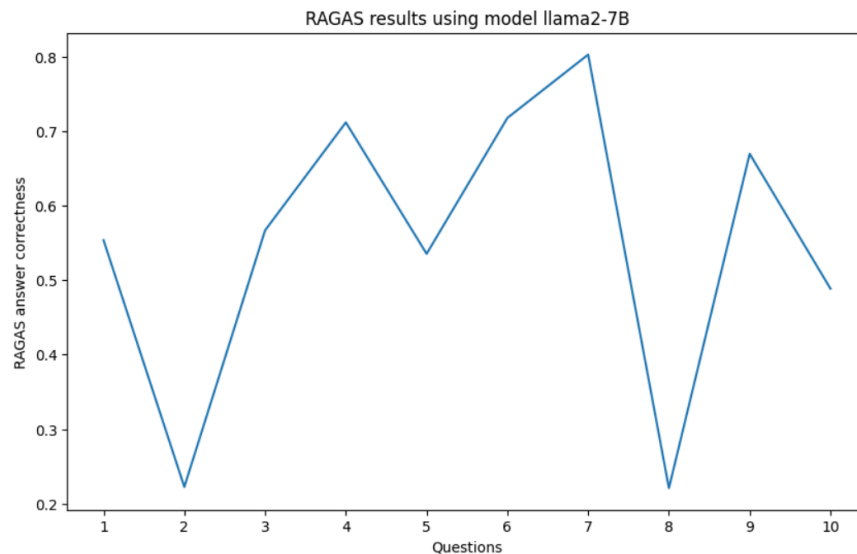


Fig 6 Run4 hard questions for GoT dataset

In Figure 5 & Figure 6, Run 4 had its models configured to these hyper-parameters for the set easy and hard questions:

- MiniLM L6:
  - Batch size = 32
- Llama 7B:
  - Temperature = 1.0
  - Max new tokens = 500
  - Repetition penalty = 1.1
- Text splitter:
  - Chunk size: 50

Seeing that the first run had the best scores and the rest of the runs had worst scores when changing the parameters led me to believe that I had to be doing something wrong when

changing the questions. Upon further investigation I was making the LLM output these large responses (500, 300, 200, 250 max tokens) which was affecting the RAGAS score. Going forward to the experimental runs I made sure to combine the easy and hard questions together, let the LLM output smaller responses, and outsourced another evaluation tool to use with RAGAS.

## Experimental Run Procedure and Results

For the experimental runs some changes were added to ensure the scores were evaluated correctly this time. The max tokens for the model's output was reduced to a smaller range in order to not have scores affected by unnecessary information. The introduction of the G-Eval model was presented in order to have another metric evaluation tool for more insight. In Microsoft Cognitive Services research, the team of researchers used G-Eval's framework ability to generate detailed evaluation steps and refine scores based on token probabilities contributed to its effectiveness in human-like feedback responses that it utilizes[3].

The procedure was the same as the test runs, but this time after getting the scores for the Game of Thrones questions. Another set of questions were implemented to see how the scores would compare as the LLM had no database knowledge of these questions. These sets of questions pertained to cybersecurity, and the model would still have the Game of Thrones .txt file as the vector database. Below are the security questions given to the model in figure 7:

### Talos questions:

What is Netgear RAX30 JSON Parsing `getblockschedule()` stack-based buffer overflow vulnerability?

What is NVIDIA D3D10 Driver Shader Functionality out-of-bounds read vulnerability?

What is Google Chrome Video Encoder Metrics denial of service vulnerability?

What is llama.cpp GGUF library `header.n_kv` heap-based buffer overflow vulnerability?

What is llama.cpp GGUF library `header.n_tensors` heap-based buffer overflow vulnerability?

What is llama.cpp GGUF library GGUF\_TYPE\_ARRAY/GGUF\_TYPE\_STRING parsing heap-based buffer overflow vulnerability?

What is llama.cpp GGUF library `info-&gt;ne` heap-based buffer overflow vulnerability?

What is llama.cpp GGUF library `gguf_fread_str` heap-based buffer overflow vulnerability?

What is Weston Embedded `uC-TCP-IP` IP header loopback parsing double-free vulnerability?

What is The Biosig Project `libbiosig BrainVisionMarker` Parsing Out-of-bounds Write vulnerability?

What is Weston Embedded `uC-TCP-IP ICMP/ICMPv6` parsing denial of service vulnerabilities?

What is Imaging Data Commons `libdicom` DICOM File Meta Information Parsing Use-After-Free vulnerabilities?

What is The Biosig Project `libbiosig BrainVision` Header Parsing double-free vulnerability?

What is Weston Embedded `uC-HTTP` HTTP Server heap-based buffer overflow vulnerability?

What is The Biosig Project `libbiosig sopen FAMOS read NULL calloc` out-of-bounds write vulnerability?

Fig 7. Talos dataset questions

From the scores averages from the five runs showed a small difference in the models performance when having a knowledge base that it can quickly access. The figures below are

the RAGAS and G-Eval score averages for each run and the parameters implemented for the model in that run.

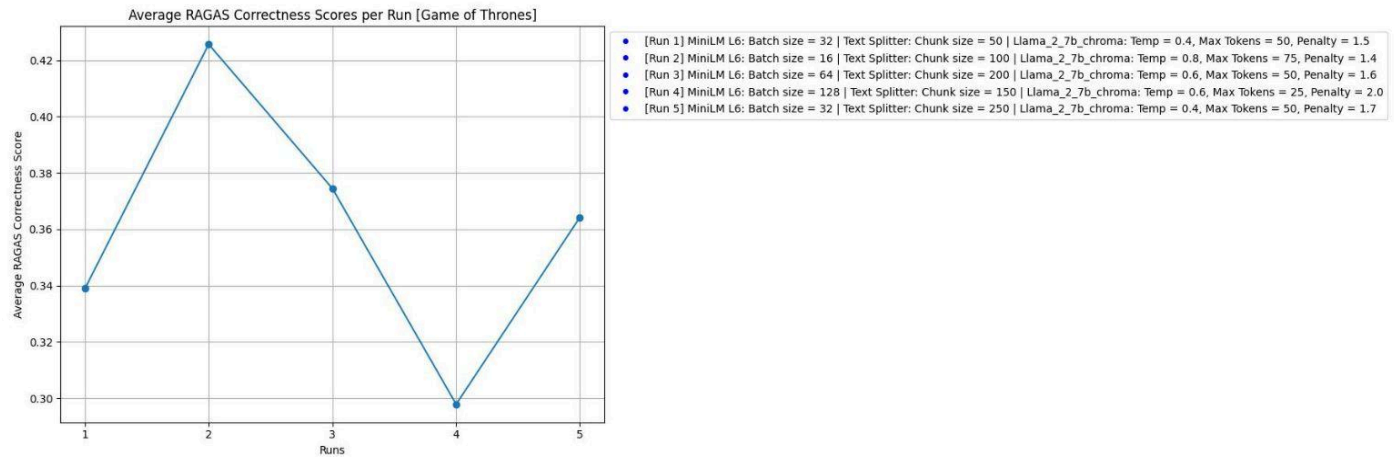


Fig 8. RAGAS Scores for GOT questions

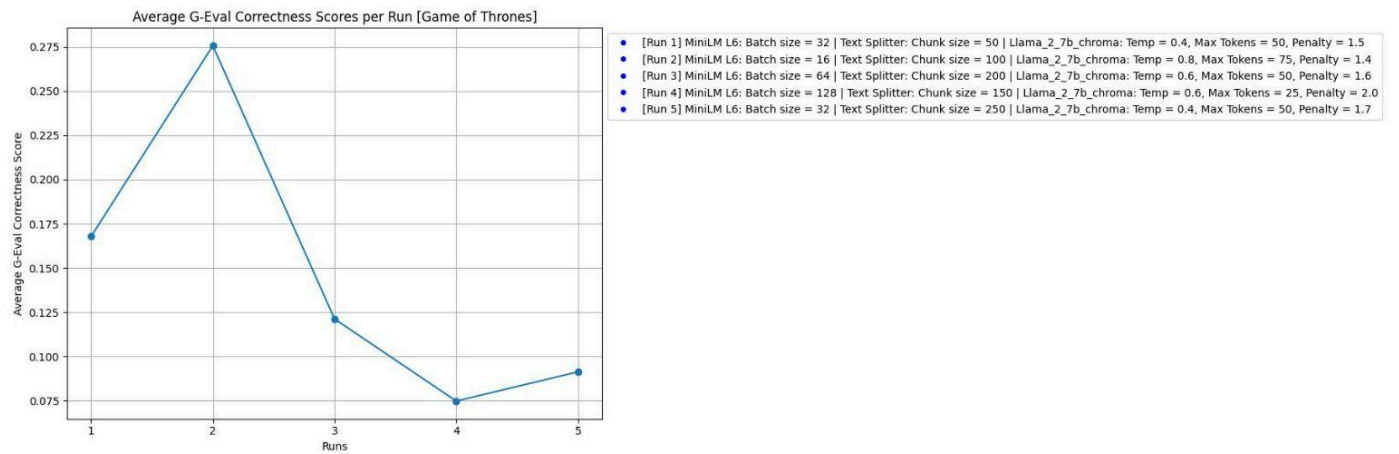


Fig 9. G-Eval Scores for GOT questions

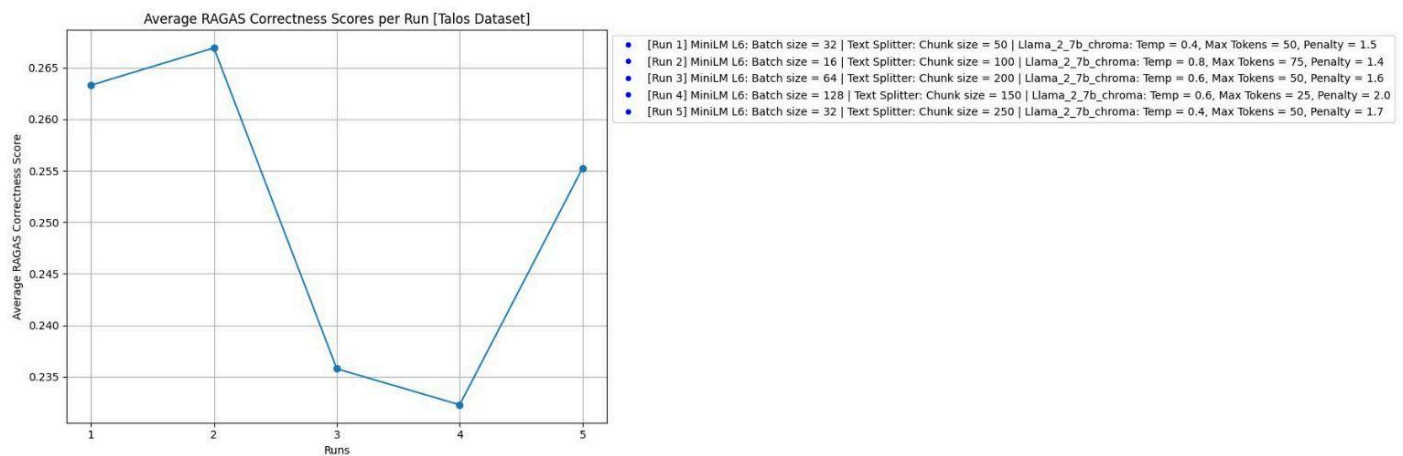


Fig 10. RAGAS Scores for Talos questions



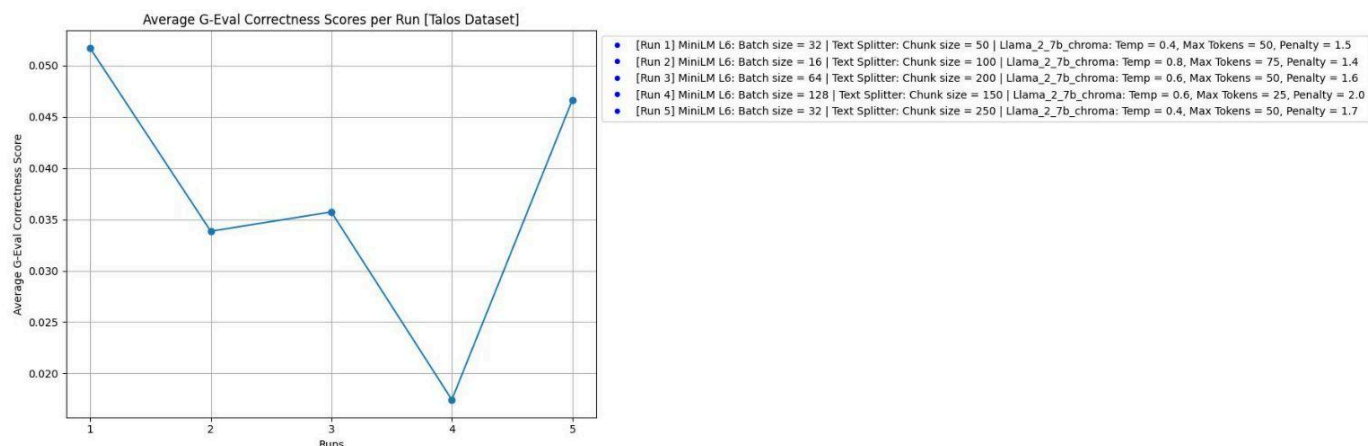


Fig 11. G-Eval Scores for Talos Questions.

Run 2 had the best results among all the experimental runs. This success can be seen by its combination of parameters that balanced the randomness and control of the generated outputs. The MiniLM L6 model was set to a batch size of 16, which helped manage the computational resources. The text splitter was set to chunk sizes of 100 with no overlap, providing a balance of manageable text segment sizes to be embedded. The Llama 7B model in Run 2 was fine-tuned with a temperature of 0.8, allowing for more creative and varied responses without losing relevance. The max new tokens parameter was set to 75, making the response detailed but not with unnecessary information. Additionally, a repetition penalty of 1.4 was used, reducing redundancy in the generated text. This combination of a higher temperature, appropriate chunk size, and balanced repetition penalty resulted in more accurate outputs, outperforming the other runs which either had too much deterministic settings or less optimal text chunking.

## Conclusion

In conclusion, this research project demonstrated the effectiveness of fine-tuning a large language model (LLM) using Retrieval-Augmented Generation (RAG) and evaluated the performance using RAGAS and G-Eval metrics. The iterative process of adjusting parameters across five experimental runs highlighted the importance of balancing randomness and control in generating outputs. Run 2 had the best performing configuration, showing how optimal parameter settings such as batch size, chunk size, temperature, max new tokens, and repetition penalty can significantly enhance the model's ability to produce accurate and relevant responses. The use of both the Game of Thrones and Talos datasets provided a comprehensive evaluation, illustrating the model's adaptability to different types of queries. An improvement in software could have greatly helped with this research, as I was utilizing Google Colab's free GPU service but had heavy constraints when being used. This research contributes to the growing knowledge on LLM optimization and the potential of RAG in enhancing the contextual relevance and accuracy of language models in various NLP applications.

## Acknowledgements


I would like to thank The University of Texas at El Paso's Computer Science department for allowing me to take this class to further expand my knowledge of the machine learning side of computer science and research. I would also like to thank Dr. Piplai for accepting me as a student for this class. His mentoring and shared experiences throughout this semester have greatly improved my skills as a problem solver. I am now better equipped to apply these skills in real-world scenarios, and I look forward to utilizing this newfound expertise in my future endeavors. His influence has been instrumental in my growth, and I am grateful for the valuable insights and practical knowledge he has shared.

## References

1. Kumar, Rohit, Dr Ram Krishna Gattani, and Kavita Singh. "Enhancing Medical History Collection using LLMs." *Proceedings of the 2024 Australasian Computer Science Week*. 2024. 140-143.
2. Es, Shahul, et al. "Ragas: Automated evaluation of retrieval augmented generation." arXiv preprint arXiv:2309.15217 (2023).
3. Liu, Yang, et al. "G-eval: Nlg evaluation using gpt-4 with better human alignment." *arXiv preprint arXiv:2303.16634* (2023).

## Supplementary Files

Below is a link to all downloadable files that contain the datasets used, results from the test runs, results from the experimental runs, and evaluation results:

 [CS4371 Summer24 Source Code & Results](#)