Fine-Tuning Large Language Model (LLM) to Answer Basic Questions for Prospective New Students at Syiah Kuala University Using the Retrieval-Augmented Generation (RAG) Method

Hary Rachmat   
*Department of Informatics*   
*Universitas Syiah Kuala*

Aceh, Indonesia  
haryrachmat10@gmail.com

Hammam Riza  
*Research Center for Artificial Intelligence and Cyber Security  
National Research and Innovation Agency (BRIN)*

Jakarta, Indonesia  
hammam.riza@brin.go.id

Taufik Fuadi Abidin  
*Department of Informatics   
Universitas Syiah Kuala*

Aceh, Indonesia  
taufik.abidin@usk.ac.id

*Abstract*—USK Mistral 7B is a large language model designed to answer basic admission questions at Universitas Syiah Kuala (USK). The model was fine-tuned using the open-source model of Mistral 7B using collected data from admissions and lectures at the university. The QLoRA and RAG techniques were used to train the model and retrieve relevant information from external data sources. The results were evaluated using the ROUGE score. Responses were generated with a score of >0.5 on ten out of 46 questions with the RAG method, and testing with the fine-tuning method was carried out on 20 questions and resulted in responses with a score of 1.0 from all questions asked. The performance of USK Mistral 7B shows its potential as an effective tool in helping students querying information about admission and lectures at USK.

Keywords— Large Language Model, Fine-tuning, RAG

# Introduction

With the rapid advancement of artificial intelligence (AI), large language models (LLMs) have revolutionized natural language processing (NLP), thus allowing computers to interact with text and language. One example of an LLM is Generative Pretrained Transformer 4 (GPT-4), which has been extensively trained on large amounts of text data, allowing GPT-4 to perform text analysis across multiple domains. LLM can be applied to various fields such as health, education, law, and other fields by conducting pre-training using large amounts of data in these fields.

ChatGPT, one of the most popular chatbot applications today, has been the subject of various studies exploring its capabilities [1]. These studies have demonstrated its potential in the medical field for aiding clinical documentation [2], in the banking sector for classifying texts [3] and in the field of law for determining potential violations. These examples highlight the diverse and practical applications of large language models like ChatGPT [4].

In addition to closed-source LLMs such as GPT-4, some studies use open-source ones like those conducted by Huang et al.[5] by adapting Llama's LLM to the legal domain to assist lawyers in preparing technical reports. Bhatti et al [6] fine-tuning the LLM Llama 70B named "SM70" to address a wide range of complex medical questions and clinical decision-making. Zhao et al [7] fine-tuning the LLM Llama 7B named "Ophtha-LLaMA2" to help diagnose eye diseases that will provide decision support for doctors. Barandoni et al [8] conducted a comparative analysis of LLMs to extract the needs of travel customers from TripAdvisor posts by Leveraging a variety of models, including open-source models such as Mistral 7B and closedsource models such as GPT-4 and Gemini. The results highlight the efficacy of open-source LLMs, especially the Mistral 7B, in achieving comparable performance to closed-source models. Research on chatbots as virtual assistants has been conducted by Jonatan & Igor [9] to improve the efficiency of service to customers. The research that has been presented has shown the potential of LLM to help work in various fields.

Based on data.usk.ac.id website [10] the number of students registering for Syiah Kuala University (USK) is increasing yearly. To provide information such as registration details, tuition fees, and other information, USK provides a website where students can find information about matters related to the university. When there is information that is not yet available on the website, prospective students can ask questions directly through social media, such as direct messages (DM) through the Instagram application, and also meet university staff directly at the Integrated Service Unit (ULT) or Public Relations (HUMAS) section. This study will examine the application of LLM as a virtual assistant as a chatbot to provide interactive information related to academic administration at USK.

# Research Methods

## Dataset Conditions

The dataset contains 231 data samples related to information about the lecture system and new student admissions at USK. The data is then preprocessed from the dataset obtained for fine-tuning the Mistral 7B model here using as many as 20 Q&A datasets, converted to Alpaca style format, and then stored in a format with .csv extension. A total of 231 datasets are also created in the form of statements and saved in a format with extensions .pdf Extensions are then saved into the huggingface repository, which will later be used in the next stage, namely the fine-tuning model [11]. Data in .pdf format will be used in the RAG method [12] to manage the data more efficiently.

## Quantization and Model Training

In Fig. 1, The fine-tuning process with the Mistral 7B model is divided into three stages, which can be seen as follows:

1. *Model Quantization and Data Preparation*



Fig. Fine-tuning *process* on the Mistral Model 7B

The fine-tuning process on the Mistral 7B model is trained with the help of the transformer library to quantize the Mistral 7B, and the BitsAndBytesConfig library interface is used to quantize the Mistral 7B USK model. The purpose of this BitsAndBytes library is to reduce the precision value of the floating point in the weight of the USK Mistral Model 7B, which is processed by quantization from precision with a high value to precision with a lower value. In this process, the Mistral 7B USK model weights are converted into int4 format through a quantization layer and stored in the GPU. The primary computing process is performed on the CUDA, which will reduce memory usage and improve the model's efficiency. This process makes it possible to fine-tune larger LLMs on consumer-grade GPUs. The fine-tuned model is stored in the hugging face repository*.*

1. *Pre-trained Model*

Before the training, an analysis of well-known LLMs from the current LLM pool was conducted to select the most suitable model for this research. Evaluation of LLMs mainly considers several key aspects such as adaptability to specific domains, compatibility in academic standards, bilingual language skills, availability of models on open source, efficiency in parameters, cost, and licensing considerations on the model. This consideration also looks at several evaluation metrics tested across various benchmarks.

1. *Parameter-Efficient Fine-Tuning (PEFT)*

Considering dataset availability, the goal of reducing the cost of the training process, and the potential for failure risk, the researchers chose the Progressive Layer Freezing and Fine-tuning (PEFT) method [13] to refine the Mistral 7B model. PEFT selectively reduces a small number of parameters in the additive model. In this way, model training can significantly reduce computational and memory storage costs. It will enable efficient adaptation of pre-trained LLMs in various application domains.

In this study, a low-level adjustment method (QLoRA) was explicitly used [14] to improve large language models. Lora method [15] involves freezing pre-trained weights on the original model and creating a new version on the matrix with lower rank values for layers and query values. This lower-rated matrix has values on significantly fewer parameters than the original model, allowing for adjustment of memory usage on GPUs with smaller storage sizes. The advantage of this method can be seen in the ability of many LoRA adapters to reuse native LLMs, thereby reducing the overall memory usage required when providing answer text in use cases in tasks on specific domains and in various cases to be applied. Unlike LoRA, the QLoRA method represents an iteration of the LoRA method that will save more memory storage. The QLoRA method takes LoRA one step further by measuring the value or weight on a LoRA adapter with a smaller matrix value to a lower precision value (e.g., the model weight value becomes 4-bit, and not the value on a model with an initial value of 8-bit). This approach can reduce the memory size and requirements of model storage. In the QLoRA method, the trained model is loaded into GPU memory with a quantized 4-bit weight value, in contrast to the 8-bit model used in the LoRA method. Despite the decrease in bit precision values, the QLoRA method can maintain a level of effectiveness comparable to that of the LoRA method. So, this study uses a parameter-efficient improvement method with QLoRA. The fine-tuning method for LLM training uses the Unsloth library and Huggingface's TRL library [16]. The Unsloth library can make finetuning LLMs 2x faster.

The hyperparameter settings used in the fine-tuning process using the Unsloth library and Huggingface's TRL library are shown in the following table.

Table 1 shows the hyperparameter settings used in the fine-tuning process using the Unsloth library and Huggingface's TRL library. It shows the configuration for creating a FastLanguageModel.from\_pretrained instance using the unsloth component, with specific configurations such as maximum sequence length, data type, and 4-bit loading [17].

Table FastLanguageModel.from\_pretrained Configuration

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| model\_name | mistralai/Mistral-7B-v0.1 |
| max\_seq\_length | 2048 |
| dtype | None |
| load\_in\_4bit | True |

In Table 2, the FastLanguageModel object provides attributes get\_peft\_model, where users can configure various parameters for customization, such as the number of attention heads, target module, dropout rate, LoRa alpha, and more. Using checkpointing gradients and other advanced techniques demonstrates unslotting's ability to optimize model performance [17].

Table FastLanguageModel.get\_peft\_model Configuration

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| r | 16 |
| target\_modules | ["q\_proj", "k\_proj", "v\_proj", "o\_proj", "gate\_proj", "up\_proj", "down\_proj",], |
| lora\_alpha | 16 |
| lora\_dropout | 0 |
| bias | None |
| use\_gradient\_checkpointing | "unsloth" |
| random\_state | 3407 |
| use\_rslora | False |
| loftq\_config | None |

The next step in Tables 3 and 4 is to initialize the Supervised Fine-tuning Trainer, which helps the fine-tuning process. This involves initializing the model, the dataset to be refined, the tokenizer, and all necessary Training Arguments (learning speed, maximum steps, weight reduction, optimization, etc.) [17].

Table *SFTTrainer* Configuration

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| model | model |
| tokenizer | tokenizer |
| train\_dataset | dataset |
| dataset\_text\_field | "text" |
| max\_seq\_length | 2048 |
| dataset\_num\_proc | 2 |
| packing | False |

Table *TrainingArguments* Configuration

|  |  |
| --- | --- |
| **Parameter** | **Value** |
| per\_device\_train\_batch\_size | 2 |
| gradient\_accumulation\_steps | 4 |
| warmup\_steps | 5 |
| max\_steps | 60 |
| learning\_rate | 2e-4 |
| fp16 | not is\_bfloat16\_supported() |
| bf16 | is\_bfloat16\_supported() |
| logging\_steps | 1 |
| optim | "adamw\_8bit" |
| weight\_decay | 0.01 |
| lr\_scheduler\_type | "linear" |
| seed | 3407 |
| output\_dir | "outputs" |

1. *USK Mistral 7B Approach Model with RAG Method*

At this stage, an approach was carried out on the Mistral 7B USK using the RAG method [18], This method aims to overcome the limitations of generative AI when it requires information that is outside the Mistral 7B USK training corpus so that this method will avoid Mistral 7B USK, which will produce inaccurate text, hallucinations, or distortions when providing answers to the given questions. in the RAG method, the data used is an external document that contains information on the USK academic system and is stored in an extended format .pdf. The information data set is then embedded to convert text into vectors stored in a database vector. The database vector used in this study is FAISS. The model to be used at this stage is a model that has been quantized into GPT-Generated Unified Format (GGUF) [19] so that it is possible to use the CPU when running the LLM by moving some of its layers to the GPU so that it can accelerate the model's performance in generating text. The query process on the model in retrieving the most relevant context of the user command with the RAG method is shown in Fig. 2.



Fig. . The RAG pipeline, during the query phase, takes the most relevant context from the user's commands, passing them to the Large Language Model [20]

1. *Model Performance Evaluation*

Model Performance Evaluation The metric used to evaluate the model's performance in this study is the Rouge Metric. Rouge metrics are used to evaluate models on NLP tasks so that they can compare the text summaries generated by the model with the summaries in the references.

- Recall-Oriented Understudy for Gisting Evaluation (ROUGE)

The Rouge metric is an evaluation metric used in NLP tasks to compare computer-generated text summaries with reference summaries (generated by humans) [21]. Rouge is mainly used to evaluate text summarization tasks. The value of the Rouge metric ranges from 0 to 1. 1 is the highest score, indicating that the computer-generated summary and the reference summary have a high degree of similarity. Rouge-1, Rouge-2, and Rouge-L compare two summaries with different details [22].

* + - 1. Rouge-1

Rouge-1 measures the accuracy of unigrams (single words) that overlap between the generated text and the reference text (Man-made).

|  |  |
| --- | --- |
|  | (1) |
|  | (2) |
|  | (3) |

* + - 1. Rouge-2

Rouge-2 measures the accuracy of overlapping bigrams between the generated text and the reference text (man-made). The rouge-2 formula is the same as rouge-1, but the words used are bigrams, not unigrams. Bigrams compensate for the problem of the position of the word Rouge-1 to some extent.

* + - 1. Rouge-L

Unlike Rouge-1 and Rouge-2, Rouge-L does not look into unigrams or bigrams but conforms to LCS (Longest Common Subsequence) or the longest sequence of words in a human-generated reference and text.

|  |  |
| --- | --- |
|  | (4) |
|  | (5) |
|  | (6) |

# Results and Discussion

The output produced in this study is a web-based chatbot application utilizing the LLM Mistral 7B, which will be used as an alternative to help prospective new students get information at Syiah Kuala University (USK). This chatbot was developed using the USK dataset, which can summarize LLM texts to generate information about the lecture system and new student admissions at USK. Several stages are carried out in utilizing and developing LLM, namely by collecting data related to lecture system information and new student admissions at USK. The data will be collected in the preprocessing stage by converting the raw data into .csv and .pdf formats. After the preprocessing stage, finetuning was carried out on the Mistral 7B model, and also, to make it easier to manage data, RAG was carried out with the concept of embeddings. The purpose of the RAGis is to overcome generative AI's limitations because whenever a question requires information outside the LLM training corpus, it will result in hallucinations, inaccuracies, or distortions in the generated text. The next stage is to test and evaluate the generated text using the ROUGE method to compare the text generated by the model with the summary of the provided references. The last stage is creating a UI or web interface accessible to users.

## Testing Results and Evaluation of Inference Results

At this stage, the chatbot was tested by asking questions. The resulting text inference results were then calculated using the Recall-Oriented Understudy for Gisting Evaluation (ROUGE) score method. Using the ROUGE metric, responses from the chatbot can be evaluated to see quantitative similarities between the reference and the answers generated by the chatbot. The results of the ROUGE calculation score on the model are shown in Table 5.

Table Score value of ROUGE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Method** | **Number of**  **Questions** | **ROUGE Score** | | |
| **R-1** | **R-2** | **R-L** |
| *Fine-tuning* | 20/20 | 1.0 | 1.0 | 1.0 |
| RAG | 15/56 | >0.5 | >0.5 | >0.5 |

## ROUGE score categories

A good ROUGE score varies based on tasks on the summary and metrics. The ROUGE-1 score with the category is excellent, with a score of around 0.5. A score above 0.5 is considered good, and 0.4 to 0.5 is moderate. For ROUGE2, a score above 0.4 falls into the good category, and 0.2 to 0.4 falls into the moderate category.

The ROUGE-L score with a good category gets a score of around 0.4, and the low category ranges from 0.3 to 0.4. While the ROUGE score is helpful, it does not consider semantic or syntactic qualities and should be complemented by other metrics and human evaluation for a complete assessment [23].

Table ROUGE Metric Value Category Table [23]

|  |  |  |  |
| --- | --- | --- | --- |
| **ROUGE**  **Metric** | **Excellent** | **Good** | **Moderate** |
| ROUGE-1 | 0.5+ | >0.5 | 0.4-0.5 |
| ROUGE-2 | - | >0.4 | 0.2-0.4 |
| ROUGE-L | - | ~0.4 | 0.3-0.4 |

## Calculating Resource Evaluation

Calculating Resource Evaluation In this study, the researcher conducted a time test on the Mistral 7B model, with fine-tuning and inference testing. Researchers used a single NVIDIA Tesla T4 GPU graphics card in Google Colab for inference and fine-tuning. Based on the fine-tuning time test and using 20 datasets, as shown in Table 7, the researcher concluded that the fine-tuning approach with the unsloth library is effective and efficient and requires minimal time. The inference time test shows that the USK Mistral 7B does not require excessive computing power during the inference process, making it an energy-efficient and efficient system to respond to user inquiries only takes 5-6 minutes to respond to user inquiries. Hence, this model will be an efficient system for handling information related to USK's academic administration in the future.

Table Time Count on model while *fine-tuning* and running RAG

|  |  |  |
| --- | --- | --- |
| **Model** | **Fine-tuning *time* (hour)** | **RAG Time (minute)** |
| USK Mistral 7B | 2 | 5-6 |

## Result Analysis

## Training Data Problems

An important factor contributing to LLM hallucinations is the nature of the training data. LLMs, such as Mistral 7B, GPT, Falcon, and Llama, undergo extensive unattended training with large and diverse datasets from various origins. Verifying this data's fairness, impartiality, and factual correctness is a challenge. As the model learns to generate text, it can find and replicate factual inaccuracies in the training data. This learning leads to scenarios where the model cannot distinguish between truth and fiction and can produce outputs that deviate from facts or logical reasoning. LLMs trained on datasets sourced from the internet can contain biased or incorrect information. This misinformation can spread to the model's output, as the model cannot distinguish between accurate and inaccurate data.

## Reduces Hallucinations

Efforts to reduce hallucinations are essential to maintain the credibility and functionality of LLMs. The primary method for identifying and mitigating these errors involves a combination of advanced metrics and critical human evaluation. These include.

* Linguistic quality metrics such as ROUGE and BLEU
* Content validity metrics, which are IE-based, QA-based, and NLI-based
* FactScore to check the accuracy of individual facts

## Retrieval-Augmented Generation (RAG) Method

Innovative methods such as SelfCheckGPT detect hallucinations by assessing the consistency of multiple answers generated to the same question. In addition, techniques such as *chain-of-thought prompting* and *Retrieval-Augmented Generation* (RAG) are constantly being explored to strengthen the model's ability to provide precise and relevant information.

## The Influence of GPUs in LLM Implementation

GPUs play an essential role in running LLMs. Dedicated GPUs with high VRAM can significantly accelerate the computation required by the model. In this study, the GPU used is the "NVIDIA Tesla T4 GPU," available on Google Colab for free. The results of testing with this GPU using the RAG method take 4-5 minutes to generate responses to the questions.

# Conclusion

Fine-tuning the Mistral 7B model to USK Mistral 7B requires enormous data so that it can produce a better LLM in answering questions related to the lecture system and new student admissions at USK. It takes a long time, approximately 2 hours, to get a fine-tuning model with a dataset of 20 question-and-answer data.

The RAG method overcomes the limitations of generative AI when it requires information outside the LLM training corpus so this method will avoid LLMs that will generate inaccurate text, hallucinations, or distortions when providing answers to given questions. This RAG method can generate answers faster because it uses external data. The RAG method allows the model to avoid limitations on generative AI models. The response generated by the RAG method is able to produce a fairly good answer by looking at the ROUGE score that has been tested.

Based on open-source engineering metrics and assessment of computing resources, it can be concluded that USK Mistral 7B has the potential to be applied because with low energy consumption it can produce a response that has a ROUGE score of >5.

LLM, especially Mistral 7B has extraordinary potential in its application in various fields such as in the field of academic and administrative services. This study shows that with the training of a small amount of information data related to academics and administration at USK, USK Mistral 7B is able to respond well to various questions. The impressive performance of LLMs such as the Mistral 7B highlights its ability as a powerful tool to assist students in obtaining information at USK.

In the future, this model can also be applied to various fields, such as implementation in faculties with different information and also in government offices that provide services to the community, such as licensing services. Further research can also be done by testing on other newer models with more advanced features. By exploring this, the results of comparisons on other models are obtained so that the best model can be found for various tasks.

##### References

[1] S. Mohamadi, G. Mujtaba, N. Le, G. Doretto, and D. A. Adjeroh, “ChatGPT in the Age of Generative AI and Large Language Models: A Concise Survey,” pp. 1–60, 2023, [Online]. Available: http://arxiv.org/abs/2307.04251

[2] H. P. Baker, E. Dwyer, S. Kalidoss, K. Hynes, J. Wolf, and J. A. Strelzow, “ChatGPT’s Ability to Assist with Clinical Documentation: A Randomized Controlled Trial,” *J. Am. Acad. Orthop. Surg.*, vol. 32, no. 3, pp. 123–129, Feb. 2024, doi: 10.5435/JAAOS-D-23-00474.

[3] L. Loukas, I. Stogiannidis, P. Malakasiotis, and S. Vassos, “Breaking the Bank with ChatGPT: Few-Shot Text Classification for Finance,” *FinNLP-Muffin 2023 - Jt. Work. 5th Financ. Technol. Nat. Lang. Process. 2nd Multimodal AI Financ. Forecast. conjunction with IJCAI 2023 - Proc.*, pp. 74–80, 2023.

[4] A. Trozze, T. Davies, and B. Kleinberg, “Large Language Models in Cryptocurrency Securities Cases: Can ChatGPT Replace Lawyers?,” pp. 1–49, 2023, [Online]. Available: http://arxiv.org/abs/2308.06032

[5] Q. Huang, M. Tao, C. Zhang, and Z. An, “Lawyer LLaMA: Enhancing LLMs with Legal Knowledge,” 2023.

[6] A. Bhatti, S. Parmar, and S. Lee, “SM70: A Large Language Model for Medical Devices,” no. 1, pp. 1–5, 2023.

[7] H. Zhao *et al.*, “Ophtha-LLaMA2: A Large Language Model for Ophthalmology,” *ArXiv [Preprint]*, pp. 1–19, 2023, [Online]. Available: http://arxiv.org/abs/2312.04906

[8] S. Barandoni, F. Chiarello, L. Cascone, and S. Puccio, “Automating Customer Needs Analysis : A Comparative Study of Large Language Models in the Travel Industry,” 2024.

[9] V. Jonatan and and A.-A. Igor, “Creation of a Chatbot Based on Natural Language Processing for Whatsapp,” *J. Database Manag.*, vol. 3, no. 4, pp. 39–53, 2023, doi: 10.14810/elelij.2023.12402.

[10] data.usk.ac.id, “Data Mahasiswa Daftar Ulang,” data.usk.ac.id. Accessed: Apr. 13, 2024. [Online]. Available: https://data.usk.ac.id/mahasiswa-daftar

[11] T. Fatyanosa, “Fine-Tuning Pre-Trained Transformer-based Language Model,” medium.com. Accessed: Jan. 25, 2024. [Online]. Available: https://fatyanosa.medium.com/fine-tuning-pre-trained-transformer-based-language-model-c542af0e7fc1

[12] K. T. Wicaksono, “Memahami Retrieval Augmented Generation (RAG) Melalui Rencana Pembangunan LaporMacet.com,” medium.com. Accessed: Jan. 25, 2024. [Online]. Available: https://kukuhtw.medium.com/memahami-retrieval-augmented-generation-rag-melalui-rencana-pembangunan-lapormacet-com-5990a76edda7

[13] L. Xu, H. Xie, S.-Z. J. Qin, X. Tao, and F. L. Wang, “Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment,” pp. 1–20, 2023, [Online]. Available: http://arxiv.org/abs/2312.12148

[14] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “QLORA: Efficient Finetuning of Quantized LLMs,” *Adv. Neural Inf. Process. Syst.*, vol. 36, 2023.

[15] E. Hu *et al.*, “Lora: Low-Rank Adaptation of Large Language Models,” *ICLR 2022 - 10th Int. Conf. Learn. Represent.*, pp. 1–26, 2022.

[16] D. Han-Chen, “Make LLM Fine-tuning 2x faster with Unsloth and ? TRL,” https://huggingface.co/blog/unsloth-trl. Accessed: Jun. 30, 2024. [Online]. Available: https://huggingface.co/blog/unsloth-trl

[17] K. V., “Fine-Tuning Large Language Models with Unsloth,” medium.com. Accessed: Jul. 30, 2024. [Online]. Available: https://medium.com/@kushalvala/fine-tuning-large-language-models-with-unsloth-380216a76108

[18] P. Lewis *et al.*, “Retrieval-augmented generation for knowledge-intensive NLP tasks,” *Adv. Neural Inf. Process. Syst.*, vol. 2020-Decem, 2020.

[19] M. Grootendorst, “Which Quantization Method is Right for You? (GPTQ vs. GGUF vs. AWQ),” maartengrootendorst.com. Accessed: Feb. 29, 2024. [Online]. Available: https://www.maartengrootendorst.com/blog/quantization/

[20] docs.llamaindex.ai, “High-Level Concepts,” docs.llamaindex.ai. Accessed: Jul. 28, 2024. [Online]. Available: https://docs.llamaindex.ai/en/v0.10.17/getting\_started/concepts.html

[21] C.-Y. Lin, “ROUGE: A Package for Automatic Evaluation of Summaries Chin-Yew,” pp. 74–81, 2004, doi: 10.1253/jcj.34.1213.

[22] M. U. Amanat, “LLM evaluation with Rouge,” medium.com. Accessed: Apr. 24, 2024. [Online]. Available: https://medium.com/@MUmarAmanat/llm-evaluation-with-rouge-0ebf6cf2aed4

[23] S. M. Walker II, “What is the ROUGE Score (Recall-Oriented Understudy for Gisting Evaluation)?,” klu.ai. Accessed: Jul. 30, 2024. [Online]. Available: https://klu.ai/glossary/rouge-score