AI-Assisted Grading System for General Paper Essay Exams Using Large Language Models

Abstract

Grading General Paper (GP) essays is a time-intensive and subjective process that poses significant challenges for educators, including inconsistent evaluations and delayed feedback. To address this, our project proposes an AI-assisted grading system leveraging Large Language Models (LLMs) to automate the assessment of GP essays. By evaluating grammar, factual accuracy, argumentation strength, and relevance to the topic, the system aims to provide accurate and unbiased scoring aligned with the GCE A-Level GP syllabus. This solution offers substantial time savings, ensures grading consistency, and delivers instant, actionable feedback to students. Our prototype system, built using fine-tuned LLMs and validated against human-graded essays, demonstrates strong potential to enhance productivity in education while maintaining the integrity and fairness of academic evaluation.

1. INTRODUCTION

1.1 Problem Statement

In today's education landscape, teachers face increasing pressure to deliver timely, high-quality feedback—especially for subjects like General Paper (GP), where open-ended essay responses demand careful and nuanced evaluation. The manual grading process is not only time-consuming but also cognitively taxing, requiring teachers to assess grammar, argument strength, factual accuracy, and alignment with curriculum rubrics. These challenges are magnified by large class sizes and varied student perspectives, often resulting in grading inconsistencies and delayed feedback.

1.2 Background and Motivation

General Paper plays a pivotal role in nurturing critical thinking and written expression in Junior College students. Yet, the very nature of GP essays—broad in scope and open to interpretation—makes grading both laborious and subjective. Teachers must assess the clarity and structure of arguments, evaluate the relevance and accuracy of examples, and ensure alignment with syllabus expectations. This burden limits their ability to provide individualized feedback and support student improvement.

Recent developments in natural language processing, particularly LLMs like GPT, offer promising tools to enhance this process. These models are capable of analyzing text at scale and generating human-like responses, making them well-suited for preliminary grading and feedback generation. By leveraging model essays and their corresponding scores from GPEssays.sg, our project aims to train and validate an AI-assisted grading system that ensures consistency, reduces grading time, and upholds academic standards.

1.3 Project Objective and Scope

The primary goal of this project is to build an AI-powered grading assistant for GP essays, using fine-tuned LLMs that align with established grading rubrics. The system will assess key aspects such as argument structure, grammar, factual accuracy, and topical relevance. Beyond assigning scores, it will generate actionable, transparent feedback to help students reflect and improve.

Ground truth data will be sourced from GPEssays.sg, where essays have been graded according to the GCE A-Level GP standards. Although the initial focus is on GP, the methodology is designed to be adaptable to other content-heavy, essay-based subjects such as history, literature, or geography—opening up opportunities for broader application within education.

2. DATA COLLECTION AND EXPLORATION

2.1 Data Source

To train and evaluate the AI grading model, we collected high-quality GP essays and their associated metadata from GPEssays.sg, a publicly available repository of model essays graded according to Singapore's A-Level standards. A custom Python web scraper was developed using the `requests` and `BeautifulSoup` libraries to systematically extract essay content, titles, grades, and additional meta-information across all paginated pages.

The scraper iterates through each article on the site, parses the relevant metadata, and compiles the data into a structured format. The result is a DataFrame containing essays labeled with ground-truth grades, which is then saved into a CSV file for easy access and further analysis. This dataset, enriched with metadata, forms the foundation for fine-tuning and evaluating the LLM-based grading system.

2.2 Data Augmentation

To expand the diversity and depth of our dataset, we implemented a data augmentation process aimed at simulating a variety of student writing s tyles and common essay pitfalls. This approach allowed us to create a richer set of examples to develop a model to score the essays accurately.

Starting with a set of original student essays, including the essay title and grade, we generated three additional versions for each essay using large language models accessed via the Groq and Ollama APIs.

The first type of augmentation focused on producing low-quality rewrites. Using both the LLaMA-3 (llama3-70b-8192) and Gemma-2 models, we prompted the models to simulate essays written by students who struggled with grammar, structure, and clarity. The prompts were carefully phrased to encourage the introduction of awkward phrasing, disorganized reasoning, and incorrect or poorly explained examples, while still following the original topic. To avoid duplication, the prompts emphasized drawing inspiration from the original essay without copying specific content or structure.

The second augmentation generated off-topic essays. In this case, the model was instructed to start on the assigned topic but gradually veers off-course, ultimately failing to address the core question. The prompts were adjusted to preserve grammatical correctness and a formal tone, creating essays that might superficially appear well-written but miss the point entirely.

This prompt-based augmentation strategy allowed us to simulate realistic student responses across a range of quality and relevance, without requiring manual annotation or rewriting. It also provided a scalable way to create nuanced variations for testing the robustness of language models in educational settings.

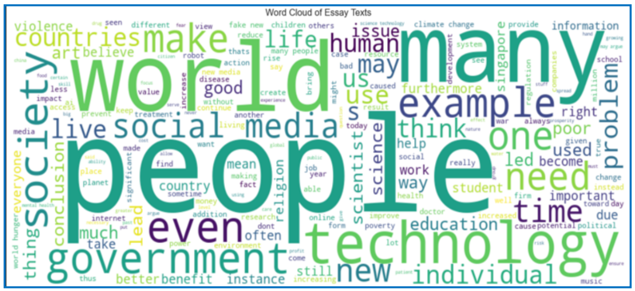
These augmented essays provide multiple perspectives or styles on the same subject, allowing us to later assess how consistently and accurately an LLM can grade different versions of content related to the same topic.

2.3 Exploratory Data Analysis

2.3.1 Thematic Coverage via Word Cloud

To better understand the thematic distribution of our training dataset, we generated a word cloud visualizing the most frequently occurring terms across the essays used for fine-tuning the language model. This visualization serves two primary purposes: (1) it provides a high-level overview of the dominant topics' students write about, and (2) it helps assess whether our dataset reflects the content diversity typical of General Paper essays. From the word cloud, we observe that terms like “people,” “government,” “society,” “technology,” “media,” and “world” appear prominently, indicating a strong focus on socio-political, ethical, and scientific themes. This supports our objective of training a grading model that is not only linguistically competent, but also contextually aware of the kinds of arguments and examples students typically present. The word cloud also serves as a quality control mechanism—if the vocabulary were too narrow or repetitive, it would signal a need to augment or diversify the training data. Overall, the word cloud offers a qualitative lens through which we verify the alignment between our dataset and the real-world grading scenarios the model is expected to handle.

2.3.2 Essay Length and Tokenization Constraints



An important factor we explored during EDA was the length of the essays, which directly influences tokenization and model input limits during both fine-tuning and inference. The essay length distribution shows that the majority of essays fall between 250 and 350 words, with the highest frequency observed around the 300–350 word range, where over 40 essays are concentrated. At a typical tokenization rate of 1.3–1.5 tokens per word, these essays translate to approximately 400–525 tokens each—comfortably within the context limits of most LLMs (e.g., 4096 or 8192 tokens). Nonetheless, a subset of essays exceeds 1500 words, which may require truncation, summarization, or chunked processing to fit within API constraints. Recognizing these limits early on is essential for designing effective prompts and ensuring smooth integration with LLM architectures.

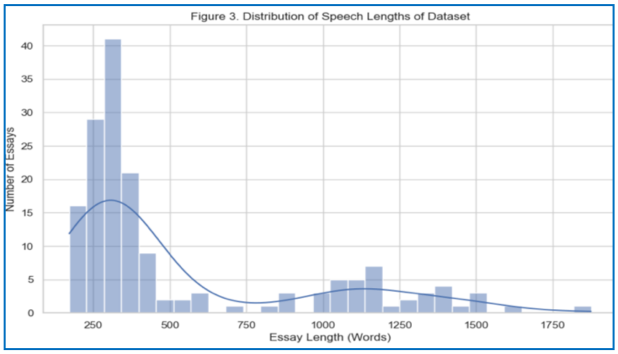
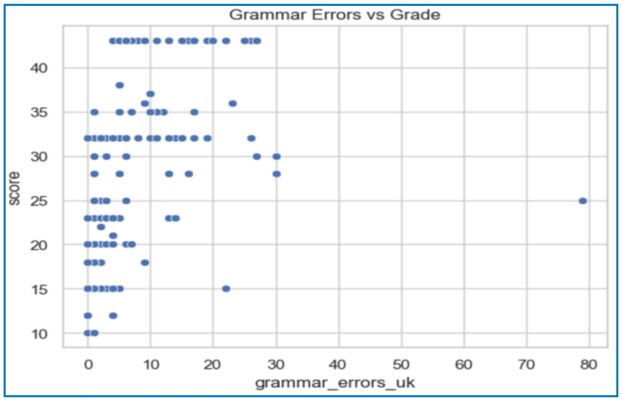


Figure 3. Distribution of Speech Lengths of Dataset

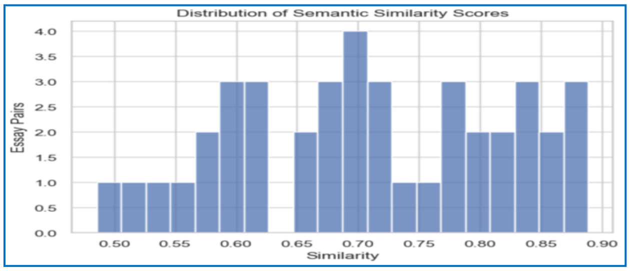
2.3.3 Grammar Errors and Scoring Trends



To understand the impact of writing fluency on scores, we plotted grammar error counts against essay grades. The trend shows that high-scoring essays (typically above 35) tend to have fewer than 10 grammar issues.

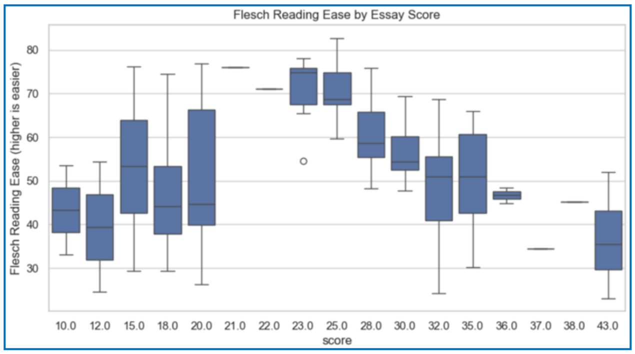
In contrast, lower scores are often associated with a higher number of errors—some exceeding 70. While exceptions exist (e.g., essays with high scores despite moderate grammar issues), the pattern suggests grammar contributes to perceived quality. This makes it a valuable auxiliary feature when training an LLM to grade essays. It helps the model learn that clarity and correctness matter—though they must be considered alongside content and structure.

2.3.4 Semantic Diversity in Augmented Essays



To assess the diversity of content generated for the same essay topics, we computed semantic similarity scores between original essays and their augmented counterparts. The goal was to ensure that while the core topic remains the same, the generated content introduces enough lexical and structural variation to provide different terms of reference. As shown in the distribution, most similarity scores range from 0.60 to 0.85, indicating a balanced level of semantic overlap—high enough to retain topic alignment, yet low enough to avoid redundancy. This suggests that the augmentations are not simple paraphrases but contribute meaningfully different perspectives on the same subject. Such diversity is essential for training robust grading models that can fairly evaluate a range of expression and argument styles.

2.3.5 Readability vs. Score: Flesch Reading Ease



This analysis reveals an inverse relationship between Flesch Reading Ease and essay scores: higher-scoring essays tend to have lower readability scores, indicating greater linguistic complexity. This aligns with expectations for General Paper-style writing, where markers value essays that demonstrate a strong command of language, critical reasoning, and sophisticated vocabulary — even at the cost of simplicity. Therefore, unlike in general writing tasks where clarity might be paramount, in this context, linguistic richness and complexity appear to be rewarded. This is a critical insight for LLM alignment: the model should not favor simpler essays but learn to recognize and score well-structured, complex academic writing appropriately.

3. METHODOLOGY

3.1 Data Pre-processing and Embedding

To enable semantic-level understanding and retrieval, we begin by embedding all essay titles—including those associated with augmented content—into dense numerical vectors. These embeddings are generated using the sentence-transformers/all-mpnet-base-v2 model from Hugging Face, which effectively captures the contextual and semantic meaning behind each title. Once generated, the vectors are indexed and stored using FAISS (Facebook AI Similarity Search), a library optimized for fast similarity search across large-scale datasets. This setup allows the system to retrieve essays that are topically like a given query by calculating cosine similarity between their embeddings—laying the groundwork for meaningful comparison and evaluation.

3.2 Rubric Extraction

To establish a fair and structured grading standard, our pipeline extracts the official marking rubric from the Cambridge International 2019 GP guide. Instead of manual distillation, we leverage Gemini 2 Flash, an LLM capable of abstracting core grading principles from extensive training on educational content. The result is a concise, general-purpose rubric that captures key dimensions of effective essay writing—readily applicable across diverse topics.

3.3 Leveraging Similarity Search for Essay Evaluation

We enhance the evaluation process by implementing a similarity search mechanism. FAISS is used again here to index a database of previously graded essays, which serve as reference points.

When a new essay is submitted, it is compared against this indexed database to retrieve the most topically similar essays—two for Gemini and one for Groq. These retrieved exemplars provide concrete benchmarks that guide the grading process, allowing both human evaluators and automated systems to assess new submissions more consistently and effectively.

Embedding and indexing essays in this way not only ensures fast retrieval but also supports fair and contextually aware grading.

3.4 LLM Grading System

With both original and augmented essays ready, we pair each submission with its corresponding topic and feed it into a Large Language Model (LLM) for grading. This setup allows us to systematically evaluate how the model handles different stylistic expressions of the same idea.

By comparing grading outcomes across original and augmented submissions, we assess:

- **Consistency** of the LLM's scoring across different versions of the same topic
- **Sensitivity** to variations in tone, structure, and vocabulary
- **Semantic robustness** in recognizing topic relevance and overall content quality

This end-to-end pipeline—spanning data augmentation, embedding, similarity search, and LLM grading— forms scalable and reliable framework for evaluating essay performance in an automated yet nuanced manner.

4. MODELS AND MODEL COMPARISON

4.1 Models Picked and their differences

We picked two models to be used as evaluators. Namely Gemini Flash and Llama3-70b. These models are picked under several consideration.

- Given the budget constraint, the project is designed to leverage on freely accessible models and tools to ensure cost-effectiveness.
- It is preferred to use an API over locally hosted LLM, as more advanced models need powerful hardware to be run locally, which is not always available.
- The models need to be able to reason, grade, and provide feedback on the papers reasonably well at a level that is at least close to that of human graders.

These constraints are what makes us choose two large language models, Gemini 2 Flash and Llama 3 (70B), because each of them has different strengths that, when combined, allow us to build a grading pipeline that is effective and robust. Gemini 2 Flash is used as the first-pass grader due to its speed, cost, and prompt instruction sensitivity. It typically answers with a score and rationale in 1.5 to 2 seconds, which can be deployed at scale. Its ability to read structured rubrics and few-shot examples is closely in the spirit of how instructors grade essays, and its answers are typically pedagogically nice, with explanations that are close to human feedback. Gemini sometimes is too lenient, especially on essays with barely logical mistakes or weak arguments.

To assist in fixing this, we introduced a second-pass grader. We introduced a strict grader, still based on the Gemini Model, and a lenient grader based on Llama 3 (70B) on Groq. Then, the optimal weightage between the strict and the lenient model is derived through the training data. The final model will be the scores of each model, multiplied by the derived weights. The specifics of this will be discussed in the prompt engineering section and the evaluation section.

This model, although slower (at around 10 seconds per essay), was stricter and analytically deep in our experiments. It was particularly adept at penalizing imprecise assertions, structural errors, and off-topic content. Even though the two models receive the same prompt, the same rubric, and the same exemplar essays, Llama will render different judgments due to its own architecture and training. Having Gemini and Llama together allows us to take the average of two independent grades and generate double rationales, which enhances reliability and transparency. When the two models agree, we can trust the grade more. When their grades differ significantly (by over 10 points), we flag the essay for human grading, simulating how instructors would consult colleagues in borderline situations. This hybrid approach achieves a balance between speed, cost, and grading accuracy while keeping up with real-world classroom traditions. It also avoids having to train an in-house model from scratch, one that would require additional data and still not come close to the depth and richness that these foundation models attain.

Our models also provide feedback on each essay in the form of an explanation for the assigned grade, which doubles as constructive input for the student. This feedback helps students understand the reasoning behind their scores and identify areas for improvement. Both Gemini and Llama generate their own explanations, and students can receive feedback from both models to better rationalize their results.

4.2 Model Parameter

In establishing our grading framework, two global-level considerations influence how we interact with large language models: temperature and context window. Both have a direct influence on the consistency, reliability, and feasibility of applying LLMs in an assessment setting.

The temperature is set to 0.0 for both models. That is because this grading assignment has to be a deterministic model, where consistency is more important than creativity. What is desired is that an essay grading should be consistent and produce the same scores for the same essay. A model that scores the same essay differently on two consecutive runs would kill confidence in the system. By keeping the temperature constant at zero, we can be sure that the same essay always creates the same output, which is something we need to make honest and reproducible grading possible.

The other significant concern is the context window, the size of tokens a model can handle from one prompt that has instructions, exemplars, rubrics, and the essay. Gemini 2 Flash has an extremely large context window (up to 1 million tokens in some architectures, according to reports) that makes it extremely flexible for complex input. On the other hand, Llama 3 (70B) on Groq has a context window of 8192 tokens. While this is fine for most essays and grading prompts, we need to take care when designing prompts especially those that have multiple exemplars or are very long to stay under this limit. When the prompt or essay is too long, we employ token truncation or reduce the number of examples retrieved from the vector store. This limitation is a big factor in why only 1 essay is loaded as the standard to Llama instead the 2 for Gemini Flash. By being intentional regarding such parameters, we achieve balance between model performance and system robustness. Decisions are made so that our graders become predictable, compliant with API needs, and resilient enough to accommodate real-world data without compromising grading integrity.

4.3 Prompt Engineering

The ability of large language models to complete the grading task efficiently depends on prompt structure. As previously stated, each prompt contains an example essay and the rubric to help the model understand it. In order to help the model adopt the proper evaluative tone, we start the prompt with a clear role description, such as "You are an experienced educator tasked with grading essays." Clear formatting and instructions are then provided to reduce ambiguity and match the model's output with the expected grading behavior. The prompt specifically asks for a response in the format of "Score: <0–100>" and "Explanation: " to make sure the model offers both a score and a justification. Both quantitative findings and qualitative comments are guaranteed to be returned in a consistent manner thanks to this formatting.

To further refine the prompt, we designed three distinct prompt engineering strategies tailored to different configurations: a basic Gemini model, an enhanced Gemini model with references, and a lenient Llama model with references. These strategies were developed with as part of the model architecture stated in the model description.

The baseline model used a straightforward prompt structure. It provided only the essay title, the essay text, and a marking rubric to the Gemini model. The instructions positioned the model as an experienced educator, tasked with assigning a score out of 100 and briefly justifying the score. This version served as a control without exposure to reference essays.

To improve grading quality, we introduced a more refined prompt for the second Gemini-based configuration. In addition to the essay and rubric, this version incorporated exemplar essays retrieved from a vector database of high-quality model answers. These examples were provided in JSON format, allowing the model to compare the student's work against expected structures and arguments. The prompt emphasized strict adherence to the rubric and encouraged more critical grading where submissions were poorly structured or lacked depth.

The third model, based on Llama-3 70B, used a similar setup to the enhanced Gemini prompt but adopted a different instructional tone. The system prompt emphasized leniency and pedagogical empathy—encouraging the model to reward effort even when the essay was not fully on topic or logically sound. While the model still penalized off-topic responses, the grading approach was intentionally more forgiving than its Gemini counterpart.

Each model returned output in a consistent format: a numeric score followed by a textual explanation. The effectiveness of the prompt engineering strategies are discussed in the evaluation.

4.4 Model Design

To conclude this model uses a vector similarity search, the system first embeds essay titles and finds pertinent exemplars. Two language models, Gemini 2 Flash and Llama 3 (70B), are given structured prompts that contain these exemplars and a rubric taken from an official marking guide. Every model produces its own grade and justification. Following parsing and averaging, these outputs are optionally marked for manual review in the event that there is a substantial score difference. Responses are kept deterministic, rubric-aligned, and pedagogically meaningful by carefully adjusting prompt formats, temperature settings, and context constraints. Now that this pipeline is operational, we can evaluate the models' performance in practice

5. EVALUATION AND FINAL MODEL

To evaluate our base model and our refined model, we split the augmented data into a train/test split. We decided there was no value in using a validation set, we do not have the resources to do meaningful hyperparameter tuning. In the ideal scenario, prompts should iteratively get better, such that the model's essay score is like the ground truth score.

Each model returned output in a consistent and structured format, comprising a numeric score and a brief textual explanation. This uniform response format was essential for downstream processing—it enabled automated parsing, alignment with human-assigned grades, and consistent comparison across different model outputs.

To evaluate the models' grading accuracy, I used absolute loss as the primary performance metric. For each essay, I calculated the absolute difference between the model's predicted score and the human-assigned score (converted to a 100-point scale), and then averaged these differences across all samples. This gave the **L1** mean absolute error (MAE) for each model. MAE was chosen for its interpretability and robustness—it treats all errors equally without over-penalizing outliers, making it especially suitable in educational contexts where both accuracy and fairness are valued.

The table below summarizes the average loss (MAE) for each model on both the training and test sets:

Model	Training Loss	Test Loss
Basic Gemini	11.15	12.08
Enhanced Gemini	7.49	6.56
Llama	16.64	17.48
Weighted Ensemble	—	7.54

Note: The weighted ensemble was evaluated only on the test set using the optimal weights derived from training data.

Recognizing that each model captured different aspects of grading—Gemini emphasizing rigor and structure, Llama offering flexibility and empathy—I implemented a weighted ensemble strategy to combine their strengths. Using *scipy.optimize.minimize*, I found the optimal linear weights that minimized total absolute loss on the training set. The final formula was:

$$\text{Weighted Score} = 0.7059 \cdot \text{Gemini}_{\text{score}} + 0.2941 \cdot \text{Llama3}_{\text{score}}$$

While the enhanced Gemini model achieved the lowest average loss on the test set (6.56), it was designed to be deliberately strict, emphasizing alignment with structure, argument quality, and rubric expectations. This strictness, while effective for benchmarking, may not fully account for the nuances of student effort or less conventional writing styles.

In contrast, the weighted ensemble offers a more balanced and fairer evaluation by integrating Llama's more empathetic grading approach. By assigning approximately 70% weight to Gemini and 30% to Llama, the ensemble retains most of Gemini's precision while softening overly harsh penalties. This makes the model more aligned with how a human educator might assess borderline cases. It rewards effort, partial understanding, and creativity, even when the essay deviates slightly from ideal academic form.

Although the ensemble's test loss (7.54) is marginally higher, it reflects a trade-off between strict accuracy and fairness to diverse student responses. As such, the ensemble may be better suited for deployment in educational settings where scoring equity and encouragement are just as important as rubric compliance.

6. KEY TAKEAWAYS

6.1 Key Insights

6.1.1 Survivorship Bias in Training Data

The model was trained on published model essays (high-scoring examples), creating a skewed understanding of "good" essays.

Why It Matters: Real student essays often include errors (e.g., grammar mistakes, weak arguments) that the system isn't exposed to, leading to unreliable feedback for average/poor essays.

6.1.2 Title ≠ Content Alignment

The model uses title embeddings (via FAISS) to find similar essays but doesn't check if the essay's content aligns with its title.

Why It Matters: A student could write an essay titled "Do Schools Kill Creativity?" but focus on unrelated topics like climate change. The system might still retrieve similar titles, missing the off-topic content.

6.1.3 Token Limitations Restrict Grading

Free-tier APIs (Gemini, Groq) have token limits (~6,000 tokens for Groq), forcing truncation of long essays.

Why It Matters: Critical sections of essays might be cut off, leading to incomplete feedback. For example, a 1,000-word essay might lose its conclusion, skewing the grade.

6.1.4 Deterministic Grading ≠ Human Nuance

Models are set to temperature=0.0 for consistency, but this removes flexibility in judging subjective elements (e.g., creativity).

Why It Matters: Human graders might reward a creative but imperfect argument, while the model penalizes it strictly.

6.2 Limitations

6.2.1 Rubric Extraction

Gemini's rubric extraction works for explicit essay requirements, but it may miss implicit requirements, for example, the essays should be written in a particular context.

6.2.2 Topic Drift Undetected

The use of title-based vector similarity fails to detect whether the essay written by the student has deviated from the topic.

The current model architecture does not analyze the relevancy of the arguments presented by the student and there is a potential that even if the student has ventured off topic, he can still receive a good grade for it.

This is erroneous so after retrieving the terms of reference for essays written for a similar topic, the LLM should be configured to analyze the content – to see if the essay content has deviated from the essay topic.

6.2.3 No Fact-Checking Mechanism

We have prompted the model to evaluate the structure and grammar of the essays. However, it is unable to verify factual claims.

6.3 Challenges

6.3.1 Data Scarcity and Diversity

The team does not have access to real student essays and only has access to model essays published on the website.

Furthermore, the model essays for a particular topic lacks diversity (e.g. no cases with factual inaccuracies/poor grammar) to test for the model's content detection capabilities.

As such, the mode is unable to validate the performance of essays that are written poorly.

6.3.2 API dependency and cost

Free-tier token limits from Grok/Gemini's API restricts the grading of the number of essays.

Each essay can easily exceed the token limits, especially when candidates must write between 500 to 800 words on one question of their choice. In our EDA, even our samples are up to 400 words long.

In the latest exam conducted in 2024, there were a total of 10,889 students who took the exams. Suppose the API were to be called to grade the essays, and we want to maintain using the free tier – then it will slow down feedback given. Otherwise, costs will be incurred if we want to analyze the number of essays submitted.

6.3.3 Validation gaps

Even as we augmented the data to create a variety of data for the same topic, we lack the expertise to grade the generated essays.

As a result, we are unable to evaluate how far apart the scores given by the system differ from the scores graded by teachers.

This will give rise to confidence issues as we are unable to ascertain if the model aligns with the grading standards.

6.4 Areas of improvement

6.4.1 Data Augmentation

Given the limited dataset on poorly written essays, the team leveraged GenAI to produce essays with intentional errors (factual inaccuracies, grammatical errors, topic drift etc) to test on the robustness of the model.

It would be good if we were able to partner with schools to collect real student essays for balanced data training.

6.4.2 Enhancing Validation of Model

Have essays graded by teachers and compare the scores graded by the AI model.

Implement a confidence score system by the model – let it suggest a proposed score with % confidence and give reasons for the proposed score.

6.4.3 Improvements to Topic Adherence

We can add contact aware embeddings – such as embedding the entire essay instead of just the title and compare against the title's embedding.

We have also experienced improvements to the prompts to indicate whether the essay content is in line with the topic of the essay.

6.4.4 Overcoming token/api limitations

Given that Free-tier APIs restrict how much text can be processed, resulting in truncated essays or reduced examples.

We can refine our inputs by

a. Local Preprocessing:

Use smaller, free models (like Mistral-7B) to summarize essays or check grammar before sending them to paid APIs.

Example: Mistral can shorten a 1,000-word essay to 300 words, saving tokens for critical analysis.

b. Chunk Long Essays:

Split essays into sections (e.g., introduction, arguments, conclusion) and grade each part separately.

7. CONCLUSION

The integration of Large Language Models (LLMs) into the grading of General Paper essays demonstrates significant potential to transform how educators assess student work.

By automating preliminary feedback on grammar, argument structure, and topic relevance, our AI-assisted system reduces grading time while maintaining alignment with Singapore's GCE A-Level standards.

However, this project underscores the importance of balancing automation with human oversight.

While LLMs like Gemini and Llama-3 excel at pattern recognition and rubric-based scoring, they lack the contextual nuance and ethical judgment that teachers bring to evaluations.

Challenges such as detecting subtle topic drift, verifying factual accuracy, and overcoming training data biases (e.g., reliance on high-scoring model essays) highlight the need for hybrid workflows. For instance, flagging borderline essays for human review or augmenting LLM feedback with domain-specific fact-checking tools can bridge these gaps.

Going forward, to improve on the model, 3 points stand out:

- **Increase the diversity of data** – we can partner with schools to collect anonymized essays which have been graded. This will improve the model's ability to recognize the common errors and reward originality.
- **Enhancing the interpretability of our model** – we can strive to provide teachers with explainable AI insights (such as highlighting off topic paragraphs, or what were the issues that resulted in such a score being generated). This would build trust in the model's recommendations.
- **Optimizing cost** – Exploring lightweight or locally hosted models for preprocessing tasks such as grammar checks. This will potentially reduce token limit APIs. This will allow the model to be used for a large number of essays.

Ultimately, the goal is to develop a model that has the trust of teachers – that the model can generate the strengths and weaknesses of what was written by the student before giving a grade to the essay.

It is not intended to replace teachers but to assist and empower them in their work. By cutting down on time spent grading essays, they will have the ability to focus on teaching their students what matters, such as critical thinking, communication skills and engage them in complex issues plaguing our world today.

REFERENCES

GPEssays.sg. GP model essays. GPEssays.sg. URL: https://gpessays.sg/gp-model-essays/.

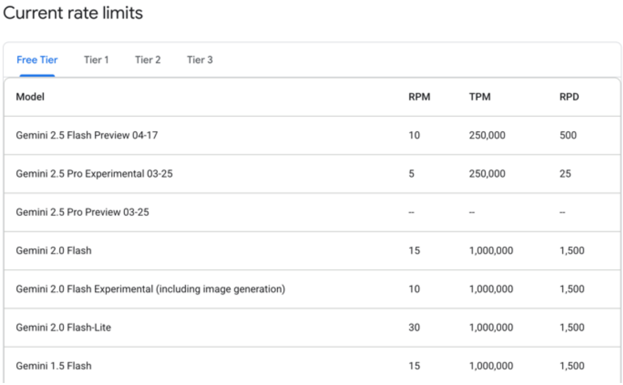
Singapore GCE A Level General Paper requirements and scheme of assessment 8881_y25_sy_pdf__updated_.pdf

Number of students taking the general paper exam - Release of 2024 Singapore Cambridge GCE Advanced Level Examination Results

Group12 BT5153-Essay Code. 2025. *BT5153-Essay*. Available at: https://github.com/graysonjova/BT5153-Essay

APPENDIX

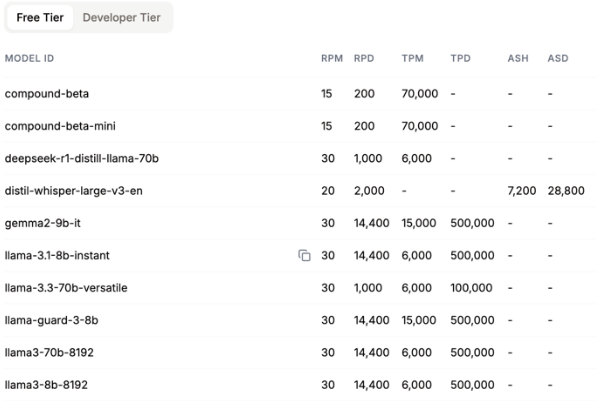
Rate limitations from Gemini



Current rate limits

Model	RPM	TPM	RPD
Gemini 2.5 Flash Preview 04-17	10	250,000	500
Gemini 2.5 Pro Experimental 03-25	5	250,000	25
Gemini 2.5 Pro Preview 03-25	--	--	--
Gemini 2.0 Flash	15	1,000,000	1,500
Gemini 2.0 Flash Experimental (including image generation)	10	1,000,000	1,500
Gemini 2.0 Flash-Lite	30	1,000,000	1,500
Gemini 1.5 Flash	15	1,000,000	1,500

Token limitations from GroqCloud:



MODEL ID	RPM	RPD	TPM	TPD	ASH	ASD
compound-beta	15	200	70,000	-	-	-
compound-beta-mini	15	200	70,000	-	-	-
deepseek-r1-distill-llama-70b	30	1,000	6,000	-	-	-
distil-whisper-large-v3-en	20	2,000	-	-	7,200	28,800
gemma2-9b-it	30	14,400	15,000	500,000	-	-
llama-3.1-8b-instant	30	14,400	6,000	500,000	-	-
llama-3.3-70b-versatile	30	1,000	6,000	100,000	-	-
llama-guard-3-8b	30	14,400	15,000	500,000	-	-
llama3-70b-8192	30	14,400	6,000	500,000	-	-
llama3-8b-8192	30	14,400	6,000	500,000	-	-

Credits/Disclaimer: Deepseek and ChatGPT was used to develop the code and develop part of the essay writeup.