Automated Grading of Free-Text Student Submissions Using Large Language Models.

Subtitle

School of Computer Science & Applied Mathematics
University of the Witwatersrand

Sphamandla Mbuyazi
2618115

Supervised by Prof. Richard Klein

April 13, 2025

Ethics Clearance Number: XX/XX/XX

A proposal submitted to the Faculty of Science, University of the Witwatersrand, Johannesburg, in partial fulfilment of the requirements for the degree of Bachelor of Science with Honours

Abstract

Abstract things....

Declaration

I, Sphamandla Mbuyazi, hereby declare the contents of this research proposal to be my own work. This proposal is submitted for the degree of Bachelor of Science with Honours in Computer Science at the University of the Witwatersrand. This work has not been submitted to any other university, or for any other degree.

Acknowledgements

Thanks World.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

In education technology, auto grading of free text has long been a sought-for solution, particulary in large-scale assessments and technical subjects like Computer Science , where assessments cannot be easily reduced to multiple choice questions. As the number of students enrolled in a course increases per instructor and the curriculum becomes more demanding, educators face a mounting pressure to provide fair, consistant and fast feedback.

This posses a need to try to automate the process of marking and giving feedback. In the state of art of education right now, your instructor grades your assessments and 4 years later provides you with a feedback, as we can argue without proof that early feedback helps students perform better in their studies.

At the same time, Large Language Models(LLMs) like GPT-4, Claude, and open-source alternative such as LLaMA and DeepSeek are proving to be not just the conversational tools but they are capable evaluators, summarizers, and even tutors. Their impressive capacity for natural Language understanding and generation raises an exciting possibility:

Can these models be harnessed to grade student responses automatically and meaningfully? –and perhaps even provide personalized feedback that guides learning?

1.1.1 Purpose of this review

The aim of this review is to syntesize and critically examine recent research on the use of LLMs for autograding and feedback generation in educational contexts. We shall look at how these models have been applied to assess student submissions, what techniques have been used to improve reliability and fairness, and how feedback mechanisms are integrated to support and elevate student learning.
By comparing methodologies, models and findings we aim to infer what is working,

what remains challenging, and where there's space for improvement. This should ultimately inform how we design our own LLM-based grading system for technical subjects, one that minimizes educators workload while enhancing the feedback experienca

1.1.2 Brief overview of the key themes

Across the literature I have examimed, several key themes and technical strategies emerge.

Theme	**Approaches / Observations**
Grading Accuracy	LLMs like GPT-4 and fine-tuned BERT models show strong agreement with human graders.
Feedback Generation	Models like BeGrading and GPT-4 in science writing generate formative feedback students find useful.
Explainability	SHAP explanations and linguistic feature analysis offer insight into deep model decisions.
Data Efficiency	Active learning (uncertainty, topology, hybrid methods) helps reduce labeling needs.
Prompt Engineering	Careful prompt design significantly improves LLM grading consistency and relevance.
Peer/Human-in-the-loop	Hybrid systems using peer grading or instructor scaffolding improve fairness and training.
Open-Source Model Potential	LLaMA 2 and Falcon perform comparably to commercial models in bioinformatics grading tasks.

Table 1.1: Themes and Observations in AI-based Educational Assessment

We shall explore the above themes in deapth next section, critically comparing the approaches, results and proposed direction of each study.

Chapter 2

Body

2.1 Body.

2.1.1 From Black-Box to transparent graders

We trust teachers because when we ask them why, they give us a sensible responses; can we do the same for machines?

In the early days of Automated Essay Scoring(AES), the focus was primaraly on **Accuracy**: this means really we were answering the question could a machine assign the same score a human would? Systems like e-raters while wildy adopted, remained blacked boxes. What this essentially means is it can produce grades without justification. Without clear reasoning for scores, students were left with limited guidance for improvement, and the instructors struggled to verify the fairness and reliability of these systems.

Mok (2023) addressed this limitation by developing a deep learning AES model based on a Multi-Layer Perceptron (MLP) architecture, trained on the ASAP Grade 7 narrative writing dataset. The model predicted rubric-level scores across four dimensions — ideas, organization, style, and conventions. Importantly, this work introduced SHAP (SHapley Additive exPlanations) to improve interpretability. SHAP provided both global (dataset-wide) and local (individual essay) feature attributions, making it possible to understand how linguistic features (such as lexical diversity, grammatical structure, or cohesion) contributed to predicted scores.

The model extracted over 1,500 linguistic indices using the SALAT toolkit, representing textual characteristics across five linguistic domains. Through regularization (Lasso and Ridge), the most informative features were selected. The SHAP analysis revealed that longer essays with richer lexical variety, clearer cohesion, and fewer grammatical errors were consistently rated higher by both human and automated systems. This provided not just a mechanism for score prediction, but a foundation for pedagogically meaningful feedback.

Despite these advances, a challenge remained: although SHAP offered detailed attribution scores, the model lacked the ability to transform those insights into actionable, human-readable feedback for students. Mok (2023) acknowledges that while these ex-

planations are useful to researchers and educators, additional work is needed to make them accessible and educationally impactful for learners.
This work forms a foundation for more recent research that aims to not only predict scores with transparency, but to deliver feedback that supports learning. In doing so, it repositions the role of the AES system — from a passive grader to an active participant in the learning process. The table below summarizes this study.

Aspect	**Details**
Study Objective	Develop an explainable deep learning model to score essays and highlight linguistic features affecting grades
Dataset Used	ASAP Grade 7 Narrative Essays (1,567 essays, scored on 0–3 scale across four rubrics)
Model Type	Multi-Layer Perceptron (MLP) with 2–6 hidden layers
Input Features	1,592 linguistic features extracted via SALAT toolkit (e.g., cohesion, syntax, grammar)
Feature Selection	Lasso & Ridge Regression; low-variance filtering
Explainability Technique	SHAP (KernelSHAP, DeepSHAP, GradientSHAP)
Top Features for High Scores	• Essay length • Lexical diversity • Sentence complexity • Cohesive ties • Correct grammar usage
Key Strength	SHAP provides global and local explanations; reveals what textual features matter most
Main Limitation	Explanations not yet in student-readable, feedback-ready form

Table 2.1: Summary of Mok (2023) — Explainable AES with SHAP Analysis

2.1.2 Feedback Generation and Instructional Support via Large Language Models

Recent research has moved beyond simply using LLMs to assign grades, exploring their potential to deliver personalized feedback — a critical element in promoting student learning. Feedback not only clarifies why a response was graded a certain way but also guides students on how to improve. Studies now show that, with appropriate guidance, LLMs can simulate this instructional role.

In one study, Poličar et al. (2025) applied LLMs in a bioinformatics course to grade written assignments and provide feedback. Six models were tested, including commercial options (GPT-4, GPT-3.5, Claude) and open-source alternatives (LLaMA 2, Falcon 40B). Using a blind evaluation, students rated feedback without knowing its source. Notably, feedback from LLMs was often rated as helpful as that from human teaching assistants. Open-source models performed comparably to commercial ones, suggesting viable, privacy-conscious pathways for educational deployment.

This success relied on prompt engineering: feeding the model structured inputs like the correct answer, grading rubrics, and example responses. This enabled models to produce relevant, contextualized feedback without the need for fine-tuning.

Similarly, Impey et al. (2024) investigated GPT-4's grading capabilities in science writing. Given a rubric and model answer, GPT-4 reliably scored and explained its decisions, demonstrating instructional potential across domains.

Both studies emphasize that when guided properly, LLMs can offer explanations and suggestions — not just evaluations. However, limitations remain. LLMs may overlook subtle domain-specific issues, and the quality of their feedback heavily depends on prompt clarity and structure.

Aspect	**Poličar et al. (2025)**	**Impey et al. (2024)**
Educational Context	Bioinformatics course (university-level, ¿100 students)	MOOCs in astronomy and astrobiology (Coursera platform)
Task Type	Written explanations of data analyses in assignments	Short essay-style answers to open-ended science questions
Models Evaluated	GPT-4, GPT-3.5, Claude, LLaMA 2 (13B, 70B), Falcon 40B	GPT-4
Grading Setup	Blind evaluation comparing LLM feedback to human TA feedback	Comparison to instructor grades; GPT-4 guided with rubric and model answer
Feedback Approach	Prompt engineering using rubrics, sample answers, and solutions	Rubric-based grading and rubric generation by the model
Key Findings	LLM feedback rated equal to human TAs; open-source models performed well	GPT-4 grading reliable and more consistent than peer grading
Limitations	Occasional domain-specific misses; dependent on prompt quality	Performance tied closely to rubric design; limited subject scope

Table 2.2: Summary of Studies on LLM-Based Feedback Generation

2.1.3 Scaling Automated Grading with Active Learning and Data Efficiency

One of the major limitations of early automated grading systems was the reliance on large, hand-labeled datasets. For each new exam question or assignment, new data had to be collected and annotated by human graders — a process that is expensive, time-consuming, and difficult to maintain across academic years or changing syllabi.

To address this, several studies have explored active learning and weak supervision as strategies to reduce labeling requirements while maintaining model performance. These approaches aim to identify the most informative student submissions for human annotation, so that models can learn effectively from fewer examples.

A foundational contribution to this area is the Active Learning Literature Survey by Settles (2010), which outlines a taxonomy of strategies such as uncertainty sampling, query-by-committee, and expected error reduction. These techniques are highly relevant in educational contexts, where human labeling is costly and scalability is essential. The survey laid the groundwork for newer applications of active learning in grading tasks.

Building on these principles, Firoozi et al. (2023) applied active learning directly to the task of Automated Essay Scoring (AES). Their study evaluated three methods for selecting student essays for annotation:

- Uncertainty-Based Selection (selecting essays where the model is least confident),
- Topological-Based Selection (selecting a diverse, representative sample),
- and a Hybrid Method, which combines both uncertainty and representativeness.

The results were significant: using just 1.8% of the total dataset with the topological method, their model achieved 95% of full-model accuracy. The hybrid method provided slightly lower accuracy but greater stability across sample sizes. The study also demonstrated the effectiveness of fine-tuning BERT on strategically selected essays — achieving strong grading performance with far fewer labels.

These findings suggest that LLM-powered grading systems do not need large training sets, especially when paired with smart sampling strategies. For institutions with limited resources, these methods offer a practical path forward: instructors only need to grade a small portion of responses, which are then used to train models that generalize to the rest of the cohort.

Aspect	**Settles (2010)**	**Firoozi et al. (2023)**
Contribution Type	Survey of active learning strategies	Applied active learning to AES using real essay data
Context	General ML framework (text, image, bioinformatics, etc.)	AES on ASAP dataset with $\sim$13k student essays
Methods Evaluated	Uncertainty sampling, query-by-committee, error reduction	Uncertainty-based, topological, and hybrid selection
Models Used	Conceptual (broad ML)	BERT-based AES model
Key Findings	Active learning reduces labeling needs in various domains	1.8% of data yielded 95% accuracy using topological selection
Implications for Grading	Foundation for later AES strategies	Effective grading at scale with minimal human-annotated data
Limitations	Not grading-specific	Model performance depends on sample quality and essay prompt complexity

Table 2.3: Summary of Studies on Scaling AES with Active Learning

2.1.4 Designing Prompts That Teach the Model how to Grade

. Unlike traditional machine learning models, which often require large datasets and model retraining, Large Language Models (LLMs) like GPT-4 and Claude offer a flexible alternative through prompt engineering. Rather than changing the model's architecture or weights, researchers modify the input instructions — giving the model examples, rubrics, and specific grading criteria — in order to guide its behavior.

This approach is especially valuable in educational settings where questions, learning outcomes, and grading schemes often change. Rather than training a new model every year, instructors can adapt their prompts to fit new assessment contexts.

The study by Bengtsson and Kaliff (2024) investigated this strategy in the context of a university-level programming course. Students submitted answers to open-ended programming questions, and GPT-4 was used to assess their responses. The model was not fine-tuned; instead, it was guided using structured prompts that included:

- The expected solution,
- Marking criteria,
- And example answers.

This allowed GPT-4 to evaluate correctness, partial understanding, and even misconceptions, much like a human grader. Importantly, the study found that the LLM's scores aligned closely with instructor assessments, indicating that a well-structured prompt can replicate domain-specific human grading with high reliability.

In a similar direction, Impey et al. (2024) explored whether GPT-4 could generate its own grading rubric for short-answer science responses. In this case, the model was first given a model answer and examples of correct responses. It then generated both the rubric and the grades for unseen answers. The model's output was consistent with instructor grades, and its explanations added clarity to the assessment. However, the authors noted that the model's grading quality depended heavily on how clearly the prompt was written, reinforcing the importance of human guidance in the prompt design process.

These findings show that prompt engineering functions as a low-cost, high-impact alternative to model retraining. It empowers educators to adjust LLM behavior across subjects and cohorts, while maintaining grading consistency and interpretability. However, this method does require pedagogical expertise: poorly designed prompts can introduce ambiguity or bias, particularly in highly technical domains.

2.1.5 A Subsection about Citation Style

Citations are important. Citation style for Computer Science is:

- When used in the text, use the authors with the date in brackets:
  Klein and Celik [2017] say very important things.
- When used as a reference after a face, put everything in brackets:
  Import things are true [Klein and Celik 2017].

Aspect	Bengtsson & Kaliff (2024)	Impey et al. (2024)
Context	Programming assessments (university-level CS course)	Short-answer science responses (astronomy MOOCs)
Model Used	GPT-4	GPT-4
Technique	Prompt engineering with expected answers and rubrics	Prompted GPT-4 to generate grading rubric and assign scores
Evaluation	Compared to instructor scores	Compared to instructor grading and peer evaluations
Findings	High agreement with human graders; accurate on domain concepts	GPT-4 reliable, but sensitive to prompt clarity
Limitations	Requires domain-specific prompt design	Rubric generation quality varies with input phrasing

Table 2.4: Summary of Studies Using Prompt Engineering and Rubric Conditioning

2.1.6 Compiling

Remember to compile multiple times to resolve references. Usually:

```
pdflatex file.tex
bibtex file
pdflatex file.tex
pdflatex file.tex
```

Chapter 3

Floats

LaTeX decides how to place images. It also does the referencing for you as seen in Figure 3.1. If you have subimages, they should have their own captions and labels – look into the subfig or subfigure packages.

Figure 3.1: This is an image

Figure captions are at the bottom. Table title are at the top of the table as seen in Table 3.1 on the following page. There is a package called BookTabs which is *way* better for tables and you should learn how to use that instead.

Usually let LaTeX handle the placement of floats unless you *really* need to force it to do something else. The `float` package used above allows you to use `H` as the placement which means *here and only here*. When using the float package, the placement options are:

1. h – a gentle nudge to place it here if possible
2. t – top of a page
3. b – bottom of a page
4. H – here and only here, do not move it at all
5. p – on its own page

Table 3.1: Table Name

Col1	Col2
R0,C0	R0,C1
R1,C0	R1,C1

Chapter 4

Some Referencing Tricks

CleverRef and VarioRef are helpful:

- Normal Ref: See Figure 3.1
- CleverRef: See Figure 3.1 and Table 3.1
- CleverRef+VarioRef: See Figure 3.1 on page 9 and Table 3.1 on the facing page

Chapter 5

IDE/Editors

Overleaf has a great online editor for latex. Use it.

Appendix A

Extra Stuff

A.1 What is an appendix?

An appendix is useful when there is information that you need to include, but breaks the flow of your document, e.g. a large number of figures/tables may need to be shown, but maybe only one needs to be in the text and the rest are just included for completeness.

References

[Klein and Celik 2017] R. Klein and T. Celik. The Wits Intelligent Teaching System: Detecting student engagement during lectures using Convolutional Neural Networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2856–2860, Sep. 2017.