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?

1.2 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

1.3 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.

1.4 Body.

1.4.1 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 explanations 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 1.2: Summary of Mok (2023) — Explainable AES with SHAP Analysis

1.4.2 Feedback Generation and Instructional Support via Large Language Models

1.4.3 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].

1.4.4 Compiling

Remember to compile multiple times to resolve references. Usually:

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

Chapter 2

Floats

LaTeX decides how to place images. It also does the referencing for you as seen in Figure 2.1. If you have subimages, they should have their own captions and labels – look into the subfig or subfigure packages.

Figure 2.1: This is an image

Figure captions are at the bottom. Table title are at the top of the table as seen in Table 2.1 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 2.1: Table Name

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

Chapter 3

Some Referencing Tricks

CleverRef and VarioRef are helpful:

- Normal Ref: See Figure 2.1
- CleverRef: See Figure 2.1 and Table 2.1
- CleverRef+VarioRef: See Figure 2.1 on page 5 and Table 2.1 on the facing page

Chapter 4

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.