DEVELOPING AN AI-ASSISTED GRADING SYSTEM USING LARGE LANGUAGE MODELS

by

Andrei Modiga

A PROJECT PROPOSAL

Presented to the Faculty of

The School of Computing at the Southern Adventist University

In Partial Fulfilment of Requirements

For the Degree of Master of Science

Major: Computer Science

Under the Supervision of Professor Anderson

Collegedale, Tennessee

August, 2024

DEVELOPING AN AI-ASSISTED GRADING SYSTEM USING LARGE LANGUAGE MODELS

Andrei Modiga, M.S.
Southern Adventist University, 2024

Adviser: Scot Anderson, Ph.D.

Grading written student work is a critical component of education, directly influencing the learning experience and providing essential feedback to both students and educators. This project focuses on building a high-level AI system that leverages Large Language Models (LLMs) to automate and enhance the grading process. Specifically, the system classifies student responses by semantic similarity, streamlining the grading process and allowing educators to provide more consistent and efficient feedback. By developing this AI-assisted grading system, the project aims to reduce the manual workload on teachers, improve the consistency of grading, and provide timely feedback to students.

Contents

Contents v

List of Figures ix

List of Tables xi

1 Introduction 1

2 Background 3

2.1 AI-Assisted Grading Techniques 3

2.1.1 Automated Grading of Handwritten Responses 4

2.1.2 Challenges in Grading Handwritten Exams with AI 4

2.1.3 Advancements in Large Language Models for Grading 5

2.1.4 Reducing Grading Workload with Machine Learning 5

2.2 Summary 6

3 Proposal 7

3.1 Summary of Requirements 7

3.1.1 OICLearning Grading Process 8

3.1.2 Filled-Form Assignments 9

3.1.3 Free-Form Written Assignments 10

3.1.4 Objective-Form Assignments/Assessments . . . 11
3.2 Proposed Approach . . . 12
3.2.1 Python-Based AI Grading System . . . 12
3.2.2 Web Application Integration . . . 13
3.2.3 Beta and User Testing . . . 13
3.2.4 Final Changes, Bug Fixes, Testing, and Documentation . . . 14
3.3 Final Deliverables . . . 14
3.4 Development Prerequisites . . . 15
3.5 Task Delineation . . . 16
3.5.1 Task Breakdown . . . 16
3.5.2 Timeline . . . 17

4 Testing and Evaluation Plan **19**
4.1 Overview . . . 19
4.2 Testing Phases . . . 19
4.2.1 Unit Testing . . . 19
4.2.2 Integration Testing . . . 20
4.2.3 System Testing . . . 20
4.2.4 User Acceptance Testing . . . 20
4.3 Evaluation Metrics . . . 20
4.3.1 Accuracy . . . 20
4.3.2 Efficiency . . . 21
4.3.3 Usability . . . 21
4.3.4 Scalability . . . 21

5 Conclusion **23**

A Requirements Specification **25**

A.1 Task Delineation 25

A.2 Hardware Requirements 25

A.3 Application Requirements 26

A.3.1 General 26

A.3.2 Download and Installation 26

A.3.3 Opening and Managing Assignments 26

A.3.4 Presentation Mode 26

A.3.5 Edit Mode 26

A.3.6 Settings 27

B Optional Requirements **29**

Bibliography **31**

List of Figures

List of Tables

Chapter 1

Introduction

In the educational landscape, grading written student work is a task of high importance, directly influencing the learning experience and providing crucial feedback to both students and educators. While traditional grading methods are generally effective, they can be time-consuming and labor-intensive, particularly for teachers managing large volumes of student responses. The need for efficient and scalable grading solutions has become increasingly evident as educators seek to streamline their workloads without compromising the quality of their assessments.

Recent advancements in artificial intelligence, particularly in the development of Large Language Models (LLMs), offer a promising approach to this challenge. LLMs, such as OpenAI's GPT-4, have shown remarkable proficiency in understanding and generating human-like text by analyzing vast datasets and recognizing patterns in language. These models have the potential to transform the grading process by automating the evaluation of student responses, thereby reducing the workload on educators and providing quicker feedback to students.

This project explores the development of an AI-assisted grading system that leverages LLMs to classify student responses by grouping those that are seman-

tically similar. By identifying clusters of answers that convey the same meaning, regardless of phrasing or structure, the system can assist educators in grading more efficiently while maintaining the integrity of their evaluations.

The primary goal of this project is to build and implement this AI-assisted grading system, including the ability to recognize handwritten text, and integrate it into an educational platform like OICLearning.com. By focusing on building a practical solution, the project aims to enhance the grading process, making it quicker and more consistent, thereby allowing teachers to focus on more critical aspects of instruction while ensuring that students receive fair and accurate feedback.

However, the integration of LLMs into the grading process presents certain challenges that must be carefully addressed. One of the key considerations is the model's ability to accurately differentiate between nuanced meanings and appropriately group responses that, while semantically similar, may vary in their correctness. Additionally, the effectiveness of these models in handling a diverse range of student expressions and potential errors needs to be thoroughly evaluated.

Chapter 2

Background

Emerging research in AI-assisted grading is rapidly evolving, especially in handling complex tasks such as handwritten responses in educational assessments. This chapter explores studies that demonstrate both the effectiveness and limitations of using AI technologies, such as Large Language Models (LLMs), in automating the grading process. The research further highlights the gap that the proposed AI system for OICLearning.com will address.

2.1 AI-Assisted Grading Techniques

Recent advancements in artificial intelligence have introduced new possibilities for automating the grading of student work, particularly in handling diverse response types, including handwritten solutions. These systems aim to reduce manual grading workload while maintaining accuracy and fairness. Below are the techniques that are relevant to the proposed project.

2.1.1 Automated Grading of Handwritten Responses

AI tools, such as MathPix and GPT-4, have been leveraged to grade handwritten responses in exams. For instance, Liu et al. (2024) [3] demonstrate the effectiveness of using a pre-trained GPT-4 model to evaluate semi-open handwritten responses in university-level mathematics exams. Their study suggests that such systems can offer reliable and cost-effective initial grading, although human verification is still essential to ensure accuracy and reliability.

Similarly, Kortemeyer (2023) [1] conducted a feasibility study on grading handwritten physics derivations using MathPix for scanning handwritten solutions and GPT-4 for grading. The study found an R-squared value of 0.84 when compared to human graders, showing that AI tools can effectively assist in providing formative feedback. However, the study concluded that AI should still complement human graders, particularly in final evaluations, due to the complexity and diversity of student responses.

2.1.2 Challenges in Grading Handwritten Exams with AI

Grading handwritten exams presents several challenges. Converting handwritten text into a machine-readable format is one of the primary obstacles, as noted by Kortemeyer et al. (2024) [2]. For example, in grading thermodynamics exams, they found that using fine-grained rubrics for entire problems often led to errors, while grading parts of problems separately was more reliable but tended to miss nuanced details. This is particularly true for hand-drawn graphics, such as process diagrams, where AI struggles to distinguish essential content from extraneous information.

This highlights the need for developing robust AI models capable of accurately

parsing handwritten responses while ensuring high precision in scoring complex visual and textual content. These challenges need to be addressed when designing the grading system for OICLearning.com, which aims to automate both online and scanned assignment grading.

2.1.3 Advancements in Large Language Models for Grading

The use of LLMs, such as GPT-4, marks a significant advancement in AI-assisted grading. These models have shown promise in understanding and generating human-like text, making them suitable for recognizing patterns in student responses and grouping answers by semantic similarity. For instance, Liu et al. (2024) [3] evaluated GPT-4's ability to handle diverse response types and found that it can reliably assess even semi-open responses.

However, integrating LLMs into grading processes comes with its own challenges. One of the key issues is ensuring that the AI can accurately differentiate between nuanced meanings in student responses. Another challenge is enhancing the system's ability to process handwritten responses, which remains a significant hurdle in automating grading effectively.

2.1.4 Reducing Grading Workload with Machine Learning

Weegar and Idestam-Almquist (2024) [**?**] investigated machine learning methods for grading short answers in computer science exams. Their research found that even with manual review, the workload could be reduced by up to 74%. This reduction was achieved through a two-step grading process involving clustering of similar answers and automated scoring. This kind of workload reduction is a key goal of the proposed AI grading system for OICLearning.com.

2.2 Summary

In conclusion, while AI technologies such as GPT-4 and MathPix offer promising tools for automating the grading process, there are still significant challenges to overcome, particularly in grading handwritten responses and ensuring accuracy. The proposed system for OICLearning.com will incorporate lessons learned from these studies, using LLMs to automate the grading of various assignment types and reduce manual effort for educators.

Chapter 3

Proposal

This chapter outlines the key requirements for developing an AI-assisted grading system using Large Language Models (LLMs) like GPT-4, implemented in Python. The goal is to automate the grading process for various types of assignments on OICLearning.com and eventually incorporate this system into a web application for broader accessibility and usability.

3.1 Summary of Requirements

The project aims to automate the grading process for different types of assignments submitted on OICLearning.com. The system will utilize LLMs for grading assignments, both scanned and online submissions, using a combination of Python-based AI tools and later integration into a web app. The following sections describe the manual grading process requirements and how the AI-based grading system will be developed to fulfill these needs.

3.1.1 OICLearning Grading Process

OICLearning.com does not currently have an existing grading process in place. The proposed AI-assisted grading system will be implemented from the ground up to handle various types of assignments, both scanned and online submissions. The system will streamline the grading process for educators by utilizing LLMs and automating key tasks that would otherwise require manual effort.

3.1.1.1 Big Picture Process

The current manual grading process involves several steps:

1. A teacher creates an assignment on OICLearning.com.
2. A student submits an assignment.
3. The teacher grades the assignment.
4. The teacher copies the grades to the Learning Management System (LMS).

The new AI-assisted system will automate steps 3 and 4, significantly reducing the time and effort required from educators.

3.1.1.2 Creating Assignments

Assignments on OICLearning.com fall into three categories:

1. **Filled-Form Scanned Assignments**: These are written tests that students fill out manually and are later scanned and uploaded.
2. **Free-Form Written Assignments**: These assignments have flexible content, and the location of questions and answers may vary.

3. **Filled-Form Online Assignments**: These are completed online directly through the platform.

The AI system must be able to handle grading for all three types of assignments.

3.1.2 Filled-Form Assignments

For filled-form assignments, the AI system must meet the following requirements:

3.1.2.1 Requirements

1. The assignment is printed as a form by the teacher for the students to fill out.
2. A PDF of the assignment is pre-loaded into OICLearning.
3. The system must identify the locations of Name and ID number (or Email).
4. Question and Answer locations are identified by the teacher using a user-friendly interface to draw rectangles on the pre-loaded PDF, specifying the question number (e.g., 1.a., 1.b., etc.) and points.

The system should support both teacher and student uploads of PDFs, automatically assigning them to the correct students and handling various PDF lengths.

3.1.2.2 Process for Grading Filled-Form Assignments

1. After uploading the assignment, the AI identifies the student using handwriting recognition.
2. Upon accessing the grading page, the AI lists questions and whether they are graded.

3. The teacher selects a question to grade or review, and the AI starts grouping answers, which can be manually adjusted by the teacher.

4. Default grouping by AI includes all answers. Teachers can create additional groups, drag and drop answers, rename groups, and override AI decisions.

5. Once groups are finalized, the teacher grades each group by selecting representative answers, adding comments, and assigning points.

6. The grades are then applied to all members of each group, and the teacher proceeds to the next question.

7. Once all questions are graded, a final score is recorded for each student.

3.1.3 Free-Form Written Assignments

Free-form assignments require a slightly different approach due to their variability.

3.1.3.1 Requirements

1. Teachers must identify Question and Answer pairs, specifying question numbers and points.

2. Assignments are uploaded as PDFs, and the AI handles varying lengths by grouping them appropriately for each student.

3. The AI system allows for manual marking of each answer within the PDF, flagging unanswered questions, and overwriting old data if new PDFs are submitted before the deadline.

3.1.3.2 Process for Grading Free-Form Assignments

1. The teacher uploads the assignment, and the AI processes the content to identify question locations.
2. Answers are grouped by the AI, with manual adjustments available to the teacher.
3. Teachers can add comments and assign points as they grade each group.
4. Grades are automatically applied to all students in each group, with a final score calculated upon completion.

3.1.4 Objective-Form Assignments/Assessments

Objective-form assignments are distinct and require a focus on grouping readability and markdown input for mathematical content.

3.1.4.1 General Issues and Requirements

1. Answers must be legible, with the ability to scroll to view all responses.
2. Blank responses should be automatically grouped under "Not Submitted."
3. Multiple answers can be selected and grouped at once, facilitating batch processing.
4. Markdown input is required, including support for LaTeX for complex mathematical expressions.
5. The system must isolate selected PDF areas for grading, potentially converting these to images for clarity.

3.2 Proposed Approach

The AI-assisted grading system will be developed using Python and integrated with LLMs like GPT-4 to automate the grading process. The initial phase involves developing a Python-based backend that interacts with scanned PDFs and online submissions to recognize and categorize student responses. The long-term goal is to integrate this backend into a web application to provide an intuitive user interface for teachers and students.

3.2.1 Python-Based AI Grading System

The Python-based backend will use OCR (Optical Character Recognition) for scanned assignments and natural language processing capabilities of LLMs for understanding and grading responses.

3.2.1.1 Development Steps

1. Develop OCR functionality to accurately convert scanned assignments into text format using tools like Tesseract OCR or commercial alternatives.
2. Integrate GPT-4 models to understand and categorize student responses based on semantic similarity.
3. Implement algorithms to group similar responses together for efficient grading.
4. Develop a database system to store student responses, grades, and feedback securely.
5. Implement a user interface for teachers to upload assignments, view AI-generated groupings, and adjust them as needed.

6. Develop grading algorithms to assign points and provide feedback, incorporating both AI and manual grading options.

7. Test the system with sample data to ensure accuracy and reliability.

3.2.2 Web Application Integration

Following the backend development, the system will be integrated into a web application.

3.2.2.1 Integration Steps

1. Develop a web-based front end using frameworks like React or Angular.

2. Connect the Python backend with the web front end to allow seamless interaction for users.

3. Implement authentication and role management to secure the application and ensure only authorized users can access grading features.

4. Incorporate APIs to facilitate communication between the frontend, backend, and LLM services.

5. Test the web application with real users to gather feedback and improve usability.

3.2.3 Beta and User Testing

The application will undergo thorough beta testing to ensure all features function as intended. This phase includes:

- **Functionality Testing**: All grading features and functionalities are rigorously tested to ensure they work correctly.
- **Performance Optimization**: The system is optimized for speed and efficiency, ensuring quick response times during grading.
- **User Testing**: Real users, such as teachers and students, will test the application to provide feedback on usability and functionality.
- **Compatibility Testing**: The system is tested across various devices and browsers to ensure compatibility and smooth performance.

3.2.4 Final Changes, Bug Fixes, Testing, and Documentation

After beta testing, the final task includes:

- Fixing any identified bugs and making necessary optimizations.
- Completing thorough testing to ensure all remaining issues are resolved.
- Creating comprehensive documentation for users and developers, detailing the system's functionalities, usage, and development setup.
- Preparing the application for deployment on OICLearning.com and future integration into other platforms if needed.

3.3 Final Deliverables

Once the project is complete, the following deliverables will be provided:

- **Application Source Code**: The full source code for the AI-assisted grading system.

- **Administrative Access to Project Source Control**: Access to the code repository for ongoing maintenance and updates.
- **User and Developer Documentation**: Detailed guides on the application's general structure, codebase, and usage instructions.
- **Testing and Evaluation Report**: A comprehensive report on all testing phases, including beta testing results and user feedback.
- **Final Project Report**: A detailed report summarizing the project's goals, implementation, challenges, and outcomes.

3.4 Development Prerequisites

To develop and test the AI-assisted grading system, the following hardware and software are required:

- **Hardware**: High-performance servers or cloud instances with GPU support for running LLMs, modern computers for development, and various devices for compatibility testing (e.g., tablets, smartphones).
- **Software**: Python, TensorFlow/PyTorch for AI model implementation, React/Angular for front-end development, and Git for version control.
- **Tools**: OCR software for text recognition in scanned documents, database management systems (e.g., PostgreSQL), and cloud services for deployment and scaling.
- **Access to LLMs**: Licenses or API access to GPT-4 models for integration and testing.

3.5 Task Delineation

This section outlines the specific tasks and timelines required to complete the project.

3.5.1 Task Breakdown

1. **Requirement Analysis and Planning** (2 weeks)
2. **Backend Development** - Develop and test the Python-based AI grading system (months).
3. **Frontend Development** - Develop the web application interface and integrate it with the backend (months).
4. **System Testing** - Perform extensive testing of both backend and frontend with sample data (month).
5. **Deployment and Feedback** - Deploy the application on OICLearning.com and gather user feedback (month).
6. **Documentation and Final Reporting** (weeks)

3.5.2 Timeline

Task	Duration
Requirement Analysis and Planning	weeks
Backend Development	months
Frontend Development	months
System Testing	month
Deployment and Feedback	month
Documentation and Final Reporting	weeks

Chapter 4

Testing and Evaluation Plan

4.1 Overview

This chapter outlines the plan for testing and evaluating the AI models used in the grading process. The goal is to ensure the system's accuracy, reliability, and usability.

4.2 Testing Phases

4.2.1 Unit Testing

Individual components of the system will be tested to ensure they function correctly in isolation. This includes testing the OCR module, the LLM integration, and the database operations.

4.2.2 Integration Testing

Testing will focus on ensuring that all components work together seamlessly within the overall system. The interaction between the frontend, backend, and AI models will be thoroughly tested.

4.2.3 System Testing

Comprehensive testing of the entire system will be conducted to ensure all features function as expected. This phase includes stress testing and performance benchmarking.

4.2.4 User Acceptance Testing

The system will be tested by end-users, including teachers and students, to ensure it meets their needs and expectations. Feedback collected will be used to make necessary adjustments.

4.3 Evaluation Metrics

4.3.1 Accuracy

The accuracy of the AI models will be evaluated based on their ability to correctly grade student responses and group them appropriately. Metrics such as precision, recall, and F1-score will be used.

4.3.2 Efficiency

The time required to grade assignments using the AI system will be compared to manual grading to assess efficiency gains.

4.3.3 Usability

User feedback will be collected through surveys and interviews to evaluate the usability of the system and identify areas for improvement.

4.3.4 Scalability

The system's performance will be tested under different workloads to ensure it can handle a large number of users and assignments without degradation in performance.

Chapter 5

Conclusion

This project aims to develop an AI-assisted grading system that leverages Large Language Models (LLMs) to automate and enhance the grading process. By focusing on building and implementing the system, the project seeks to streamline grading for educators and improve the educational experience for students. The successful development and integration of this system will contribute to the field of AI in education by providing a practical solution to a common challenge faced by educators.

Appendix A

Requirements Specification

A.1 Task Delineation

Detailed tasks as outlined in the Proposal chapter, providing a clear roadmap for development.

A.2 Hardware Requirements

- add here

A.3 Application Requirements

A.3.1 General

The application should be user-friendly, secure, and scalable, with support for multiple assignment types and integration with existing educational platforms.

A.3.2 Download and Installation

Instructions for setting up the development environment and deploying the application on servers.

A.3.3 Opening and Managing Assignments

Features for educators to create, upload, and manage assignments, as well as view and grade student submissions.

A.3.4 Presentation Mode

A user interface that allows educators to view AI-generated groupings of student responses and adjust them as needed.

A.3.5 Edit Mode

Functionality for educators to modify assignment details, grading criteria, and provide feedback to students.

A.3.6 Settings

Options to configure application settings, user preferences, and integration with other systems.

Appendix B

Optional Requirements

Additional features that may enhance the system, such as:

- add here

Bibliography

[1] Gerd Kortemeyer. Toward ai grading of student problem solutions in introductory physics: A feasibility study, November 2023. 2.1.1

[2] Gerd Kortemeyer, Julian Noh, and Daria Onishchuk. Grading assistance for a handwritten thermodynamics exam using artificial intelligence: An exploratory study, June 2024. 2.1.2

[3] Tianyi Liu, Julia Chatain, Laura Kobel-Keller, Gerd Kortemeyer, Thomas Willwacher, and Mrinmaya Sachan. Ai-assisted automated short answer grading of handwritten university level mathematics exams, August 2024. 2.1.1, 2.1.3

[4] Rebecka Weegar and Peter Idestam-Almquist. Reducing workload in short answer grading using machine learning, June 2024.