

Automatic exam essay grader with artificial intelligence feedback

Final Project Report

TU857

BSc in Computer Science (Infrastructure)

**Joseph Egan**

**C20389531**

**Supervisor: Dr Aneel Rahim**

School of Computer Science

Technological University, Dublin

**Date: 11/04/2025**

Abstract

Since 2016, there has been a 13.8% increase in the number of enrolled students in Irish universities (HEA Statistics), reflecting a growing student population. This increase has contributed to a widening lecturer-to-student ratio problem in Ireland. Currently, Ireland has a student-to-staff ratio of 22.38:1 (OECD Data), significantly higher than the OECD average of 15.58:1. This disparity places additional pressure on lecturers, particularly for labour-intensive tasks such as grading essays and providing personalised feedback.

Detailed feedback on written work is essential for students, as it supports their academic growth and development. However, due to increased workloads, lecturers may struggle to offer the level of individualised feedback that helps students improve key skills such as writing and critical analysis.

An AI-assisted essay grading and feedback system could help address this issue. By automating aspects of the grading process, such a system can provide timely, constructive feedback on elements such as grammar, argument structure, and coherence. This would enable lecturers to focus on more complex, personalised teaching while ensuring students receive consistent, high-quality feedback. Implementing an AI-driven system in this context could enhance the learning experience and help educators manage the demands of increasing class sizes across Irish higher education.

Declaration

I hereby declare that the work described in this dissertation is, except where otherwise stated, entirely my own work and has not been submitted as an exercise for a degree at this or any other university.

Signed:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Joseph Egan

Date: 11/04/2025

Acknowledgements

I would like to thank my supervisor Dr Aneel Rahim for his guidance throughout the course of the project. I am also grateful to my family for supporting me.

Table of Contents

[1. Introduction 7](#_Toc157603088)

[1.1. Project Background 7](#_Toc157603089)

[1.2. Project Description 7](#_Toc157603090)

[1.3. Project Aims and Objectives 7](#_Toc157603091)

[1.4. Project Scope 7](#_Toc157603092)

[1.5. Thesis Roadmap 7](#_Toc157603093)

[2. Literature Review 8](#_Toc157603094)

[2.1. Introduction 8](#_Toc157603095)

[2.2. Alternative Existing Solutions 8](#_Toc157603096)

[2.3. Technologies Researched 8](#_Toc157603097)

[2.4. Other Relevant Research 8](#_Toc157603098)

[2.5. Existing Final Year Projects 8](#_Toc157603099)

[2.6. Conclusions 8](#_Toc157603100)

[3. Experiment / Software Design 9](#_Toc157603101)

[3.1 Introduction 9](#_Toc157603102)

[3.2. Software Methodology 9](#_Toc157603103)

[3.3. Requirements Gathering 9](#_Toc157603104)

[3.4. Overview of System 9](#_Toc157603105)

[3.X. Oher Sections 9](#_Toc157603106)

[3.X. Conclusions 9](#_Toc157603107)

[4. Experiment / Software Development 10](#_Toc157603108)

[4.1. Introduction 10](#_Toc157603109)

[4.2. Software Development 10](#_Toc157603110)

[4.X. Oher Sections 10](#_Toc157603111)

[4.X. Conclusions 10](#_Toc157603112)

[5. Testing and Evaluation 11](#_Toc157603113)

[5.1. Introduction 11](#_Toc157603114)

[5.2. System Testing 11](#_Toc157603115)

[5.3. System Evaluation 11](#_Toc157603116)

[5.4. Conclusions 11](#_Toc157603117)

[6. Conclusions and Future Work 12](#_Toc157603118)

[6.1. Introduction 12](#_Toc157603119)

[6.2. Future Work 12](#_Toc157603120)

[6.3. Conclusions 12](#_Toc157603121)

[Bibliography 13](#_Toc157603122)

[Appendices 14](#_Toc157603123)

# 1. Introduction

This section will cover the introduction, description, and the scope of my project.

## Project Background

In the modernising world there is a growing need for third level education in degrees to enter the working world as can be seen through the education by generation graph below. This is causing issues and strains on the Irish University education system.

A graph of a number of people

AI-generated content may be incorrect.

(<https://www.cso.ie/en/releasesandpublications/ep/p-cpp8/censusofpopulation2022profile8-theirishlanguageandeducation/levelofeducation/>)

## Ireland was one of the founding members of the Organisation of Economic Co-operation and Education (OECD) in 1960 and currently faces significant challenge with its staff-to-student ratios with one of the worst students to lecturer ratios of the current organization members. This means that lecturers are, relatively speaking, dealing with more students compared to lecturers in other countries, for example Ireland has 22.38:1 student to lecturers whereas the United States has 13.63, and the UK has a ratio of 13.50. (TODO OCD data)

## This ratio imbalance creates excess strain on lectures, especially for labour intensive tasks such as grading essays and providing detailed personalised feedback. The growing number of Irish students and limited ability to increase lecture staff creates a need for tools that can solve this problem.

## Project Description

My automatic exam essay grader with AI feedback program allows students to get estimates of the quality of their exam practice and possibly points of feedback and advice on weaknesses and potential areas of improvements for their study. This system will use a combination of Bidirectional Encoder Representations from Transformers (BERT) and Long Short-Term Memory (LSTM) machine learning models to grade practice exam papers. The results of this paper and the grading rubric, and the paper will be passed on into an AI that will rewrite the grade and feedback into a more human readable format and with additional context.

## 

## Project Aims and Objectives

Overall aim and some milestones along the way to achieve the aim

* Developing and training machine learning models to assess essays based on predefined rubrics.
* Implementing a web-based interface allowing students to upload essays and select the appropriate grading model.
* Designing a backend system to manage grading models and facilitate interaction with the web interface.
* Integrating an AI-powered feedback system to refine grading results into structured, human-readable comments.
* Evaluating the effectiveness of the system against manual grading benchmarks and user feedback.

## Project Scope

The main focus of the project is the automated grading of essays using pre-trained machine learning models and using a pre-existing artificial intelligence like ChatGPT or Deep Seek to rewrite the feedback into a more human readable format and with additional context. It is not designed to write essays for students or help them get passed Ai or plagiarising checkers. The AI API is supposed to enhance the grading process, i.e. explain the results in personalised terms, not grade an essay on its own. This is not about building my own large language model artificial intelligence or any other type of artificial intelligence.

## Thesis Roadmap

One sentence summary of the following chapters

# Literature Review

This chapter provides a literature review on related works and technologies, including existing AI-based and machine learning based grading systems.

# Experiment / Software Design

This chapter discusses software design, methodology, and system architecture.

# Experiment / Software Development

This chapter outlines the development process and key implementation details.

# Testing and Evaluation

This chapter covers testing and evaluation, detailing system performance and user feedback.

# 6. Conclusions and Future Work

This chapter concludes the report and discusses potential areas for future development.

# 2. Literature Review

## 2.1. Introduction

In this chapter …

This chapter provides an overview of machine learning models, a brief overview of artificial intelligence and their applications in automated essay grading is provided in this section. It examines existing solutions, technologies researched, and relevant academic studies that inform this project’s approach. Additionally, a comparison with other AI grading systems highlights key strengths and limitations as well as the problems and challenges related to it.

There are already tools available to testing a person’s essay that can be freely used, examples such as ChatGPT at (https://chat.openai.com/) and the GitHub automated essay scoring project (https://github.com/sankalpjain99/Automatic-Essay-Scoring). These will be discussed in terms of comparison with the functionality available. There are also other suitable technologies that were researched during development including existing ai such as Chat GPT or Gemini, datasets, programming languages, and cloud providers.

The technologies that I have used and researched, and other research that I have put into for this project will also be discussed.

## 2.2. Alternative Existing Solutions

Various AI-powered essay grading tools already exist, offering different levels of automation and accuracy. Examples include:

* **ChatGPT**: A general-purpose AI capable of analysing and generating text but not specifically designed for structured grading.
* **GitHub’s Automated Essay Scoring Project**: An open-source initiative that uses machine learning to score essays based on predefined rubrics.
* **Turnitin Feedback Studio**: A widely used plagiarism detection and grading tool that provides automated feedback on writing mechanics but lacks deeper conceptual understanding.
* **Grammerly**: An English language writing assistant software tool. It reviews the spelling, grammar, and tone of a piece of writing as well as identifying possible instances of plagiarism. It can also suggest style and tonal recommendations to users and produce writing from prompts with its AI.

While these solutions demonstrate the potential of AI in education, they often fall short in terms of personalised feedback, adaptability to specific curricula, and transparency in scoring criteria. This project seeks to address these gaps by integrating explainable AI techniques and tailored feedback generation.

## 2.3. Technologies Researched

Programming languages, operating systems, etc.

As part of this I have investigated several different technologies which have been useful for

## 2.4. Other Relevant Research

Domain specific research

Scientific paper specific research Strategies for Deploying Unreliable AI Graders in High-Transparency High-Stakes Exams (“Strategies for Deploying Unreliable AI Graders in High-Transparency High-Stakes Exams,” n.d.), is a paper regarding possible ways to use a NPL AI grading system designed to grade midterm tests regarding high level explanations of code. This is a similar topic to main, in the context that we are both developing systems to grade students work, but we differ in core objects of or projects. They want to develop a system that would officially grade a student’s test instead of a human, such as a lecturer or teacher’s assistant, instead of my objective of producing a tool to assist students and lecturers grading practice exam papers. I feel my project covers the niche of a tool to assist lecturers and an exam preparation and self-improvement tool to assist students, whereas their paper discusses ways to develop a tool to replace humans from grading papers. College Exam Grader using LLM AI models (“College Exam Grader using LLM AI models,” n.d.), is a paper regarding an artificial intelligence exam grader based on ChatGPT-4.0. This is a project with the exact same goal as mine, Here, we propose an AI based exam grader that can not only ease educators’ burden but also produce accurate, consistent, and precise grading results.(“College Exam Grader using LLM AI models. Highlighted text,” n.d.). Their approach is different than mine, because they are using the ai to grade the exam paper and offer feedback, whereas I am using a machine learning model to grade the papers and the ai to personalise the feedback to the students. Automatic Grading of Computer Programs: A Machine Learning Approach (“Automatic Grading of Computer Programs: A Machine Learning Approach,” n.d.), is a paper regarding a machine learning model to assess the competency of a Programmer by grading their computer programs. This is similar to my project in the context that it uses a machine learning model to grade assessed work but differs in that it is used to grade the individual who write the programs through their programs. This is an example to learn how to implement a machine learning model, but it is more related to professional workplace application compared to my academic focused project.

## 2.5. Existing Final Year Projects

Title: Machine Learning for Predictive Analysis and Recommender System Student: Glory Pierce Eguare Description (brief): An application that generates a predictive algorithm for the fuel consumption efficiency of automotives What is complex in this project: The What technical architecture was used: Model template view architecture Explain key strengths and weaknesses of this project, as you see it. Strengths: The use of a detailed dataset auto-mpg, to train the machine learning algorithm. Weaknesses: The normalisation of the ranges of his dataset to facilitate easier training, may 12 have led to skewed data biases to his results. This would have disproportionally punished more specialised vehicles, i.e. an unusually heavy car would look like it would have unexpectedly terrible mpg and raise the value of more average cars.

Title: Generating expenditure tracking and analysis from hard copy receipts Student: Cillian Keohane Description (brief): A system to scan receipts and process them into an expenditure tracker What is complex in this project: The use of a Wikimedia API and the machine learning algorithm to process the receipts. What technical architecture was used: Model, View, and Controller architecture Explain key strengths and weaknesses of this project, as you see it. Strengths: the use of the scanner and photo processor, and the machine learning algorithm that adapts to the individual user and improves over time. Weakness: It is not synchronized to other banking devices to automate the process of taking in receipts.

## 2.6. Conclusions

# 3. Experiment / Software Design

## 3.1 Introduction

## 3.2. Software Methodology

The software methodology that I am currently choosing to use for my project is the Waterfall software methodology. I am choosing to use this methodology, because of its core principle of creating iterations of the project. The Waterfall methodology can be broken down into 5 phases.

* Phase 1: Planning and Requirements Analysis
* Phase 2: Design
* Phase 3: Development
* Phase 4: Testing
* Phase 5: Deployment

I am choosing to use the Waterfall methodology, because of the flexibility and continuous development aspects of the methodology. In order to improve and adapt the project over time during the completion of the project. I also chose the waterfall methodology due to the fact that I am making this project individually instead of in a group or collection of groups. My lack of partners means that I cannot benefit from using any other methodology.

## 3.3. Requirements Gathering

## 3.4. Overview of System

My automatic essay grader with AI feedback (AEGAF) program allows students to get estimates of the quality of their essays and possibly points of feedback and advice on weaknesses and potential areas of improvements for their essays. Students will choose what question they are answering and then post their essay in a doc or pdf format into my system. This system will use a combination of the grading rubric for that question and my machine learning model. The results of this essay, the grading rubric and the essay will be passed on into an AI that will write the grade and feedback into a more human readable format, for example you scored 80% on your essay, but need to focus on grammar and giving an example to support your point. The system will them post the result and feedback to the user.

## 3.X. Oher Sections

Include a diagram

## 3.X. Conclusions

# 4. Experiment / Software Development

## 4. Experiment / Software Development

## 4.1. Introduction

## 4.2. Software Development

## 4.X. Oher Sections

## 4.X. Conclusions

# 5. Testing and Evaluation

## 5.1. Introduction

The Testing and Evaluation section contains the testing and evaluation plan, process, and development of the project. It covers the methods used to test each module, details evaluation metrics, and outlines the approaches for assessing the overall performance and user experience.

## 5.2. System Testing

The methodology for the development of the project is a Waterfall approach. This involves performing development cycles over a development period, for example get a component done in one week. I aim to get units of the system, for example document input, essay analysis, rubric application, and feedback generation, programmed and then validation tested. I will then do system testing to validate the integration of the components. Automated test cases will be written to verify functional requirements, and simulated user scenarios will assess the system.

## 5.3. System Evaluation

Evaluation will focus on two key aspects:

1. Performance Metrics: The system’s speed, accuracy of grading, and relevance of feedback will be compared against manual grading benchmarks.
2. UserFeedback: I hope to get students and lecturers to provide input and performance evaluation on the system’s usability and the value of its feedback. This input will guide iterative improvements.

The system evaluation is further discussed in more detail in section 4 of this report.

## 5.4. Testing grading models

There are several tests, both primitive and automated, in my demo model. Some of them are primitive print statements, like the print("hello World") in the first section of my Model.ipynb file to test the interactive kernel or the print("Model trained and saved successfully!") are used to see if the model training is finished. Other tests are automated like the test for the mean absolute error used to test the accuracy of my model. These tests are important to the maintenance, usability, and functionality of my demonstration model.

## 5.5. Testing frontend View page of the website

There are several tests, both primitive and automated,

## 5.6. Testing Controller file of the website

There are several tests, both primitive and automated,

## 5.7. Testing backend Model file of the website

There are several tests, both primitive and automated,

## 5.8. Conclusions

# 6. Conclusions and Future Work

## 6.1. Introduction

This is the section

## 6.2. Future Work

* Development of a specialised artificial intelligence: For this project I used an API of an existing artificial intelligence. A potential future addition to the project could be the development of an open-source artificial intelligence trained for just this project.
* Expansion of database of exam paper models: I have only trained the models for 3 exam paper. The training of models for every past paper in TUD could be a potential future for this project.
* Development of mock papers: I made an example of a 2025 exam paper for Advanced Security 1 for demonstration for my Interim report in December. It is still used as an example in my project. An area of future development could be the creation of model specific mock papers for students to practice with.
* Creation of Generative Pre-trained Transformer machine learning models: My project currently only uses LSTM and BERT models for grading, but GPT models could also be made to aid in the grading process.

## 6.3. Conclusions

# Bibliography

# Appendices