Automated Essay Grading Model Using NLP and Machine Learning Approaches

Adeyinka Mariam Abiola

MSc Master of Science in Data Science
The University of Bath
2022/23

Automated Essay Grading Model Using NLP and Machine Learning Approaches

Submitted by: **Adeyinka Mariam Abiola**
for the degree of MSc in Data Science
at the University of Bath
September 2023

Copyright

Declaration

This dissertation is submitted to the University of Bath in accordance with the requirements of the degree of Master of Science in the Department of Computer Science. No portion of the work in this dissertation has been submitted in support of an application for any other degree or qualification of this or any other university or institution of learning. Except where specifically acknowledged, it is the work of the author.

Abstract

Automated essay grading (AEG) is a transformative innovation in education, revolutionizing the evaluation of written assignments for efficiency and consistency. This project delves into AEG, harnessing natural language processing, machine learning, and deep learning techniques. By utilizing a diverse array of features from the extensive Automated Student Assessment Prize (ASAP) dataset publicly available on Kaggle, the study meticulously evaluates key algorithms, including Support Vector Machine, Random Forest, and Long Short-Term Memory networks, in automating essay grading. Notably, both the Random Forest and Long Short-Term Memory (LSTM) algorithms stand out as front runners, demonstrating remarkable agreement with human graders. They both achieved an impressive Quadratic Weighted Kappa score of 0.97, echoing human-grade assessments, although there are distinctions in their MAE performance.

Furthermore, the investigation into grade normalization techniques uncovers their intricate interplay with model performance and human grading standards. These discoveries offer invaluable insights to educators and researchers, enabling them to refine automated grading systems for enhanced consistency and reliability. This study not only advances the field by shedding light on model selection and deep learning methodologies but also bridges the chasm between theory and real-world application in educational assessment, making it a pivotal contribution to the domain.

Contents

List of Figures

List of Tables

Acknowledgements

I would like to express my heartfelt gratitude to my supervisor, Dr. Alok Joshi, for his unwavering support and guidance throughout this journey. Your expertise and encouragement has been invaluable in shaping this work, and I appreciate your advice and continued support to submit this project as an academic paper beyond its initial scope.

I am deeply thankful to my parents, siblings, and the blessings of Almighty Allah for their love, prayers, and encouragement. Your belief in me has been my greatest motivation.

Lastly, I want to express my gratitude to myself for the determination and dedication invested in completing this project and educational journey. This work would not have been possible without the support of many others, and I am truly grateful for their contributions.

This project has been a journey of personal growth and achievement.

Chapter 1

Introduction

Essay grading and feedback is a tool used in the educational sector to evaluate and provide constructive criticism into the strengths and weaknesses of students' knowledge and writing skills (Adarkwah, 2021). There are various types of essays that students are required to write, including argumentative essays, descriptive essays, persuasive essays, and expository essays, among others (MasterClass, 2021). Each of these essays requires a different set of writing skills, and it is essential to evaluate and provide feedback on them accordingly. Natural Language Processing (NLP) is a sub-field under Artificial Intelligence which focuses on enabling computers to understand, interpret and generate human language (IBM, n.d.). NLP has been around for years and has been applied in various ways including sentiment analysis, chat-bots, virtual agents like Siri and Alexa, Machine translation like Google translate, etc (Jain, Kulkarni and Shah, 2018). NLP has been used to automate the process of essay grading and feedback by applying machine learning algorithms and NLP techniques to recognize patterns and features of the written text. These automated systems can save teachers time and effort while providing objective grading and personalized feedback to students thus improving their writing skills and knowledge.

1.1 Problem Statement

Automated essay grading system is a rapidly growing area that can potentially improve effectiveness and efficiency for teachers and educational institutions. The traditional method of essay grading and providing feedback to students can be time-consuming, subjective, and labor-intensive for teachers, especially when they have many students to grade. The process often requires the teachers to spend significant amount of resources, time, and effort to grade students' essays where teachers must read through each student's essay and provide meaningful feedback on the students' work which often could lead to a lack of consistency in the evaluation due to fatigue, the introduction of bias and limited feedback due to time constraints (Schinske and Tanner, 2014). Additionally, the limitations of human graders could hinder the learning process and development of the student's writing skills when provided with insufficient feedback.

Automated essay grading systems have been emerging by automatically grading essays and providing personalized feedback to students solving the limitations of human graders using various natural language processing (NLP) techniques and machine learning algorithms (Hearst, 2000). However, there are several automated grading systems currently existing, although they have limitations in terms of their ability to handle several context of essays such as academic essays, job application essays, scholarship essays, opinion pieces, creative writing essays, and among others, insufficient accuracy due to the algorithms and range of essay types used to train the model, limited customized feedback (Matthews et al., 2012), and lack of transparency of the grading process (Kumar and Boulanger, 2020). As such, there is a need for further research and development in this area to address these limitations and improve the effectiveness and reliability of automated essay grading systems.

1.2 Aim

This project aims to expand on previous work by developing an automated essay grading model using natural language processing (NLP) techniques, machine learning and deep learning algorithms that considers various low- and high-quality essays which accurately provides students grades independent of the prompts by adapting the model to the essays thus easing the efforts and workloads for teachers and providing prompt feedback to students.

1.3 Objectives

- To conduct a thorough literature review on existing automated essay grading systems, outlining their strengths and limitations.
- To collect and pre-process a large dataset of essays across different contexts and grade levels.
- To develop and evaluate an automated essay grading model to accurately grade different contexts of essays using a large variety of essay datasets, various NLP techniques, machine learning and deep learning algorithms.
- To evaluate the accuracy and effectiveness of the developed model by comparing the results with human graders and existing automated systems.

1.4 Significance and Impact

The development of a highly accurate and efficient automated essay grading model using NLP and machine learning can have various significance and impacts on teachers, students, and educational institutions. This project will provide more time for teachers to focus on other tasks such as lesson planning and student support, resulting in improved learning outcomes and academic performance for students. It will also provide access and faster feedback to students which can improve their performance, especially students with learning difficulties or disabilities. Additionally, educational institutions can potentially save costs by reducing the need for hiring additional grader/teaching assistants or reducing the amount of time required for grading. Also, this project can help institutions have standardized grading across multiple teachers, courses, or even institutions. Furthermore, the project will contribute to the development of state-of-the-art NLP techniques, machine learning and deep learning algorithms for automated essay grading systems, which can have broader applications in education and beyond. Specifically, the project can encourage further research in the area of automated grading for other specific contexts of assignments, such as mathematics, programming, and science, leading to continued advancements in the field. Additionally, the technology can be applied to the recruitment sector to evaluate job applications, the law sector to evaluate law briefs, and the medical sector to evaluate medical reports.

1.5 Overview of Dissertation

This dissertation is organized into distinct chapters, each contributing to a comprehensive exploration of automated essay grading (AEG) and its methodologies. The following chapters provide a structured journey through the research:

Chapter 2 - Literature, Technology and Data Survey provides a comprehensive review of automated essay grading (AEG), exploring NLP, machine learning, and deep learning. Critically analyzing existing studies, transparency issues, and examining the potential impact of AEG on student learning.

Chapter 3 - Methodology delves into the methodology of this project. Discussing the ASAP dataset, feature extraction methods, machine learning and deep learning models, grade normalization, experimental setup, and evaluation metrics used in this project.

Chapter 4 - Result and Discussion presents and analyzes the experimental results obtained from the experimentation of the three machine and deep learning models used, including algorithm performance and grade normalization implications. Also, alternative approaches and their potential outcomes were considered.

Chapter 5 - Conclusion and Future Work is the final chapter of this dissertation which summarizes key findings, discusses practical implications for educators and researchers, reflects on methodology strengths and limitations, and outlines future research directions.

Chapter 2

Literature, Technology and Data Survey

2.1 Overview

Automated Essay Grading (AEG) has become increasingly prominent in educational settings, harnessing the power of Natural Language Processing (NLP) techniques, Machine Learning (ML) and Deep Learning (DN) algorithms. These systems offer the potential to streamline the assessment process and provide timely feedback to students. However, achieving accurate and comprehensive essay evaluation presents numerous challenges, including the need for a deep understanding of language, context, and topic, as well as potential biases in grading. This literature review explores the evolution of automated essay grading systems, the algorithms and models used, contextual factors influencing essay assessment, datasets used and existing limitations and research gaps.

2.2 Literature, Technology and Data Review

Automated essay grading (AEG) systems have benefited from various natural language processing (NLP) techniques, including latent semantic analysis (LSA) for analyzing semantic content (Kaur and Sasi Kumar, 2019), Coh-Metrix for measuring readability and comprehensibility (McNamara et al., 2014), and automated essay scoring for assigning grades based on predefined rubrics (Hussein, Hassan and Nassef, 2019). However, accurately assessing essay quality poses significant challenges due to the need for a deep understanding of language, context, and topic. Developing effective algorithms for this task thus requires substantial amounts of data and expertise, while also carrying the potential for bias resulting from subjective perspectives of human creators, leading to inaccurate and unfair evaluations.

2.2.1 Historical Development of AEG Systems

The field of automated essay grading (AEG) has been in existence for several decades, with its origins dating back to the 1960s. The first AEG system, Project Essay Grade (PEG), was developed by Ellis Page, who utilized basic statistical methods and predefined essay features to assign grades to essays (Page, 2003). However, the limitations of these systems to evaluate complex essays necessitated the development of more advanced AEG systems that employ various grading techniques. Traditional approaches in the 1990s relied on pattern matching and statistical-based methods, whereas recent systems have employed artificial intelligence and natural language processing (NLP) techniques. PEG evaluates an essay's grammar, diction, and construction (Page, 2003), while a modified version of PEG by Shermis et al. (2001) focuses on grammar checking and has a strong correlation with human evaluators.

In comparison, Landauer and colleagues introduced Intelligent Essay Accessor (IEA), which employs Latent Semantic Analysis (LSA) to grade essays based on their content and language features, producing an overall grade of the essay (Landauer, Laham and Foltz, 2000). Powers et al. (2002) proposed E-rater, an AEG system utilized to grade essays on standardized tests such as GRE and TOEFL, along with Intellimetric by Rudner, Garcia and Welch (2006) and Bayesian Essay Test Scoring System (BESTY) by Rudner and Liang (2002), which leverage NLP techniques to assess an essay's style and content. The development of these systems and NLP in the 1990s and early 2000s allowed for the creation of more sophisticated AEG systems, capable of analyzing and grading more complex essays. Since then, considerable progress has been made in the field of AEG, with the advent of new algorithms and models utilizing advanced NLP techniques to analyze and grade essays.

2.2.2 Review of Existing NLP and Machine Learning Models in AEG Systems

Automated essay grading (AEG) is an amalgamation of natural language processing (NLP) techniques and machine learning (ML) algorithms. ML algorithms utilize statistical models to detect patterns in data. AEG research has used various Natural language processing, Machine learning and Deep learning models, including supervised, unsupervised, and neural networks, to grade essays and provide students with more comprehensive feedback. Popular algorithms used in AEG include Latent Semantic Analysis (LSA), which compares concepts in essays to those in a pre-scored corpus to determine grades, Landauer, Foltz and Laham (1998) have yielded inter-rater correlations that fall within the range of 0.64 to 0.84, as well as correlations of 0.59 to 0.89 between the LSA-based system and human graders, and Latent Dirichlet Allocation (LDA), which identifies the main topics in essays and evaluates them based on relevance (Blei, Ng and Jordan, 2003). Support Vector Machines (SVM) is another algorithm used to classify essays based on various criteria, such as grammar and coherence, with high levels of accuracy (Gandhi, 2018). Adamson, Lamb and December (2014) applied LSA to automatic essay grading system, their study assessed inter-rater agreement among human graders using kappa scores, which ranged from 0.629 to 0.819. However, the LSA model showed limited success in generating predictions that aligned with human graders, as indicated by a kappa score of 0.243 for the eighth essay set, suggesting only a slight level of non-random agreement with human scores. Li and Yan (2012) trained an SVM model on pre-graded essays achieving 86% precision. Ghanta (2019) compared three supervised ML models (linear regression, random forest, and SVM) on over 12,000 human-graded essays and found that the random forest model outperformed the others when compared with human scores as it achieved the lowest mean absolute error of 1.22 for the test dataset, compared to the 1.42 and 1.83 mean absolute error obtained by the linear regression and support vector regression model respectively. Meanwhile, unsupervised learning algorithms such as the voting algorithm implemented by Chen et al. (2010) have also shown success without the use of pre-graded essays and can be applied to any language with minimal modifications as it does not use any particular language features. They used two Chinese essay datasets to evaluate their model and compared it with supervised learning algorithms. On the first dataset the voting model achieved an agreement rate with the human graders by 94.5% outperforming the SVM (93.6%) and Bayesian (93.4%), on the second dataset the voting system achieved 92.7% also outperforming SVM (91.8%) and Bayesian (89.7%). However, the (Chen et al., 2010) model did not perform well on high-quality essays with creative ideas as the model used similes (similarity functions) to grade essays. Mahana and Apte (2012) employed a simple linear regression model using a dataset of approximately 13,000 essays consisting of eight different sets of essay contexts achieving an average kappa score of 0.73 and found that the model performed well only on persuasive essays (set 1) which are free from contextual text with a kappa score of 0.80 and poorly on other essays such as the narrative essays based on personal experience and imagination with a kappa score of 0.68 which contains lexical and complex sentence structure. Additionally, grammar, synonyms of keywords, and usage-specific features were not considered, and they suggested the use of advanced NLP features, such as N-grams, to improve the model's performance for contextual essays. Kim et al. (2004) developed an intelligent grading system using Probabilistic Latent Semantic Analysis (PLSA) to score descriptive examination papers automatically acquiring about 74% accuracy. However, the system relied on semantic similarity between a student's paper and a model paper to build linguistic semantic knowledge, which may not accurately estimate contextual semantic similarity.

2.2.3 Review of Existing Deep Learning Models in AEG Systems

More recently, deep learning algorithms such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) have been employed in AEG. CNN utilizes convolutional filters to extract features from essays and has demonstrated effectiveness in identifying sentence-level features such as transitions and topic sentences (Taghipour and Ng, 2016). Shehab, Elhoseny and Hassanien (2016) used a Neural Network, specifically Learning Vector Quantization (LVQ), to grade pre-scored essays from computer and information systems students with the results showing an agreement with the teachers grades in between 70 - 90% accuracy. While the model exhibited good performance, it required large pre-graded datasets for training. Liang et al. (2018) proposed a Siamese neural network architecture for automated essay scoring, taking automated essay scoring as a ranking problem by ranking the order of pair essays based on their quality. The study by Dasgupta et al. (2018) introduced an innovative RNN model utilizing the ASAP dataset, based on a qualitatively enhanced Long Short-Term Memory (LSTM) architecture, which harnessed word and sentence embedding in conjunction with linguistic and psychological features. Their model exhibited strong performance, achieving a Quadratic Weighted Kappa (QWK) score of 0.79. Zhu and Sun (2020) also used the LSTM with word embedding utilizing GloVe on the ASAP dataset achieving a 0.70 QWK score.

2.2.4 Limitations and Research Gaps in Existing AEG Systems

However, previous research of AEG has had limitations in their approaches. For instance, some studies relied solely on keyword matching to evaluate essays (*e.g.*, Patil and Patil, 2014), which fails to account for synonyms, word order, and lexical variability. Other studies focused on semantic meaning alone, without considering structural and syntactical style (Cutrone, Chang and Kinshuk, 2011; Song and Zhao, 2013). Devi and Mittal (2016) employed machine learning techniques with Ontology, including Latent Semantic Analysis, Generalized Latent Semantic Analysis, Maximum Entropy, and Bilingual Evaluation Understudy (BLEU), to apply subjective evaluation. However, their approach required an extended Ontology that covers the concepts of all computer science subjects which they found to provide more accurate results than the model without ontology. Rokade et al. (2018) proposed a semantic analysis and ontology-based approach to determine the knowledge orientation of essays, instead of relying on keyword matching, which may not be effective if students use their own knowledge or synonyms to describe ideas. These limitations highlight the need for more comprehensive and sophisticated methods for AEG that consider all aspects of writing, including content, structure, syntax, and semantics.

2.2.5 Review of Essays and Dataset Consideration for the Development of AEG Systems

Another important factor in NLP-based AEG and feedback systems is the context of the essays being evaluated (Hussein, Hassan and Nassef, 2019; Lexalytics, 2023). The type of essay, such as persuasive, expository, narrative, or argumentative, can significantly impact the system's performance, as each essay type has different characteristics and expectations from the reader. For instance, persuasive essays aim to convince the reader of a particular point of view, while expository essays provide information and explain a topic. Narrative essays tell a story, and argumentative essays present arguments and counterarguments on a topic (MasterClass, 2021). Additionally, the intended audience, purpose, and topic or prompt of the essay can also affect its context. For example, an essay intended for a scientific audience may require a different style and tone than one written for a general audience. The purpose of the essay may be to inform, persuade, or entertain, and the topic or prompt may require specific knowledge or expertise (Examples.com, Accessed 2023). Therefore, to ensure accurate and comprehensive evaluation and feedback, the system must be trained on a diverse set of essays that cover different topics, genres, and audiences. The size and composition of the dataset are also critical factors in achieving higher accuracy in automated essay grading (Pisarova, 2021). Larger and more diverse datasets that are representative of the target population are necessary to capture the variability in writing styles and topics. Several bench-marking datasets, such as the ASAP (Automated Student Assessment Prize) dataset sponsored by The William and Flora Hewlett Foundation (Ben Hamner, 2012), have been widely used by researchers to develop and evaluate automated essay scoring models. These datasets consist of eight different sets of essays written by high school students and graded by human graders based on various aspects of writing such as organization, clarity, and development of ideas and was used by Mahana and Apte (2012); Adamson, Lamb and December (2014). Other datasets have also been gathered from educational institutions or various available online databases.

2.2.6 Transparency and Impact of AEG Systems on Student Learning

Despite the potential of NLP-based automated essay grading systems to enhance the grading efficiency and consistency, there are several limitations that require attention. One of these limitations is the systems' inability to capture all writing aspects, including creativity and style, which are crucial for certain essay types (Ramesh and Sanampudi, 2022). These raises concerns that such systems may foster formulaic writing and lead to a lack of originality and creativity in students' writing (Turner, 2018). Furthermore, there are research gaps that necessitate further investigation. A significant research gap is the lack of transparency in AEG systems, with many systems utilizing opaque algorithms, making it difficult to comprehend their grading decisions. This lack of transparency could impede the progress and advancement of AEG systems. Another research gap is the dearth of studies examining the impact of AEG on student learning. While AEG systems offer a more efficient grading process, it remains uncertain whether they provide the same level of feedback as human graders or enhance student learning. More research is required to evaluate the potential benefits and drawbacks of using AEG systems in classrooms and their impact on student learning outcomes.

2.3 Summary

In conclusion, automated essay grading systems have undergone significant evolution, employing diverse algorithms encompassing statistical, mathematical, machine learning, deep learning, and natural language processing techniques. Despite their potential to streamline grading processes, these algorithms face challenges in accurately assessing essays and capturing the creativity of writers.

Recent developments in automated grading, particularly the utilization of Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN), have demonstrated their proficiency in extracting intricate features from essays, including sentence-level attributes like transitions and topic sentences. Importantly, these neural network architectures offer adaptability to various essay contexts, without the need for domain-specific knowledge.

Studies employing CNNs and RNNs in conjunction with comprehensive datasets like the ASAP (Automated Student Assessment Prize) dataset have shown promise in enhancing the accuracy and effectiveness of automated essay grading systems. The incorporation of these cutting-edge techniques not only extends the boundaries of automated grading but also provides practical insights for their application in educational settings.

Therefore, this project will focus on the utilization of advanced approaches such as Recurrent Neural Network (RNN), specifically Long Short-Term Memory (LSTM) models, alongside word embedding which has the potential to make substantial contributions to this evolving field. Additionally, experimenting with 2 machine learning algorithms (SVM and Random Forest) has shown promising results in the literature review. By leveraging these state-of-the-art methodologies and harnessing the rich ASAP dataset, this project represents a timely and significant step forward in advancing automated essay grading systems.

Chapter 3

Methodology

3.1 Overview

Building upon the valuable insights and limitations identified in the literature, technology and data review, this methodology section serves as the framework for achieving the following specific objectives in this project:

1. To systematically assess the efficacy of machine learning models in the context of automated essay scoring by evaluating their performance when applied to a diverse array of syntactic, lexical, and contextual features extracted from essays.
2. To conduct an in-depth evaluation of the performance of deep learning methodologies, specifically focusing on their utilization of word embedding, in the automated essay scoring domain.
3. To scrutinize the influence of grade normalization techniques on the performance of deep learning models employed for automated essay scoring, thereby interpreting the significance of grade normalization in enhancing or diminishing model effectiveness.

The potential of automated essay grading within the educational sector hinges on a systematic approach and robust methodology to be employed, which ensures the accuracy, reliability and practicality of the predictive models developed. This project aims to harness the potential of automated essay grading, offering enhanced efficiency and consistency in the grading process for the benefit of educators and students.

This project was carried out using the Python programming language in a Jupyter Notebook environment. Python is the most used language for machine learning and deep learning projects due to the availability of existing libraries.

This chapter outlines the methodology, which comprises different stages, as illustrated in Figure 3.1, with each stage contributing to the overall efficacy of the predictive model. The methodology framework began with the collection of the ASAP dataset from Kaggle (Ben Hamner, 2012) and subsequent preprocessing, establishing a foundational dataset for analysis and modeling. Following this, meaningful features were extracted using NLP tool kit (NLTK), Count Vectorizer and Word2Vec, encapsulating the linguistic, structural and contextual attributes of essays, ensuring a holistic assessment. At the core of the methodology is the definition and training of three distinct models and regression algorithms – Support

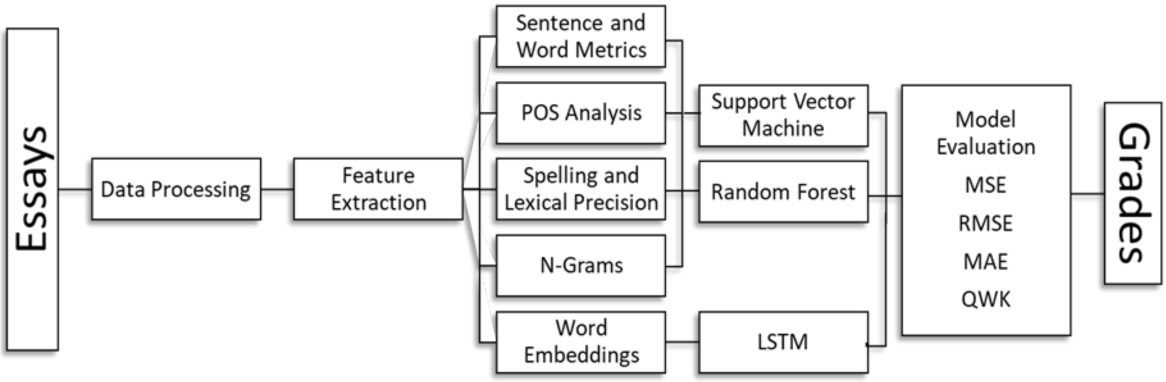


Figure 3.1: Automated Essay Grading Framework

Vector Machines (SVM), Random Forest (RF) and Long Short-Term Memory (LSTM). These models were selected based on their promising results from the literature review. Rigorous evaluations were conducted utilizing multiple metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Quadratic Weighted Kappa (QWK), offering a comprehensive assessment of the model's performance. Ultimately, the methodology ends in the prediction of grades thus the use of regression algorithms, aligning the project with the broader aim of advancing automated essay grading systems.

3.2 Data Collection: ASAP Dataset

The foundation of this project relies on the Automated Student Assessment Prize (ASAP) dataset which is underpinned by its exceptional relevance to the objectives of the project. This dataset was made available by the Hewlett Foundation and accessed through Kaggle (Ben Hamner, 2012). Comprising 12,978 essays grouped into eight different sets, encompassing two essay prompt types. The first type includes prompts for Persuasive, Narrative, and Expository essays (PNR), where students were asked to share their opinions about specific topics. The second type involves Source Dependent Responses (SDR), where students read short passages and answered questions based on their comprehension. This diversified dataset reflects on the real-worlds educational scenarios, wherein students respond to varied prompts that demand differing levels of critical thinking, linguistic complexity and coherence, thus facilitating a comprehensive analysis of automated essay scoring across diverse essay genres, mirroring the multifaceted nature of academic writing.

The essays were graded by two human graders, who followed specific guidelines for each set of prompts. The scores given by these graders were combined to give a final score, which had different ranges depending on the set of prompts. The students who wrote these essays were in grades 7 to 10, and each essay contained between 150 to 650 words. The dual-grading system where essays were independently evaluated by two human graders provides the opportunity to benchmark and validate the performance of predictive models. As such, the dataset serves as a cornerstone in realizing the projects objectives of developing reliable automated essay grading models that can be practically integrated into the educational settings. The description of each essay set is detailed in Appendix C.3.1 and a more elaborate description can be found on the Kaggle Page `https://www.kaggle.com/competitions/asap-aes/data`.

This dataset offers a rich variety of essay types, score ranges, and grade levels, making it a valuable resource for automated essay grading research and would be utilizing the resolved scores for each essay set. For an illustrative overview of the dataset's attributes, refer to Table 3.1 which breaks down the different parts of each essay set.

Set	No of Essays	Essay Type	Score Range	Grade Levels	Avg word length	Attributes
1	1783	PNR	2 – 12	8	350	5
2	1800	PNR	1 – 6	10	350	8
3	1726	SDR	0 – 3	10	150	5
4	1770	SDR	0 – 3	10	150	5
5	1805	SDR	0 – 4	8	150	5
6	1800	SDR	0 – 4	10	150	5
7	1569	PNR	0 – 30	7	250	13
8	723	PNR	0 – 60	10	650	17

Table 3.1: Description of ASAP Dataset Essay sets

3.3 Data Processing

Data cleaning and preprocessing play a pivotal role in guaranteeing the quality, reliability, and validity of the analysis conducted in this project. The process of transforming raw data into a structured and refined format is instrumental in mitigating potential biases, enhancing the accuracy of model training, and ensuring a robust foundation for subsequent analysis. The automated student assessment prize (ASAP) dataset, a valuable resource for this project, had already undergone some initial preprocessing steps, including named entity recognition (NER) to anonymize the dataset to mitigate ethical issues. The data processing and tokenization steps began by extracting the necessary attributes by removing attributes with missing values to ensure data consistency, as Table 3.1 illustrated that different essay sets had different number of attributes which added complexity to the project.

Before embarking on the feature extraction phase, a comprehensive essay cleaning process was performed. This preparatory step involved the removal of stopwords and specific characters and punctuation that held limited analytical significance. For instance, words beginning with "@"—a characteristic of placeholders used during NER was removed. Stopwords, being commonly used words such as "and" "the," "is," etc., offer minimal substantive content and can introduce noise into the analysis (Naveen, 2023). Their removal not only streamlines the dataset but also aids in capturing more meaningful patterns in the essays. The technique for the stopword removal utilized pre-existing NLTK English corpus and the removal of special characters and punctuation was done using regular expressions. Following the cleaning process, essay tokenization using the NLTK library was performed, meticulously breaking down the essays into individual sentences and words. These preprocessed and tokenized essays were then integrated into a structured data frame, meticulously primed for subsequent analysis and model training, thereby enhancing analytical efficiency. In Appendix E.1, the Python code for preprocessing and tokenization is provided.

Lastly, it is worth noting that the grades assigned by human graders exhibited varying ranges across different essay sets, as demonstrated in Table 3.1. To ensure a level playing field for comparison and analysis, these scores were normalized to a unified scale spanning from 0 to 10. This normalization facilitated was implemented to evaluate the models against normalized or unnormalized data.

3.4 Feature Extraction

In this phase, the objective is to extract quantifiable attributes from the essays that can be utilized for predictive model development. Various NLP metrics and techniques were employed to capture different and distinctive aspects of the essays to provide a comprehensive analysis of the essay's contents, structure, and linguistic properties (Ramesh and Sanampudi, 2022). The features that were extracted were guided to mimic the grading criteria such as the style, organization, content, coherence and language conventions from the ASAP dataset that were the benchmark for grading the essays by the human graders.

3.4.1 Structural and Lexical Characteristics

The metrics used include sentence count, word count, unique word count, character count, and average word length which provides insights into the structural and lexical aspects of the essays. Sentence count and word count offer an understanding of essay length and structural organization, reflecting the completeness and coherence of the essay's arguments. Unique word count and average word length provide indications of lexical richness, vocabulary diversity, and level of elaboration. Character count acts as an alternative measure of essay length, further enhancing the depth of structural analysis (Verma and Srinivasan, 2019). In Appendix E.2, the Python code for extracting structural and lexical features from the essays are provided.

3.4.2 Syntactic and Linguistic Characteristics

Part-of-speech analysis is conducted to discern the distribution of grammatical categories within the essays. The occurrence of different parts of speech, including nouns, verbs, adjectives, adverbs, pronouns, prepositions, conjunctions, and determiners, is quantified. This analysis contributes to understanding the syntactic composition and linguistic characteristics of the essays (Verma and Srinivasan, 2019). In Appendix E.3, the Python code for extracting syntactic and linguistic features from the essays are provided.

3.4.3 Spelling Precision Characteristics

Two aspects of lexical precision are examined, Spelling errors and the misspelled words. The identification and quantification of spelling errors contribute to the assessment of the essays' linguistic accuracy and attention to detail. The presence of spelling errors can indicate the writers' proficiency in language usage, their adherence to conventions, and the level of editing and proofreading, thus providing a fine-grained perspective on areas of lexical imprecision (Verma and Srinivasan, 2019). In Appendix E.4, the Python code for extracting spelling characteristics features from the essays are provided.

3.4.4 N-gram Features

N-gram features (Bag-of-Words) constitute sequences of N items, typically words, encompassing bi-grams (two-word sequences) and tri-grams (three-word sequences). Their primary mission is to capture the intricate web of contextual associations deeply embedded within the essays, a facet of writing that significantly impacts coherency and persuasiveness (Poudel, 2018). Contextual association within an essay takes on paramount importance. It pertains to the way ideas, evidence, and arguments, and the seamless interconnection ensures coherence, logical flow, and thematic consistency (Lexalytics, 2023). The power of an essay hinges on its ability to orchestrate these components harmoniously, ultimately reinforcing its overall message and persuasive power.

To leverage the potential of N-gram features for the automated essay grading model, the Count Vectorizer technique was utilized. This method generated a spectrum of features based on N-gram occurrences, ranging from 1-gram to 3-grams, inspired by the suggestions from (Mahana and Apte, 2012). For instance, tri-grams, which encompass sequences of three words, have the capacity to unveil intricate linguistic patterns concealed within the essays. These N-gram features illuminate the local context of essays, offering insights into how words and ideas interrelate in proximity. This goes beyond the conventional analysis of essays and enables the model to capture the subtle yet profound connections between words and concepts.

For instance, consider the tri-gram "scientific research methods" This sequence encapsulates a specific contextual association within an essay, shedding light on the depth of understanding conveyed by the student. By quantifying these features across essays, the model advances its comprehension of how students structure arguments, convey ideas, and establish relationships between concepts. In Appendix E.5, the Python code for extracting n-gram features from the essays are provided.

3.4.5 Word Embedding: Semantic Mapping

Word embedding offers a profound theoretical foundation by mapping words to high-dimensional numerical vectors, where each dimension encodes specific semantic or contextual information (Dutta, 2022). In this project, the Skip-gram architecture of Word2Vec was employed to calculate average feature vectors for essays by extracting 300 features. These feature vectors serve to encapsulate the semantic essence of each essay.

Skip-gram is a Word2Vec model that excels in predicting contextual words based on a target word. Through a rigorous mathematical framework involving techniques like stochastic gradient descent and negative sampling, Skip-gram optimizes word vectors in the vector space, ensuring that words with similar meanings or contexts cluster together (Kulshrestha, 2020). This optimization process is foundational to the creation of meaningful word embedding.

In the context of this project, these Skip-gram-based word embeddings are instrumental in quantifying linguistic, structural, and contextual attributes across essays. Words within essays are similarly mapped into a vector space, enabling the model to understand the nuanced relationships between words in the grading context. For instance, consider the word “persuasive”, through Word2Vec, it becomes part of a vector space where words like “convincing,” “argumentative,” and “compelling” are close neighbors, signifying their semantic similarity. This allows the model to not only analyze the essays based on traditional linguistic attributes but also grasp the deeper semantic nuances with the text. By quantifying these features across essays, the model obtains multidimensional representations of linguistic, structural and contextual attributes. The resultant multidimensional representations lay a robust foundation for subsequent model development and sophisticated analysis, ultimately contributing to the advancement of automated essay grading systems. In Appendix E.6, the Python code for extracting word embedding features from the essays are provided.

3.5 Algorithms

In this section, the detailed algorithms utilized for predicting essay scores based on the extracted features are discussed. Each algorithm contributes a distinct approach to the project, offering insights into their respective strengths and limitations.

3.5.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) emerges as a robust and versatile algorithm for classification tasks, notably in evaluating aspects like grammar and coherence, all while maintaining a strong track record of accuracy (Gandhi, 2018). SVM is a supervised learning algorithm with the primary objective to identify an optimal hyperplane that effectively separates data points while maximizing the margin between different classes. For this project, SVM is used for a regression task which minimizes margin violations while constructing a predictive function that approximates the relationship between features and target values. SVM is robust against overfitting and captures non-linear relationships, but its complexity increases with the number of data points (Sethi, 2023).

SVM revolves around the optimization of a cost function that meticulously balances the trade-off between margin maximization and margin violation minimization. In Support Vector Regression (SVR), the model embarks on a quest to find a hyperplane that minimizes the epsilon-insensitive loss function. Simultaneously, it integrates a regularization term (C) into its calculations, wielding this term as a tool to govern and mitigate the risks of overfitting. Within this framework, the loss function introduces a margin of error (epsilon), granting the model the flexibility to ignore data points that fall within this margin during training. The goal is to equip the SVR model with the ability to construct a predictive function that accurately forecasts target values—in this case, the grading of essays.

SVR was chosen as one of the initial candidates to predict grades using the extracted features from the essays. The dataset used was split using the conventional 70-30 split to ensure a robust evaluation. The optimal parameter values used are:

Parameter	Value
Regularization term	1.0
Epsilon	0.2

Table 3.2: SVM Parameter Values

It is vital to underscore that the decision to employ SVM in this project is underpinned by a thorough exploration of relevant literature. Previous studies by Gandhi (2018); Li and Yan (2012); Ghanta (2019) have attested to SVMs proficiency in essay grading, frequently achieving exceptional levels of concurrence with human graders. Furthermore, SVM's unique ability to decipher intricate relationships between features and target values renders it eminently suitable for the task at hand.

3.5.2 Random Forest

The Random Forest algorithm constructs multiple decision trees during training and aggregates their predictions to improve generalization and robustness. In the context of regression, the algorithm predicts scores by averaging the predictions of individual decision trees. Random forest is also robust against overfitting and handles non-linear relationships but can be computationally expensive for large datasets. Random Forest can effectively handle the various linguistic features extracted from essays. It can capture intricate relationships between different features and their contribution to predicting essay grades (Beheshti, 2022). Like the SVR model, the dataset was split into training and testing sets. The Random Forest model used in this project consists of an ensemble of one hundred decision trees and trained on the extracted features. This ensemble approach, marked by the diversity of these trees, proved highly effective in capturing the intricate relationships among various features and their collective contribution to predicting essay grades. After conducting various experiments with different parameter combinations, it was discovered that the model achieved the best performance with the following set of optimal parameter values listed below:

Parameter	Value
N estimator	100
Random State	42

Table 3.3: Random Forest Parameter Values

The decision to employ Random Forest is based on its performance in the different literature's. Ghanta (2019) demonstrated that Random Forest outperformed other supervised machine learning models, including linear regression and SVM, in the context of essay grading. Notably, it achieved the lowest mean absolute error, a testament to its prowess in predicting essay scores, as validated against human graders.

3.5.3 Long-Short Term Memory (LSTM)

Long Short-Term Memory (LSTM), a specialized recurrent neural network, excels in capturing temporal dependencies within sequential data. LSTM theoretical framework is grounded in the concept of sequential data processing. Unlike traditional feedforward neural networks, LSTM is equipped to handle data sequences by incorporating memory cells and gating mechanisms. It effectively addresses the vanishing gradient problem, which often plagues standard RNNs. In essence, LSTM can retain and propagate information over long sequences, enabling it to capture temporal dependencies — a critical requirement when assessing essays with coherent narratives.

In this project, LSTM processes essays as sequences, leveraging Word2Vec embedding. Three variations of LSTM architectures were considered: a 2-layer LSTM, a 3-layer LSTM, and a 4-layer LSTM. Each LSTM model was composed of recurrent neural network (LSTM) layers followed by a dropout layer to mitigate overfitting and a dense output layer for prediction. LSTM accommodates intricate relationships and data patterns based on its core that lies in its gated mechanism, where it selectively updates, reads and writes to the memory cell. However, its efficacy is contingent on careful hyperparameter tuning, and its computational intensity demands substantial resources (Zhao et al., 2017).

In the context of essay grading, LSTM can understand the sequential structure of essays and capture the nuanced linguistic patterns that might not be captured by traditional algorithms. It leverages Word2Vec embeddings to capture semantic relationships among words. The use of a recurrent neural network was inspired from the literature by Dasgupta et al. (2018) and Zhu and Sun (2020) where they utilized Convolutional neural networks such as LSTM with sentence embedding, linguistic features and word embedding using GloVe. This inspiration led to the exploration of LSTMs potential in understanding essays at a deeper level, considering their sequential composition. After conducting various experiments with different parameter combinations, the model achieved the best performance with optimal parameter values listed below:

Parameter	Value
Input features	300
Activation	Relu
Optimizer	rmsprop
Loss function	mse

Table 3.4: LSTM Parameter Values

3.6 Evaluation Metrics

This section defines and discusses the evaluation metrics employed to assess the performance of the automated essay scoring models. These metrics provided quantitative insights into the accuracy and effectiveness of the models' predictions.

- Mean Squared Error (MSE): MSE measures the average of the squared differences between the predicted and actual values. In the context of automated essay grading, MSE serves as a crucial metric to quantify the average squared deviation between the grades predicted by the automated essay scoring models and the actual grades assigned by human graders. A low MSE indicates that the predicted grades are closely aligned with the true grades(Frost, 2023a). MSE is calculated as shown in Equation 3.1:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.1}$$

  Where n is the number of samples, y_i represents the actual human grade, and $\hat{y}_i$ is the predicted grade for the ith sample.

- Root Mean Squared Error (RMSE): RMSE is the square root of the MSE and provides a measure of the average magnitude of the error between predicted and actual grades. It is particularly useful for understanding the extent of errors in the same unit as the original data. Like MSE, a lower RMSE signifies a better model fit and predictive accuracy (Frost, 2023b). RMSE is calculated as shown in Equation 3.2:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{3.2}$$

- Mean Absolute Error (MAE): MAE calculates the average absolute differences between the predicted and actual grades. It offers a more interpretable measure of prediction error since it is not influenced by the square term as in MSE. MAE is less sensitive to outliers, making it a suitable metric when outliers could impact the analysis. A lower

MAE indicates a closer alignment of predicted and actual grades (Frost, 2023a). MAE is calculated as shown in Equation 3.3:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{3.3}$$

- Quadratic Weighted Kappa (QWK): QWK assesses the agreement between predicted and actual grades while considering the agreement that could occur by chance. It ranges between -1 and 1, with higher values indicating better agreement. QWK considers both the actual and predicted grades' distributions. The use of the Quadratic Weighted Kappa score was the recommended evaluation metric by the Hewlett foundation and has been utilized by researchers in the field of automated essay grading. The formula for QWK involves observed and expected agreement matrices and is often calculated using a weighted Cohen's Kappa formula. A higher QWK score demonstrates the model's ability to capture nuanced distinctions in grades (Frost, 2022). QWK is calculated as shown in Equation 3.4:

$$\text{QWK} = \frac{\sum_{i=1}^{k}\sum_{j=1}^{k} w_{ij} \cdot O_{ij}}{\sum_{i=1}^{k}\sum_{j=1}^{k} w_{ij} \cdot E_{ij}} \tag{3.4}$$

  Where: k is the number of categories
  O_{ij} is the observed agreement between rater i and rater j
  E_{ij} is the expected agreement between rater i and rater j by chance
  w_{ij} is the weight associated with the pair (i, j), i.e., the difference between the grades.

  As shown in Equation 3.5, the weights matrix can be calculated using:

$$w_{ij} = \frac{(i-j)^2}{(N-1)^2} \tag{3.5}$$

These evaluation metrics collectively provide a comprehensive understanding of the models' predictive performance. MSE, RMSE, and MAE offer insights into the magnitude of the prediction errors and how consistently the model predicts grades across different essays, while QWK captures the agreement beyond random chance, essential in contexts where grades' distribution is diverse. The choice of these metrics ensures a robust evaluation framework that considers different aspects of model performance and suitability for automated essay scoring applications (Shehab, Elhoseny and Hassanien, 2016).

3.7 Summary

The methodology chapter established a structured framework to unlock the potential of automated essay grading. The aim is to create accurate, reliable models that streamline the grading process for educators and students. Employing Python, along with libraries like NLTK, Scikit-Learn, and Keras/Tensorflow, to work with the Automated Student Assessment Prize (ASAP) dataset — a diverse collection of essays graded by human evaluators.

The methodology comprises data preprocessing, feature extraction, algorithm selection (Support Vector Machines, Random Forest, and Long Short-Term Memory), and rigorous model evaluation using metrics such as Mean Squared Error, Root Mean Squared Error, Mean Absolute Error, and Quadratic Weighted Kappa. This robust framework provides a foundation for assessing the models' performance in the Results and Evaluation section, ultimately contributing to the advancement of automated essay grading systems.

Chapter 4

Result and Discussion

4.1 Overview

This chapter is the bridge between theory and practice, connecting the methodological foundation from Chapter 3 with the practical outcomes of the model evaluations. It systematically assesses the predictive accuracy and effectiveness of models in automated essay scoring, leveraging evaluation metrics. Exploring various models, such as Support Vector Machine (SVM), Random Forest, and Long Short-Term Memory (LSTM), scrutinizing their unique insights into essay grading. This exploration not only guides model selection but also deepens the understanding of linguistic features' role in written communication assessment. Employing K-Fold cross-validation ensures robust, unbiased assessments of model performance across diverse scenarios. Throughout this chapter, the methodology was seamlessly linked with practical results, systematically evaluating each project objective's contributions. The results and evaluation began with a comprehensive model comparison based on the training and testing of the models, harnessing the strengths of SVM, Random Forest, and LSTM models. Subsequent sections focus on specific project objectives, examining their outcomes and implications. Ultimately, the goal is to cultivate a deeper understanding of how these models perform in automated essay grading and their real-world applicability.

4.2 Model Comparison

In this section, a comparative analysis of the three selected models utilized in this project is evaluated. Firstly, the SVM and random forest models utilized the same input data which were the features extracted from the essays encompassing a range of linguistic aspects including sentence and word metrics, POS analysis, spelling accuracy, lexical precision and N-grams. These features were utilized to gain insights into the influence of various linguistic and structural characteristics of the essays, thus adding interpretability to the models prediction.

Both the SVM and Random Forest model were trained using all 8 essay sets in the dataset which was split using the train and test split function in Scikit learn. The dataset was divided into training and testing subsets with 70% of the data for training and 30% for testing. The Models were trained on the training data using the ".fit()" method with their hyperparamters passed and the grades of the essays were predicted using the testing data and the ".predict" method. In Listing 4.1, the training and testing code snippet of an SVM model is outlined.

Listing 4.1: Training an SVM model

```
svm = SVR(C=1.0, epsilon=0.2)
svm.fit(train_X, train_y)
pred_y = svm.predict(test_X)
```

Similarly, the training and testing of Random Forest is similar to that of SVM which can be seen by the example code snippet in Listing 4.2.

Listing 4.2: Training a Random Forest model

```
rf = RandomForestRegressor(n_estimators = 100, random_state = 42)
rf.fit(train_X, train_y)
pred_y = rf.predict(test_X)
```

In the LSTM model, the input data diverges notably as it comprises of word embedding generated through Word2Vec, a technique adept at capturing semantic relationships within essays. The choice of word embedding, and LSTM architecture aligns with the objective of exploring deep learning methodologies for essay grading. Word embedding allow the model to discern complex semantic relationships within essays, while LSTMs sequential learning capability enables it to capture intricate contextual patterns. The LSTM model is characterized by specific hyperparameters outlined in Table 3.4, including a 2-layer architecture and a recurrent dropout rate of 0.4, facilitating effective learning and generalization. The datasets used for the LSTM model are also split into 70% training and 30% testing subsets using the Scikit learn library. Additionally, the datasets also undergoes partitioning into training and testing subsets using K-Fold cross validation with 5 folds, ensuring a robust and unbiased assessment of model performance across diverse essay sets and scenarios (Hundley, 2022). The Long Short-Term Memory (LSTM) model, designed as a sequential neural network, is trained on the training data for each fold, allowing it to learn complex sequential patterns intrinsic to essays. Subsequently, this trained model is employed to predict essay grades on the respective testing data for each fold. In Listing 4.3, the training and testing code snippet of a LSTM model is outlined.

Listing 4.3: Training a LSTM model

```
lstm_model = get_model()
lstm_model.fit(training_vectors, y_train, batch_size=64, epochs=50)
y_pred = lstm_model.predict(testing_vectors)
```

This comprehensive model comparison highlights the diversity in input data, training approaches, and hyperparameters among the SVM, Random Forest, and LSTM models. Each model is evaluated using the same metrics which are MSE, MAE, RMSE and QWK, offering valuable insights into their performance in the task of automated essay grading.

4.3 Contribution 1: Evaluation of Machine Learning Models on Linguistic and Structural Features of Essays

The aim of Objective 1 was to systematically assess the performance of machine learning models in the context of automated essay scoring, with a focus on their effectiveness when applied to a diverse set of syntactic, lexical, and contextual features extracted from essays.

4.3.1 Results and Analysis

The evaluation results, summarized in Table 4.1, shed light on the predictive performance of these models across multiple evaluation metrics, including Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and Cohen's Quadratic Kappa (QWK) score.

Model	MSE	RMSE	MAE	QWK
SVM	53.82	7.34	3.81	0.54
Random Forest	4.76	2.18	1.06	0.97

Table 4.1: Performance Metrics for SVM and Random Forest Models

The results of this evaluation revealed significant distinctions in the performance of two prominent models, the Support Vector Machine (SVM) and Random Forest, when applied to the extensive feature set extracted from essays. Notably, the Random Forest model demonstrated superior performance across all evaluation metrics, achieving a Mean Absolute Error (MAE) of 1.06 and a Quadratic Weighted Kappa (QWK) score of 0.97. In contrast, the SVM model exhibited comparatively lower performance, with an MAE of 3.81 and a QWK score of 0.54 as shown in Figure 4.1. In Appendix A, the SVM and Random Forest-based approach to predicting essay grades is illustrated using a sample essay as a case study.

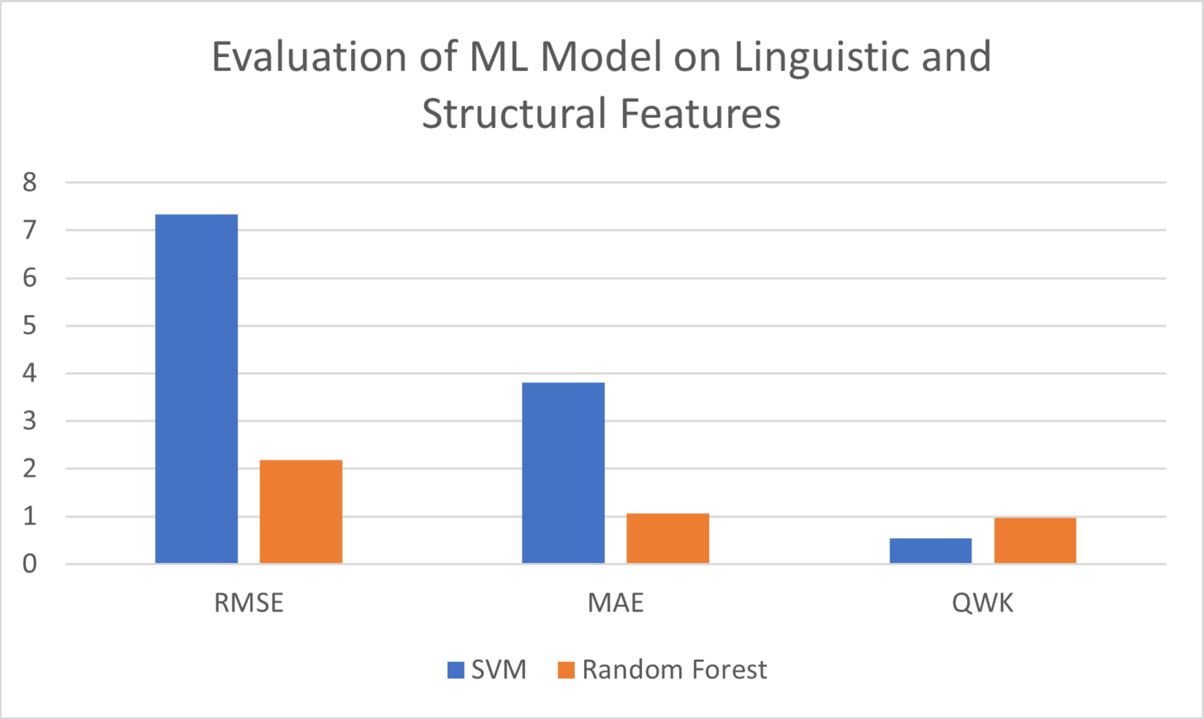


Figure 4.1: Performance of SVM and Random Forest Models

These findings align with previous research by Ghanta (2019), who also explored the performance of Random Forest and SVM models in automated essay scoring. In their study, Ghanta (2019) reported an MAE of 1.22 for Random Forest and 1.83 for SVM, emphasizing the efficacy of ensemble learning methods like Random Forest in handling complex feature sets and enhancing predictive accuracy.

4.3.2 Contributions to the Field

This study makes a notable contribution to the field of automated essay scoring by providing empirical evidence of the superiority of Random Forest over SVM in handling diverse linguistic and structural features. The results underscore the importance of model selection when confronted with complex feature sets extracted from essays. Moreover, the findings reinforce the existing literature on the effectiveness of ensemble learning methods and their potential to enhance predictive accuracy in automated essay scoring scenarios.

4.4 Contribution 2: Evaluation of Deep Learning Models on Contextual Features of Essays

Objective 2 of this study aimed to perform an extensive evaluation of deep learning methodologies, with a specific focus on their utilization of word embedding, within the context of automated essay scoring.

4.4.1 Results and Analysis

The evaluation of objective 2, employed Long Short-Term Memory (LSTM) models with varying numbers of layers (2-layer, 3-layer, and 4-layer), all trained using Word2Vec embedding. The performance of these models was meticulously assessed, yielding the following outcomes in Table 4.2:

Model	MSE	RMSE	MAE	QWK
LSTM 2 - Layer	4.86	2.20	1.12	0.97
LSTM 3 - Layer	4.99	2.23	1.12	0.97
LSTM 4 - Layer	5.27	2.29	1.14	0.97

Table 4.2: Performance of LSTM Models with Word2Vec Embeddings

These outcomes offer invaluable insights into the effectiveness of deep learning models, specifically Long Short-Term Memory (LSTM) networks, in the domain of automated essay scoring. Remarkably, all three LSTM architectures demonstrated consistent performance levels as shown in Figure 4.2, yielding identical Quadratic Weighted Kappa (QWK) scores of 0.97.

This performance surpasses the results reported in existing literature. For instance, the model outperforms the study conducted by Zhu and Sun (2020), who achieved a QWK score of 0.70 while utilizing GloVe word embeddings in conjunction with an LSTM model. Similarly, the model surpasses the findings of Dasgupta et al. (2018), who employed an LSTM model incorporating sentence embeddings and linguistic features, attaining a QWK score of 0.79. In Appendix A, the LSTM-based approach to predicting essay grades is illustrated using a sample essay as a case study. These results underscore the robustness and efficacy of the approach in automated essay scoring, marking a notable advancement in the field.

4.4.2 Contributions to the Field

The evaluation of deep learning models, specifically Long Short-Term Memory (LSTM) networks, utilizing Word2Vec embeddings in the context of automated essay scoring yields significant

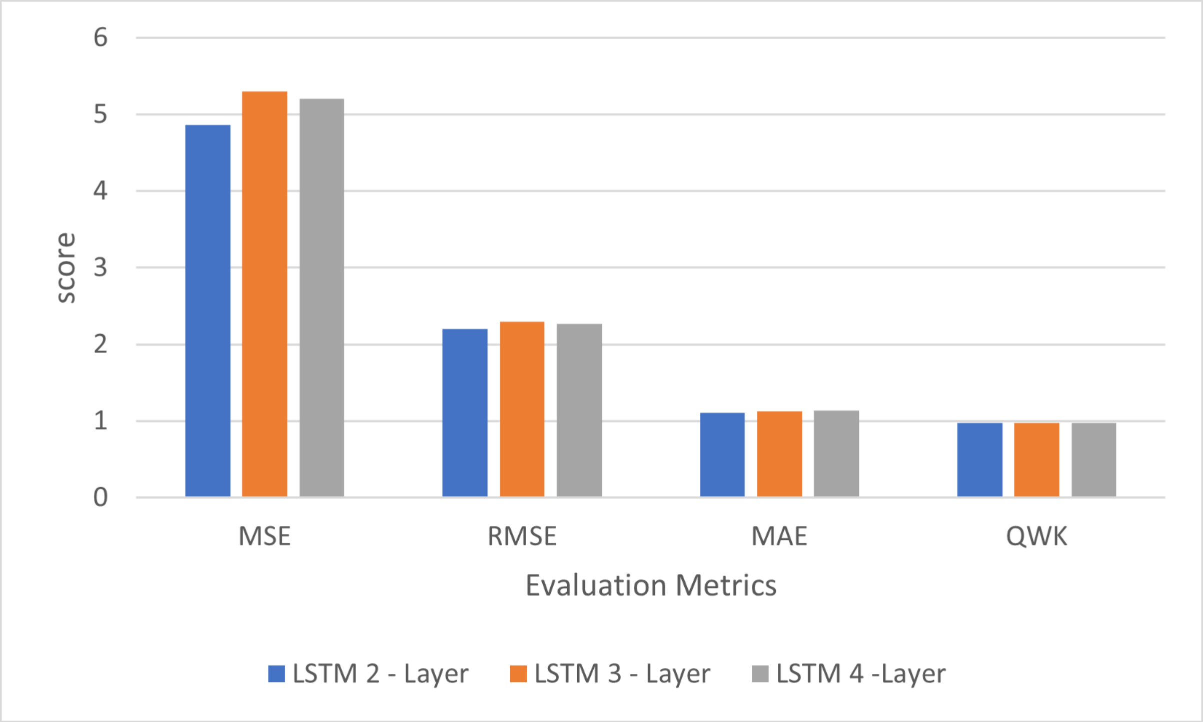


Figure 4.2: Performance of LSTM Models with Word2Vec Embeddings

contributions to the field. Notably, all LSTM architectures consistently achieved a Quadratic Weighted Kappa (QWK) score of 0.97, demonstrating their robustness and reliability in this domain. These results surpass previous research, outperforming studies that utilized alternative word embeddings or linguistic features. This performance validation of Word2Vec embeddings expands the toolkit available to researchers and educators in automated essay scoring. In summary, this study advances the understanding of deep learning methodologies in automated essay scoring, paving the way for improved performance standards and offering practical guidance for future research and implementation in educational settings.

4.5 Contribution 3: Exploring the Effect of Grade Normalization on Performance of AEG Models

To scrutinize the influence of grade normalization techniques on the performance of deep learning models employed for automated essay scoring, thereby interpreting the significance of grade normalization in enhancing or diminishing model effectiveness.

4.5.1 Results and Analysis

To investigate the impact of grade normalization techniques on the performance of deep learning models in automated essay scoring, the Long Short-Term Memory (LSTM) model was trained using Word2Vec embeddings on two variations of the training set: one with unnormalized human-graded scores and another with grade normalization. The comparative outcomes are outlined below in Table 4.3:

The results elucidate the significance of grade normalization techniques in the context of

Model	MSE	RMSE	MAE	QWK
LSTM 2 - Layer (Unnormalized Grades)	4.86	2.20	1.12	0.97
LSTM 2 - Layer (Normalized Grades)	3.25	1.80	1.35	0.68

Table 4.3: Performance Comparison of LSTM Models With and Without Grade Normalization

automated essay scoring. Notably, when grade normalization is employed, the LSTM model demonstrates a substantial reduction in Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), indicating improved predictive accuracy see Figure 4.3. However, the Quadratic Weighted Kappa (QWK) score experiences a decrease, reflecting a trade-off between accuracy and agreement with human graders. In appendix A, the LSTM-based approach to predicting essay grades is illustrated using a sample essay as a case study. These findings underscore the importance of grade normalization as a technique for enhancing the effectiveness of deep learning models in automated essay scoring, with implications for grading consistency and overall model performance.

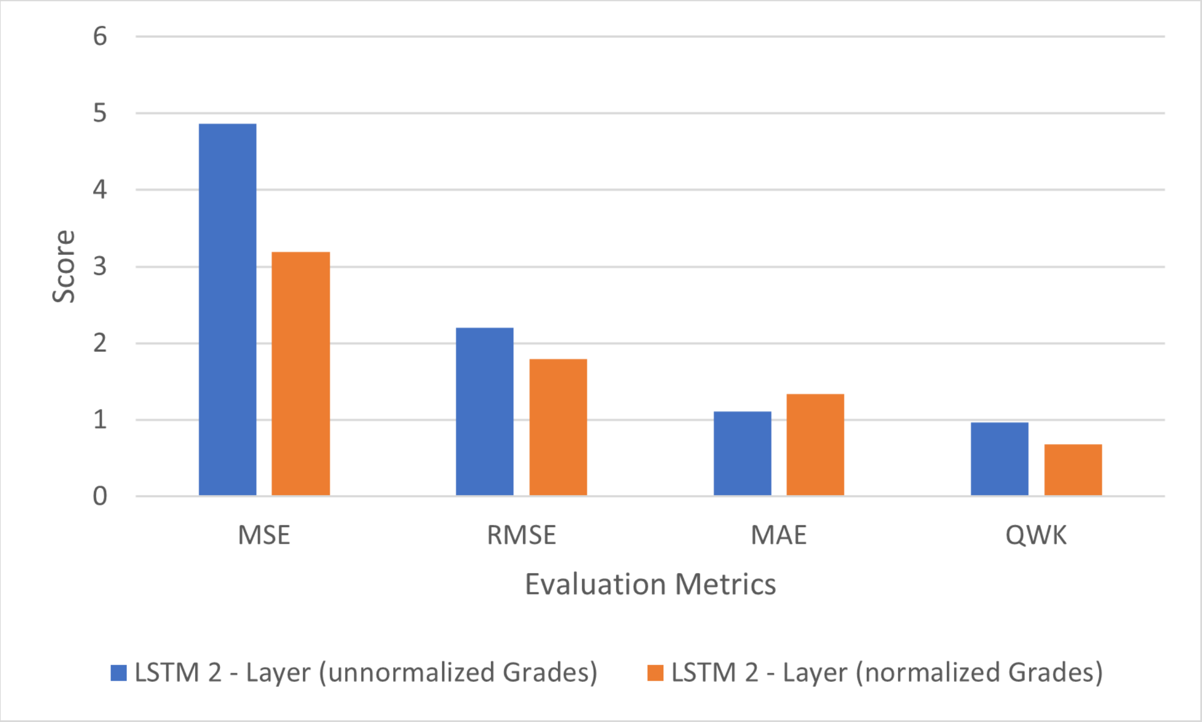


Figure 4.3: Performance Comparison of LSTM Models With and Without Grade Normalization

4.5.2 Contributions to the Field

This analysis contributes to the existing literature by shedding light on the nuanced relationship between grade normalization and automated essay scoring models. It highlights the potential trade-offs between predictive accuracy and agreement with human graders, offering valuable insights for researchers and educators seeking to optimize the performance of automated grading systems. Furthermore, this study underscores the importance of considering grade normalization as a key component in the development and implementation of automated essay scoring models, fostering a more comprehensive understanding of the grading process and its impact on model outcomes.

4.6 Discussion

This section delves into a comprehensive discussion of the methodology, the obtained results, alternative approaches that could have been employed, and the limitations of the models along with potential strategies to mitigate these limitations. The methodology for automated essay grading (AES) encompassed a multifaceted approach, combining natural language processing (NLP), machine learning, and deep learning techniques. Leveraging a diverse feature set extracted from the ASAP dataset to train and evaluate the models. The inclusion of various linguistic and structural features, including sentence and word metrics, POS analysis, spelling accuracy, lexical precision, and N-grams, facilitated a holistic assessment of essay content.

The results underscored the effectiveness of the chosen methodologies. Notably, the Random Forest model emerged as the top performer, demonstrating superior agreement with human graders, as evidenced by its Mean Absolute Error (MAE) of 1.00 and Quadratic Weighted Kappa (QWK) of 0.97. This robust performance was consistent with the LSTM model employing Word2Vec embeddings, which also achieved a QWK score of 0.97.

While the selected methodologies yielded promising results, it is crucial to consider alternative approaches and their potential outcomes. Firstly, the exploration of traditional machine learning models such as Support Vector Machine (SVM). Although SVM exhibited respectable performance, with a QWK score of 0.53, it lagged behind the Random Forest and LSTM models. This suggests that SVM might be better suited for simpler feature sets or could benefit from more extensive feature engineering.

Additionally, the adoption of state-of-the-art transformer-based models like BERT or GPT-3 could represent an intriguing avenue for future research. These models have demonstrated remarkable capabilities in natural language understanding tasks and might excel in capturing complex contextual patterns in essays. Employing such models could potentially elevate the performance standards in automated essay scoring.

It is essential to acknowledge the limitations of the developed models. One significant limitation lies in the generalizability of the results. While the models performed well on the ASAP dataset, their applicability to essays from different contexts, domains, or educational levels remains untested. Future research should include diverse datasets to enhance the robustness and generalizability of the models.

Another limitation relates to the availability and quality of human-graded essays in the training data. Biased or inconsistent grading can affect model performance. Employing crowd-sourcing or expert graders and refining the grading rubric can mitigate this issue.

Furthermore, grade normalization techniques introduced a trade-off between predictive accuracy and human grader agreement. Fine-tuning these techniques, such as exploring advanced statistical methods or leveraging external benchmarks, can help strike a balance between the two.

In conclusion, the methodology, result and discussion chapters provide valuable insights into the methodologies employed in automated essay grading. Achieving impressive results using NLP, machine learning, and deep learning techniques. Considering alternative approaches, addressing limitations, and refining the methods will be essential for advancing the field and ensuring the practical applicability of automated essay scoring systems.

Chapter 5

Conclusion and Future Work

5.1 Conclusion and Contribution

In this project, a comprehensive exploration of automated essay scoring through the lens of machine learning and deep learning methodologies was conducted. The significance of automated essay scoring in educational assessment and the need for robust and reliable scoring models were introduced.

Relevant literature in the domain, including the utilization of linguistic features, deep learning approaches, and the exploration of grade normalization techniques, was discussed, providing the groundwork for the research and highlighting gaps and opportunities within the field.

The methodology section laid out the foundation for the experiments, explaining the choice of models, input data, hyperparameters, and evaluation metrics. The training and testing procedures for Support Vector Machine (SVM), Random Forest, and Long Short-Term Memory (LSTM) models were detailed. Additionally, the significance of grade normalization in the investigation was emphasized.

Three main contributions guided the research, with each contribution representing a unique facet of automated essay scoring exploration. Contribution 1 aimed to systematically assess machine learning models' performance when applied to diverse essay features, resulting in notable distinctions between SVM and Random Forest models. Contribution 2 delved into the realm of deep learning, specifically focusing on LSTM networks and Word2Vec embeddings, achieving consistent high performance. Contribution 3 scrutinized the influence of grade normalization techniques on model performance, revealing trade-offs between predictive accuracy and agreement with human graders.

The results of the experiments provided valuable insights into the strengths and weaknesses of various models and approaches. Random Forest emerged as a robust performer, surpassing SVM, while LSTM models demonstrated remarkable consistency in performance. Grade normalization techniques highlighted the complex relationship between predictive accuracy and human grader agreement.

In conclusion, this project contributes significantly to the field of automated essay scoring by showcasing the superior performance of Random Forest and LSTM models and highlighting the importance of model selection and grade normalization techniques. The findings offer practical implications for educators and researchers seeking to enhance automated grading systems' effectiveness and consistency.

5.2 Future Work

Alternative Word Embeddings: While Word2Vec embeddings were employed in this study, future research could investigate alternative word embedding techniques such as Doc2Vec, FastText, GloVe, or BERT embeddings. Each of these methods captures different aspects of semantic and contextual information within essays. By experimenting with various embeddings, researchers can potentially uncover improved representations of essay content, leading to enhanced model performance (Dutta, 2022).

Exploring Advanced Deep Learning Architectures: This project primarily focused on LSTM networks with different layers. Future work could extend the exploration to more advanced deep learning architectures, such as Transformer-based models like BERT and GPT, which have demonstrated remarkable success in various natural language processing tasks. These architectures may capture more intricate contextual information and relationships within essays, potentially yielding even higher performance (Pedrycz and Chen, 2020). The use of BERT transformer model has shown significance in processing text in the health sector by (Afkanpour et al., 2022) which can be adapted for essay grading in the educational sector.

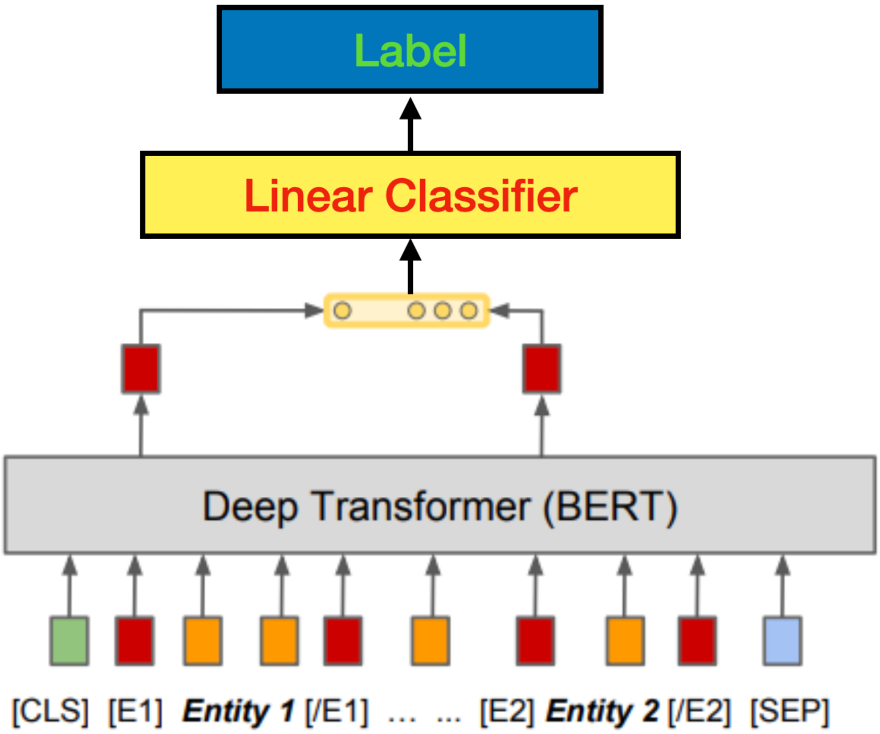

Label
Linear Classifier
Deep Transformer (BERT)
[CLS] [E1] *Entity 1* [/E1] ... ... [E2] *Entity 2* [/E2] [SEP]

Figure 5.1: BERT Model Architecture

Source: Adapted from *BERT(S) for Relation Extraction in NLP* by Wee Tee Soh, 2020, https://towardsdatascience.com/bert-s-for-relation-extraction-in-nlp-2c7c3ab487c4.

Enhancing Grade Normalization Techniques: The influence of grade normalization on model performance was explored, revealing a delicate balance between predictive accuracy and human grader agreement. Future work could delve deeper into grade normalization methods, exploring advanced statistical techniques and natural language processing approaches to improve both the accuracy and fairness of grading normalization. This can lead to more consistent and reliable automated essay scoring systems, aligning more closely with human grading standards (Singh and Singh, 2020).

Appendix A

Case Study

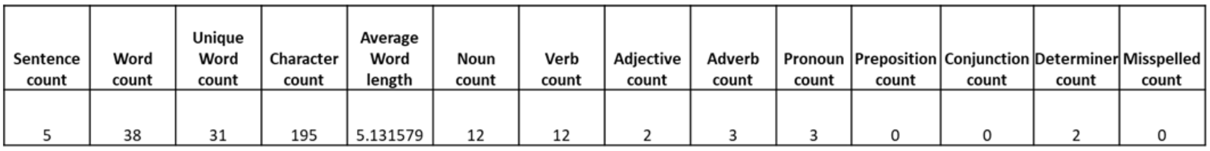
Case study 1: using SVM and random forest to predict the grade of essay with the syntactic, lexical, structural, and n-gram features

Input Essay: "The mood created by the author in the memoir would be described as happy and thankful. I say that because the author talked about how his parents were caring and selfless. They always helped out people in need. The author loved how he grew up with his Cuban culture and how he would never forget it. He said that was one household he would never forget."

Preprocessed and Tokenized Essay:

$$\begin{bmatrix} \text{"The", "mood", "created", "author", "memoir", "would", "describe", "happy", "thankful"} \\ \text{"I", "say", "author", "talked", "parents", "caring", "selfless"} \\ \text{"They", "always", "helped", "people", "need"} \\ \text{"The", "author", "loved", "grew", "Cuban", "culture", "would", "never", "forget"} \\ \text{"He", "said", "one", "house", "hold", "would", "never", "forget"} \end{bmatrix}$$

Extracted Features: The extracted features are in Figure A.1

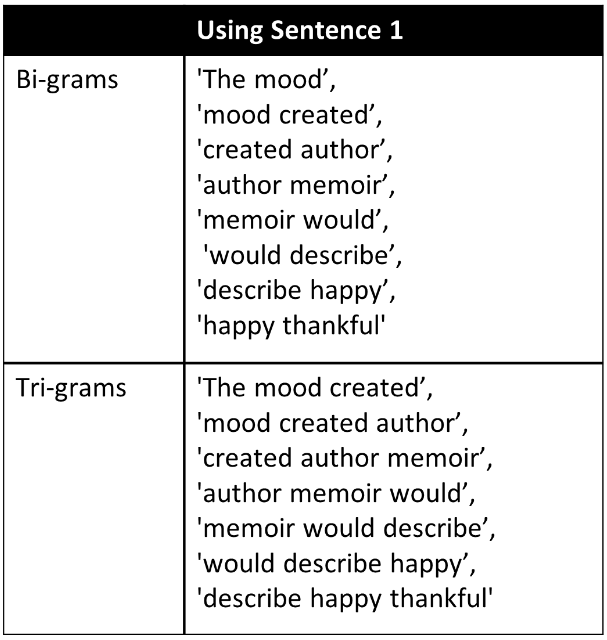


Sentence count	Word count	Unique Word count	Character count	Average Word length	Noun count	Verb count	Adjective count	Adverb count	Pronoun count	Preposition count	Conjunction count	Determiner count	Misspelled count
5	38	31	195	5.131579	12	12	2	3	3	0	0	2	0

Figure A.1: Structural, Lexical, Syntactical and Semantic Features

Extract N-Gram Words: This step involves extracting n-grams from the tokenized essay. They can be found in Figure A.2

Extracted Vectors from N-gram words using only the first sentence Bi-gram: Assuming we have only one sentence in the corpus and extracted 8 bi-grams and 7 tri-grams,



Using Sentence 1	
Bi-grams	'The mood’, 'mood created’, 'created author’, 'author memoir’, 'memoir would’, 'would describe’, 'describe happy’, 'happy thankful'
Tri-grams	'The mood created’, 'mood created author’, 'created author memoir’, 'author memoir would’, 'memoir would describe’, 'would describe happy’, 'describe happy thankful'

Figure A.2: N-Gram Words Example

the vector for the bi-grams alone would be:

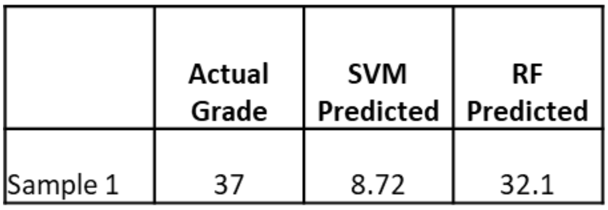
$$\begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & & & & & \end{bmatrix}$$

In this vector:

- The first eight elements correspond to the 8 bi-grams present in the first sentence. Each of these elements is set to 1 because each of these bi-grams appears once in the text.
- The remaining elements correspond to all the other tri-grams in the vocabulary (which are not present in the first bi-gram text), and they are all set to 0 because they do not appear in the bi-gram text.

Now once all the vectors are generated, the SVM and Random Forest is trained with the concatenated features.

The prediction for this essay obtained is in Figure A.3



	Actual Grade	SVM Predicted	RF Predicted
Sample 1	37	8.72	32.1

Figure A.3: Predicted SVM and Random Forest Grades

Case study 2: using LSTM to predict the grades of essays using 2-layer with word embeddings and comparing with a model that used normalized grades

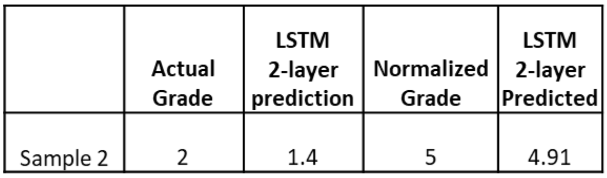
Input Essay: "Some of the obstacles the builders of the Empire State Building faced in attempting to allow dirigibles to dock there was that it wouldn't be able to tie cable tether because it adds stress to the building's frame. Also, in the story, they tell you how the steel frame of the Empire State Building would have to be modified and strengthened to accommodate the new situation."

Data Processing and Tokenization:

["Some", "obstacles", "builders", "Empire", "state", "Building", "faced", "attempting", "allow", "dirigib
"dock", "wouldnt", "able", "tie", "cable", "tether", "adds", "stress", "buildings", "frame"
"Also", "story", "tell", "steel", "frame", "Empire", "State", "Building", "would"
"modified", "strengthened", "accommodate", "new", "situation"

Word Embedding: Word Vectors are generated based on the context of words. Using a simplified example, The word "frame" would be captured in the context of a building, close to other architectural terms like "structure," "construction," or "edifice."

The prediction for this essay obtained is in Figure A.4



	Actual Grade	LSTM 2-layer prediction	Normalized Grade	LSTM 2-layer Predicted
Sample 2	2	1.4	5	4.91

Figure A.4: Predicted 2-Layer LSTM Grades with Unnormalized and Normalized Training sets

Comparing the two case studies: Count Vectorizer used in case study 1 represents text in a simple and interpretable format by focusing on word frequency. However, it does not capture semantic or contextual relationships, and it does not consider the order of words, which can result in a loss of sequential information. This makes it more suitable for tasks where the exact meaning of words or the word order is less critical, based on keyword presence.

On the other hand, Word Embeddings used in case study 2 address the challenge of capturing semantic and contextual relationships between words, placing words with similar meanings closer in the vector space. It also retains some information about word order and context, which can be valuable for tasks like natural language understanding. However, Word Embeddings can be less interpretable compared to Count Vectorizer, making it harder to understand the reasoning behind certain model decisions.

Choosing between these methods depends on the specific needs of the NLP task, where the balance between interpretability and contextual understanding plays a crucial role in determining which method to employ.

Appendix B

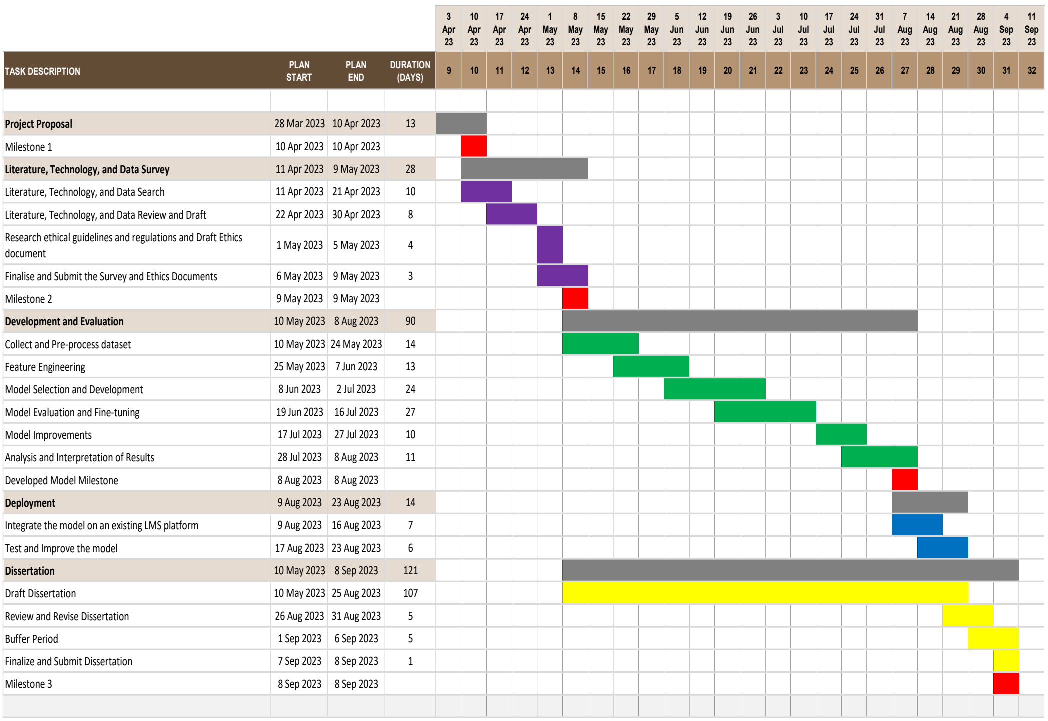
Project Plan

The project timeline has been carefully designed to allow sufficient time for each task, considering the need for breaks during the revision and exam period in May. Additionally, a 5-day buffer period has been incorporated to provide flexibility and account for any unexpected issues or delays that may arise towards the end of the project. The plan has been crafted to ensure that all deadlines and milestones are achievable and that there is ample time to review and refine work before submission.

B.1 Milestones and Deadlines

1. Project Proposal - Monday 10 April 2023, 8 PM
2. Literature, Technology, and Data Survey - Tuesday 9 May 2023, 8 PM
3. Dissertation – Friday 8 September 2023, 8 PM

B.2 Task and Gantt Chart



TASK DESCRIPTION	PLAN START	PLAN END	DURATION (DAYS)
Project Proposal	28 Mar 2023	10 Apr 2023	13
Milestone 1	10 Apr 2023	10 Apr 2023	
Literature, Technology, and Data Survey	11 Apr 2023	9 May 2023	28
Literature, Technology, and Data Search	11 Apr 2023	21 Apr 2023	10
Literature, Technology, and Data Review and Draft	22 Apr 2023	30 Apr 2023	8
Research ethical guidelines and regulations and Draft Ethics document	1 May 2023	5 May 2023	4
Finalise and Submit the Survey and Ethics Documents	6 May 2023	9 May 2023	3
Milestone 2	9 May 2023	9 May 2023	
Development and Evaluation	10 May 2023	8 Aug 2023	90
Collect and Pre-process dataset	10 May 2023	24 May 2023	14
Feature Engineering	25 May 2023	7 Jun 2023	13
Model Selection and Development	8 Jun 2023	2 Jul 2023	24
Model Evaluation and Fine-tuning	19 Jun 2023	16 Jul 2023	27
Model Improvements	17 Jul 2023	27 Jul 2023	10
Analysis and Interpretation of Results	28 Jul 2023	8 Aug 2023	11
Developed Model Milestone	8 Aug 2023	8 Aug 2023	
Deployment	9 Aug 2023	23 Aug 2023	14
Integrate the model on an existing LMS platform	9 Aug 2023	16 Aug 2023	7
Test and Improve the model	17 Aug 2023	23 Aug 2023	6
Dissertation	10 May 2023	8 Sep 2023	121
Draft Dissertation	10 May 2023	25 Aug 2023	107
Review and Revise Dissertation	26 Aug 2023	31 Aug 2023	5
Buffer Period	1 Sep 2023	6 Sep 2023	5
Finalize and Submit Dissertation	7 Sep 2023	8 Sep 2023	1
Milestone 3	8 Sep 2023	8 Sep 2023	

Figure B.1: Project Timeline

Appendix C

Resources

C.1 Literature Resource

C.1.1 Research Paper

This project utilized academic papers available through the university library and online databases such as Google Scholar, IEEE Xplore, ACM Digital Library, etc. to investigate any relevant literature and research that has been done in the area of automated essay grading using NLP.

C.1.2 Textbooks

This project made use of several textbooks available from the university library, such as Blueprints for Text Analytics using Python (Albrecht, Ramachandran and Winkler, 2021), Natural Language Processing in Action (Lane, Howard and Hapke, 2019) and Python Machine Learning (Raschka and Mirjalili, 2019) to gain a comprehensive understanding of natural language processing and machine learning techniques.

C.2 Technological Resource

C.2.1 Hardware

The development of this project made use of my personal laptop to train, evaluate the model and write the project report. Also, it made use of the university-provided internet connection to access online resources and cloud computing resources.

C.2.2 Software

This project is heavily based on Natural Language Processing and Machine learning utilizing various software tools and libraries to develop and evaluate the models. I primarily used Python programming language throughout this project with Jupypter Notebook. Pre-existing libraries were utilized such as Pandas for data exploration, cleaning, and pre-processing, Scikit-learn for the selection, training, and evaluation of SVM and Random forest models, Keras/tensorflow for building LSTM deep neural learning model, and NLTK for various Natural language processing tasks such as tokenization, etc.

C.2.3 Cloud Computing and Storage

The use of Cloud storage services such as Google Drive and Microsoft OneDrive was utilized for automatic backups of all project materials.

C.3 Data Resource

C.3.1 Secondary Data

The ASAP datasets were obtained from public online sources - Kaggle Repositories. The dataset contained 8 different 8 sets detailed below:

Essay set 1: Essay set 1 is a Persuasive/Narrative/Expository essay that contains 1783 training essays with a rubric score within the range of 1 – 6 for the 2 human raters and a resolved total score ranging from 2 – 12 which is a total score of both raters, the essays were graded based on the awareness of audience, organization and transitional language. The essays were written by Grade 8 students with the prompt to write a letter to persuade a local newspaper based on their opinions on the effects that computers have on people.

Essay set 2: Essay set 2 is a Persuasive/Narrative/Expository essay that contains 1800 essays which were scored using two domains – domain 1 is based on the writing applications with rubric range (1 – 6) for each rater and domain 2 is based on the language conventions with rubric range (1 – 4) for each rater. The score was then resolved to a total between the range of 1 – 6 and the essays were graded based on language conventions, voice, style, organization, ideas and content. The essays were written by Grade 10 students with the prompt to persuade a local newspaper based on their views of censorship in libraries and providing supporting information.

Essay set 3: Essay set 3 is a Source Dependent Response essay that contains 1726 essays written by Grade 10 students where they were given a passage from a book called "Rough Road Ahead" and were asked to write about their thoughts of the passage. The scoring rubric for each rater and the resolved score ranges from 0 to 3 and the essays were graded based on the understanding of complexities of the passage.

Essay set 4: Essay set 4 is a Source Dependent Response essay that contains 1770 essays written by Grade 10 students where they were given a passage titled "Winter Hibiscus by Minfong Ho" and were asked to provide a response as to why the author chose to conclude the story with a given paragraph. The scoring rubric for each rater and the resolved score ranged from 0 to 3 and the essays were graded based on the understanding of complexities of the passage.

Essay set 5: Essay set 5 is a Source Dependent Response essay that contains 1805 essays written by Grade 8 students where they were given a passage titled "Narciso Rodriguez by Narciso Rodriguez, from Home: The Blueprints of Our Lives" and were asked to narrate the mood illustrated by the author in the memoir. The scoring rubric for each rater and the resolved score ranges from 0 to 4.

Essay set 6: Essay set 6 is a Source Dependent Response essay that contains 1800 essays written by Grade 10 students where they were given a passage titled "The Mooring Mast by Marcia Amidon Lüsted" and were asked to use specific information from the passage to back up their understanding of a certain paragraph. The scoring rubric for each rater and the resolved score ranges from 0 to 4.

Essay set 7: Essay set 7 is a Persuasive/Narrative/Expository essay that contains 1569 training essays with a rubric score within the range of 0 – 15 for the 2 human raters and a resolved total score range of 0 – 30 which is the total score of both raters, the essays were graded based on idea, styles, organization and convention. The essays were written by Grade 7 students with the prompt to write a story about patience.

Essay set 8: Essay set 8 is a Persuasive/Narrative/Expository essay that contains 723 training essays with a rubric score within the range of 0 – 30 for the 2 human raters and a resolved total score range of 0 – 60 which is a total score of both raters, the essays were graded based on ideas and content, organization, sentence fluency and conventions. The essays were written by Grade 10 students with the prompt to write a true story that included laughter.

C.4 Expert Guidance Resource

Lastly, the expert guidance from my Supervisor, Dr. Alok Joshi was invaluable throughout this project. His overall guidance and direction ensured the project met all deadlines, executed successfully, and met the academic standards.

Appendix D

Ethical Considerations

The following ethical considerations were identified and addressed throughout the project:

1. Privacy and Confidentiality: Utilized anonymized datasets of graded essays to develop the model and took all necessary precautions to ensure the privacy and security of the data by ensuring that any identifiable information was removed from the datasets before use, and stored the data securely on cloud-based storage services such as Google Drive and Microsoft OneDrive.

2. Fairness and Bias: Implemented techniques such as cross-validation to ensure that the model is robust and not overly influenced by any particular subset of data.

3. Transparency: Utilized techniques and algorithms that are easy to understand and interpret, also documented the entire process thoroughly.

Appendix E

Code

This section outlines the main codes used for preprocessing and tokenization of essays, and the codes for feature extraction of the structural and lexical characteristics, the syntactic and linguistic characteristics, spelling characteristics, N-Grams and Word Embeddings.

Listing E.1: Preprocessing and Tokenization Code

```
def preprocess_and_tokenize_essay(essay):
    """
    Preprocess the essay by removing words starting with "@",
       removing stop words, and tokenizing into sentences and words.

    Args:
        essay (str): The input essay text.

    Returns:
        list of list of str: A list of tokenized sentences, where
           each sentence is a list of tokenized words.
    """
    # Clean the essay by removing words starting with "@"
    x = [i for i in essay.split() if not i.startswith("@")]
    removed_words = ' '.join(x)

    # Remove stop words
    stop_words = set(stopwords.words('english'))
    word_tokens = word_tokenize(removed_words)
    filtered_words = [w for w in word_tokens if w not in stop_words]

    # Tokenize into sentences
    tokenizer = nltk.data.load('tokenizers/punkt/english.pickle')
    raw_sentences = tokenizer.tokenize(' '.join(filtered_words))

    # Remove punctuation and tokenize words
    sentences = []
    for raw_sentence in raw_sentences:
        if len(raw_sentence) > 0:
            sentence_words = re.sub("[^A–Za–z0–9]", "", raw_sentence)
            words = nltk.word_tokenize(sentence_words)
            filtered_sentence_words = [w for w in words if w]
            sentences.append(filtered_sentence_words)
    return sentences
```

Listing E.2: Structural and Lexical Feature Extraction Code

```
def count_sentences(essay):
    """
    Count the number of sentences in the essay after preprocessing.

    Args:
        essay (str): The input essay text.

    Returns:
        int: The count of sentences.
    """
    sentences = preprocess_and_tokenize_essay(essay)
    return len(sentences)

def count_words(essay):
    """
    Count the number of words in the essay after preprocessing.

    Args:
        essay (str): The input essay text.

    Returns:
        int: The count of words.
    """
    sentences = preprocess_and_tokenize_essay(essay)
    word_count = sum(len(sentence) for sentence in sentences)
    return word_count

def count_unique_words(essay):
    """
    Count the number of unique words in the essay after
       preprocessing.

    Args:
        essay (str): The input essay text.

    Returns:
        int: The count of unique words.
    """
    sentences = preprocess_and_tokenize_essay(essay)
    all_words = [word for sentence in sentences for word in sentence]
    unique_word_count = len(set(all_words))
    return unique_word_count

def count_characters(essay):
    """
    Count the number of characters in the essay after preprocessing.
    Args:
        essay (str): The input essay text.

    Returns:
        int: The count of characters.
    """
    cleaned_essay = preprocess_and_tokenize_essay(essay)
    all_words = [word for sentence in cleaned_essay for word in
       sentence]
    total_characters = sum(len(word) for word in all_words)
    return total_characters

def calculate_average_word_length(essay):
    """
    Calculates the average word length in the essay after
       preprocessing.

    Args:
        essay (str): The input essay text.

    Returns:
        int: The average word length of essays.
    """
    cleaned_essay = preprocess_and_tokenize_essay(essay)
    all_words = [word for sentence in cleaned_essay for word in
       sentence]
    total_characters = sum(len(word) for word in all_words)
    total_words = len(all_words)
    if total_words > 0:
        average_word_length = total_characters / total_words
        return average_word_length
    else:
        return 0
```

Listing E.3: Syntactic and Linguistic Characteristics Code

```
def count_pos(essay):
    """
    Count parts of speech (POS) in the preprocessed essay.

    Args:
        essay (str): The input essay text.

    Returns:
        tuple: A tuple containing counts of different POS tags
           (Nouns, Verbs, Adjectives, Adverbs, Pronouns,
        Prepositions, Conjunctions, and Determiners).
    """
    sentences = preprocess_and_tokenize_essay(essay)

    noun_count = 0
    verb_count = 0
    adj_count = 0
    adverb_count = 0
    pronoun_count = 0
    preposition_count = 0
    conjunction_count = 0
    determiner_count = 0

    for sentence in sentences:
        pos_sentence = nltk.pos_tag(sentence)
        for word, pos_tag in pos_sentence:
            if pos_tag.startswith('N'):
                noun_count += 1
            elif pos_tag.startswith('V'):
                verb_count += 1
            elif pos_tag.startswith('J'):
                adj_count += 1
            elif pos_tag.startswith('R'):
                adverb_count += 1
            elif pos_tag.startswith('P'):
                pronoun_count += 1
            elif pos_tag.startswith('IN'):
                preposition_count += 1
            elif pos_tag.startswith('CC'):
                conjunction_count += 1
            elif pos_tag.startswith('DT'):
                determiner_count += 1
            elif pos_tag.startswith('PDT'):
                determiner_count += 1
            elif pos_tag.startswith('WDT'):
                determiner_count += 1

    return noun_count, verb_count, adj_count, adverb_count,
       pronoun_count, preposition_count, conjunction_count,
       determiner_count
```

Listing E.4: Spelling Characteristics Code

```
def count_spelling_errors(essay):
    """
    Count the spelling errors in the preprocessed essay.

    Args:
        essay (str): The input essay text.

    Returns:
        int: The count of misspelled words.
    """
    # Initialize a spell checker
    spell_checker = SpellChecker()

    # Tokenize the essay into words
    sentences = preprocess_and_tokenize_essay(essay)

    # Count the number of misspelled words
    misspelled_count = 0
    for sentence in sentences:
        for word in sentence:
            if word and not spell_checker.correction(word) == word:
                misspelled_count += 1

    return misspelled_count
```

Listing E.5: N-Grams Features Code

```
# This code uses CountVectorizer to convert text data into a n-grams and bag-of-words representation
# that includes unigrams, bigrams, and trigrams, and prepares it for modeling by creating a dense array for feature extraction.

vectorizer = CountVectorizer(max_features = 1000, ngram_range=(1, 3), stop_words='english')
count_vectors = vectorizer.fit_transform(df['token_essay'])
feature_names = vectorizer.get_feature_names_out()
X_cv = count_vectors.toarray()
```

Listing E.6: Word Embeddings Features Code

```
# Functions to generate feature vectors for words and essays using a Word2Vec model.

def generate_feature_vector(words, model, num_features):
    """
    Generate a feature vector for a list of words using the Word2Vec model.

    Args:
        words (list): A list of words.
        model (Word2Vec): The Word2Vec model for word embeddings.
        num_features (int): The number of features in the word embeddings.

    Returns:
        np.ndarray: The feature vector for the input list of words.
    """
    feature_vec = np.zeros(num_features, dtype="float32")
    num_words = 0

    for word in words:
        if word in model.wv:
            num_words += 1
            feature_vec = np.add(feature_vec, model.wv[word])

    if num_words > 0:
        feature_vec /= num_words
    return feature_vec

def generate_average_feature_vectors(essays, model, num_features):
    """
    Generate the average feature vectors for a list of essays using
        the Word2Vec model.

    Args:
        essays (list): A list of essays, where each essay is
            represented as a list of words.
        model: The Word2Vec model used for feature extraction.
        num_features (int): The number of features in the Word2Vec
            model.

    Returns:
        np.ndarray: An array of average feature vectors for each
            essay in the input list.
    """
    essay_feature_vecs = np.zeros((len(essays), num_features),
        dtype="float32")
    no_of_vec = 0

    for essay in essays:
        feature_vec = generate_feature_vector(essay, model,
            num_features)
        essay_feature_vecs[no_of_vec] = feature_vec
        no_of_vec += 1

    return essay_feature_vecs
```

Appendix F

Academic Paper

This dissertation has tremendous impact and advantage in the educational sector and after the submission of this dissertation to the University of Bath to fulfil the completion of my masters degree in Data Science. I would be preparing this project as an academic paper to be submitted under the support and guidance of my supervisor, Dr. Alok Joshi.

Appendix G

Word Count

This Dissertation Report contains a word count of **197 words** for the Abstract and **10,313 words** for the Main body. The main body is written on 28 pages. **Noting** that the required word count for the abstract is 200 and the main body is 10,000 with a 10% tolerance of up to 11,000 words, and the required number of pages is between 20 and 40 pages.

Bibliography

Adamson, A., Lamb, A. and December, R.M., 2014. Automated essay grading.

Adarkwah, M.A., 2021. The power of assessment feedback in teaching and learning: a narrative review and synthesis of the literature. *Sn social sciences* [Online], 1(4), p.75. Available from: https://doi.org/10.1007/s43545-021-00086-w.

Afkanpour, A., Adeel, S., Bassani, H., Epshteyn, A., Fan, H., Jones, I., Malihi, M., Nauth, A., Sinha, R., Woonna, S., Zamani, S., Kanal, E., Fomitchev, M. and Cheung, D., 2022. Bert for long documents: A case study of automated icd coding. 2211.02519.

Albrecht, J., Ramachandran, S. and Winkler, C., 2021. *Blueprints for text analytics using python machine learning-based solutions for common real world (nlp) applications*. O'Reilly. Available from: https://learning.oreilly.com/library/view/blueprints-for-text/9781492074076/.

Beheshti, N., 2022. Random forest regression. Available from: https://towardsdatascience.com/random-forest-regression-5f605132d19d.

Ben Hamner, Jaison Morgan, l.M.S.T.V.A., 2012. The hewlett foundation: Automated essay scoring [Online]. Available from: https://kaggle.com/competitions/asap-aes.

Blei, D.M., Ng, A.Y. and Jordan, M.I., 2003. Latent dirichlet allocation. *Journal of machine learning research*, 3(Jan), pp.993–1022.

Chen, Y., Liu, C.L., Lee, C. and Chang, T., 2010. American EnglishAn unsupervised automated essay-scoring system. *Ieee intelligent systems* [Online], 25(5), pp.61–67. Available from: https://doi.org/10.1109/MIS.2010.3.

Cutrone, L., Chang, M. and Kinshuk, 2011. Auto-assessor: Computerized assessment system for marking student's short-answers automatically [Online]. *2011 ieee international conference on technology for education*. pp.81–88. Available from: https://doi.org/10.1109/T4E.2011.21.

Dasgupta, T., Naskar, A., Dey, L. and Saha, R., 2018. Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring [Online]. *Proceedings of the 5th workshop on natural language processing techniques for educational applications*. Melbourne, Australia: Association for Computational Linguistics, pp.93–102. Available from: https://doi.org/10.18653/v1/W18-3713.

Devi, M.S. and Mittal, H., 2016. Machine learning techniques with ontology for subjective answer evaluation. *Corr* [Online], abs/1605.02442. 1605.02442, Available from: http://arxiv.org/abs/1605.02442.

Dutta, M., 2022. Word2vec for word embeddings -a beginner's guide. Available from: https://www.analyticsvidhya.com/blog/2021/07/word2vec-for-word-embeddings-a-beginners-guide/.

Examples.com, Accessed 2023. Essay: Purposes, types and examples. https://www.examples.com/education/essays-examples.html.

Frost, J., 2022. Inter-rater reliability: Definition, examples amp; assessing. Available from: https://statisticsbyjim.com/hypothesis-testing/inter-rater-reliability/.

Frost, J., 2023a. Mean squared error (mse). Available from: https://statisticsbyjim.com/regression/mean-squared-error-mse/.

Frost, J., 2023b. Root mean square error (rmse). Available from: https://statisticsbyjim.com/regression/root-mean-square-error-rmse/.

Gandhi, R., 2018. Support vector machine — introduction to machine learning algorithms [Online]. Towards Data Science. Accessed on May 8, 2023. Available from: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47.

Ghanta, H., 2019. *Automated essay evaluation using natural language processing and machine learning*. Ph.D. thesis.

Hearst, M., 2000. The debate on automated essay grading. *Intelligent systems and their applications, ieee* [Online], 15, pp.22 – 37. Available from: https://doi.org/10.1109/5254.889104.

Hundley, D., 2022. How and why to perform a k-fold cross validation. Available from: https://towardsdatascience.com/how-and-why-to-perform-a-k-folds-cross-validation-adf88665893b.

Hussein, M.A., Hassan, H. and Nassef, M., 2019. Automated language essay scoring systems: a literature review. *Peerj. computer science* [Online], 5, p.e208. Available from: https://doi.org/10.7717/peerj-cs.208.

IBM, I., n.d. What is natural language processing? Available from: https://www.ibm.com/uk-en/topics/natural-language-processing.

Jain, A., Kulkarni, G. and Shah, V., 2018. Natural language processing. *International journal of computer sciences and engineering* [Online], 6, pp.161–167. Available from: https://doi.org/10.26438/ijcse/v6i1.161167.

Kaur, A. and Sasi Kumar, M., 2019. Performance analysis of lsa for descriptive answer assessment. In: H.S. Saini, R. Sayal, A. Govardhan and R. Buyya, eds. *Innovations in computer science and engineering*. Singapore: Springer Singapore, pp.57–63.

Kim, Y.S., Oh, J.S., Lee, J.Y. and Chang, J.H., 2004. An intelligent grading system for descriptive examination papers based on probabilistic latent semantic analysis [Online]. pp.1141–1146. Available from: https://doi.org/10.1007/978-3-540-30549-1_114.

Kulshrestha, R., 2020. Nlp 101: Word2vec-skip-gram and cbow. Available from: https://towardsdatascience.com/nlp-101-word2vec-skip-gram-and-cbow-93512ee24314.

Kumar, V. and Boulanger, D., 2020. Explainable automated essay scoring: Deep learning really has pedagogical value. *Frontiers in education* [Online], 5. Available from: https://doi.org/10.3389/feduc.2020.572367.

Landauer, T., Foltz, P. and Laham, D., 1998. An introduction to latent semantic analysis. *Discourse processes* [Online], 25, pp.259–284. Available from: https://doi.org/10.1080/01638539809545028.

Landauer, T., Laham, D. and Foltz, P., 2000. The intelligent essay assessor. *Intelligent systems, ieee*, 15, pp.27–31.

Lane, H., Howard, C. and Hapke, H.M., 2019. *Natural language processing in action: Understanding, analyzing, and generating text with python*. Manning Publications Co.

Lexalytics, 2023. *Context analysis in nlp: Why it's important and how it works* [Online]. Available from: https://www.lexalytics.com/blog/context-analysis-nlp/ [Accessed 2023-05-08].

Li, Y. and Yan, Y., 2012. An effective automated essay scoring system using support vector regression [Online]. *2012 fifth international conference on intelligent computation technology and automation*. pp.65–68. Available from: https://doi.org/10.1109/ICICTA.2012.23.

Liang, G., On, B.W., Jeong, D., Kim, H.C. and Choi, G.S., 2018. Automated essay scoring: A siamese bidirectional lstm neural network architecture. *Symmetry* [Online], 10(12). Available from: https://doi.org/10.3390/sym10120682.

Mahana, M. and Apte, A.A., 2012. Automated essay grading using machine learning.

MasterClass, 2021. *Guide to common types of essays* [Online]. Available from: https://www.masterclass.com/articles/guide-to-common-types-of-essays [Accessed 2023-05-08].

Matthews, K., Janicki, T., He, L. and Patterson, L., 2012. Implementation of an automated grading system with an adaptive learning component to affect student feedback and response time. *Journal of information systems*, 23, pp.71–84.

McNamara, D.S., Graesser, A.C., McCarthy, P.M. and Cai, Z., 2014. *Automated evaluation of text and discourse with coh-metrix* [Online]. Cambridge University Press. Available from: https://doi.org/10.1017/CBO9780511894664.

Naveen, 2023. Tokenization and stopword removal. Available from: https://www.nomidl.com/natural-language-processing/tokenization-and-stopwordremoval/#:~:text=Tokenization%20helps%20to%20transform%20the%20text%20data%20into,improving%20the%20efficiency%20of%20NLP%20algorithms%20and%20models.

Page, E.B., 2003. Project essay grade: Peg. *Automated essay scoring: A cross-disciplinary perspective*. Lawrence Erlbaum Associates Publishers, pp.43–54.

Patil, S.M. and Patil, S., 2014. Evaluating student descriptive answers using natural language processing. *International journal of engineering research and technology*, 03(03), pp.1335–1339.

Pedrycz, W. and Chen, S.M., eds, 2020. *Deep learning architectures* [Online], Cham:

Springer International Publishing, pp.1–24. Available from: https://doi.org/10.1007/978-3-030-31756-0_1.

Pisarova, M., 2021. Datasets in natural language processing, mt ai. Available from: https://www.translateplus.com/blog/datasets-and-natural-language-processing-ai-translation/.

Poudel, A.P., 2018. Academic writing: Coherence and cohesion in paragraph.

Powers, D.E., Burstein, J.C., Chodorow, M., Fowles, M.E. and Kukich, K., 2002. Stumping e-rater: challenging the validity of automated essay scoring. *Computers in human behavior*, 18(2), pp.103–134.

Ramesh, D. and Sanampudi, S., 2022. An automated essay scoring systems: a systematic literature review. *Artificial intelligence review* [Online], 55, pp.2495–2527. Available from: https://doi.org/10.1007/s10462-021-10068-2.

Raschka, S. and Mirjalili, V., 2019. *Python machine learning: Machine learning and deep learning with python, scikit-learn, and tensorflow 2*. Packt Publishing Ltd.

Rokade, A., Patil, B., Rajani, S., Revandkar, S. and Shedge, R., 2018. Automated grading system using natural language processing [Online]. *2018 second international conference on inventive communication and computational technologies (icicct)*. pp.1123–1127. Available from: https://doi.org/10.1109/ICICCT.2018.8473170.

Rudner, L.M., Garcia, V. and Welch, C., 2006. An evaluation of intellimetric™ essay scoring system. *The journal of technology, learning and assessment* [Online], 4(4). Available from: https://ejournals.bc.edu/index.php/jtla/article/view/1651.

Rudner, L.M. and Liang, T., 2002. Automated essay scoring using bayes' theorem.

Schinske, J. and Tanner, K., 2014. Teaching more by grading less (or differently). Available from: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4041495/.

Sethi, A., 2023. Support vector regression tutorial for machine learning. Available from: https://www.analyticsvidhya.com/blog/2020/03/support-vector-regression-tutorial-for-machine-learning/.

Shehab, A., Elhoseny, M. and Hassanien, A.E., 2016. A hybrid scheme for automated essay grading based on lvq and nlp techniques [Online]. pp.65–70. Available from: https://doi.org/10.1109/ICENCO.2016.7856447.

Shermis, M.D., Mzumara, H.R., Olson, J. and Harrington, S., 2001. On-line grading of student essays: Peg goes on the world wide web. *Assessment & evaluation in higher education*, 26(3), pp.247–259.

Singh, D. and Singh, B., 2020. Investigating the impact of data normalization on classification performance. *Applied soft computing* [Online], 97, p.105524. Available from: https://doi.org/https://doi.org/10.1016/j.asoc.2019.105524.

Song, S. and Zhao, J., 2013. Automated essay scoring using machine learning [Online]. Available from: https://nlp.stanford.edu/courses/cs224n/2013/reports/song.pdf.

Taghipour, K. and Ng, H.T., 2016. A neural approach to automated essay scoring [Online]. *Proceedings of the 2016 conference on empirical methods in natural language processing*.

Austin, Texas: Association for Computational Linguistics, pp.1882–1891. Available from: https://doi.org/10.18653/v1/D16-1193.

Turner, C., 2018. More states opting to 'robo-grade' student essays by computer [Online]. Available from: https://www.npr.org/2018/06/30/624373367/more-states-opting-to-robo-grade-student-essays-by-computer.

Verma, G. and Srinivasan, B.V., 2019. A lexical, syntactic, and semantic perspective for understanding style in text. Available from: https://arxiv.org/abs/1909.08349.

Zhao, S., Zhang, Y., Xiong, X., Botelho, A. and Heffernan, N., 2017. A memory-augmented neural model for automated grading [Online]. pp.189–192. Available from: https://doi.org/10.1145/3051457.3053982.

Zhu, W. and Sun, Y., 2020. Automated essay scoring system using multi-model machine learning. *Cs & it conference proceedings*. CS & IT Conference Proceedings, vol. 10.