A Comparative Study of Methods for Automated Essay Grading

Ishaan Arya

July 2021

Abstract

Essays hold paramount importance in assessments around the world. The grading of student written essays is a time-consuming, ambiguous, and expensive process. Therefore, various Automated Essay Scoring (AES) systems have been developed over time to automate the essay grading process. This paper presents a comparative study of the most effective machine learning models used for AES by training and testing them on the Kaggle Automated Student Assessment Prize (ASAP) dataset. The models compared include linear regression, random forest, support vector machine, naive Bayes, and multinomial logistic regression. This study aims to help future researchers choose the best models for creating novel AES systems. For model training, the data was first cleaned and normalised. Then, statistical features which best related to essay quality were extracted and the models were trained with an 80-20 train test split. Accuracy was used as the metric for evaluating the performance of different models, and due to the subjectivity of essay grading, any score prediction within one score range above or below the actual score was considered an accurate prediction. The results obtained show that random forest regression, random forest classification, and support vector regression were the best performing models for this problem on this dataset, reporting an ac-

curacy of 95.60%. In the future, this study could be extended to different machine learning models, and the effect of features and hyperparameters could be explored in greater depth.

1 Introduction

Essays are considered to be a great indicator of higher-level cognitive skills and knowledge in students. They have been used as a medium of assessment for years. However, students and educators worldwide have faced an incessant problem with this approach: the subjectivity and laboriousness of essay grading. Different examiners have different ways of interpreting and evaluating essays, causing discrepancies in how essays are scored. Additionally, due to rising global youth population, the volume of students is escalating and grading essays has become increasingly time-consuming and expensive. The desire to solve this problem has resulted in the development of various Automated Essay Scoring (AES) systems. These systems utilize computerized scoring to grade essays in a quick and unprejudiced way.

The purpose of this paper is to present a comparative study of the most commonly used systems for automated essay grading. In section 2, a literature review on predominant AES systems is presented to contextualize my research goals. In section 3, the primary machine learning models used in literature and part of my AES evaluation process are presented. These include linear regression, random forest, support vector machine, naive Bayes, and multinomial logistic regression. In section 4, my research methodology is outlined, which incorporates data processing, model training, and evaluation. In section 5, the primary results of the investigation are presented. In section 6, these results are analysed. In section 7, a conclusion is presented and future work on this topic is proposed. The primary goal of this paper is to compare how different machine learning models perform on this dataset for the automated essay grad-

ing problem, and hence, aid future researchers in selecting the best models for novel AES systems.

2 Literature Review

The origin of AES systems can be traced back to the Project Essay Grade (PEG) proposed by Ellis B. Page in 1966, which used various statistical metrics to grade student essays [17]. Since then, numerous approaches have been used to create better AES systems. In the late nineties, the Intelligent Essay Assessor (IEA) was developed, based on the Latent Sentiment Analysis (LSA) technique, initially designed for indexing and retrieval [7]. This method represents the essay as a matrix, uses Single Value Decomposition (SVD) to reduce the dimensions of the matrix, and then compares its similarity with a model answer using cosine correlation [24]. Closely following this, in 1998, the Electronic Essay Rater (E-rater) was developed by Burstein and others, which used various statistical and Natural Language Processing (NLP) methods for analyzing syntactical, structural, and topical features [4]. The E-rater V.2, developed in 2006, built on the initial E-rater by taking a smaller and more meaningful feature set and combining these features in a simple and intuitive way [3]. All of these systems came close to emulating human graders; however, they all had various shortcomings and never got adopted as mainstream technology.

According to a study conducted by Curran and others in 2013, students were skeptical of the PEG software assigning them grades, having concerns about fairness, technology acceptance, and their relationships with faculty [5]. Another study conducted in 2014 revealed that faculty also strongly opposed the use of the PEG software for essay grading [6]. In the book Machine Scoring of Student Essays: Truth and Consequences, Tim McGee presents three experiments that clearly demonstrate the inadequacies of the IEA [15]. Research also suggests that the e-rater system fails to appreciate certain aspects of writing

and undeservedly rewards others, giving unreliable scores [20, 21].

Since then, due to the rapid rise in machine learning techniques, multiple machine learning models have been used for improved automated essay grading. These are primarily supervised learning approaches which recast the task as (1) a regression task, where a continuous essay score is predicted; (2) a classification task, where an essay is classified into a smaller subset of all essays; (3) a ranking task, where two or more essays are compared and ranked based on their quality; or (4) a neural approach, where neural networks are used to make predictions [11]. For regression, the models primarily used in literature include linear regression [2, 12, 19], support vector regression [13, 18], and random forest regression [1, 9]. The models used for classification include logistic regression [10, 8] and naive Bayes classification [14, 22]. For ranking, Support Vector Machine (SVM) ranking [25] is majorly used. Recent AES systems also utilize neural approaches, primarily through Long and Short Term Memory (LSTM) models [23, 16].

3 Models

The machine learning models compared in this study were chosen based on their performance in literature. The highest performing machine learning models for AES included linear regression, random forest, support vector machine, naive Bayes, and multinomial logistic regression. The predominant goal of this section is to present the theoretical background of these machine learning models.

3.1 Linear Regression

Linear regression is a mathematical technique for modelling a linear relationship between the input variables and the output. Linear regression aims to formulate an equation which describes the output as a linear combination of the input

variables. This equation is in the form

$$y = b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n \tag{1}$$

$x_1...x_n$ refer to the features of an essay, which, when combined with certain weights, $b_0...b_n$, give a continuous value for the essay score, y.

3.2 Random Forest

Random forest regression is a supervised learning algorithm which builds an ensemble of decision trees in order to carry out regression or classification. The training data is split into random subsets and a decision tree is created from every subset of the data.

In a decision tree regression, the data is split according to a condition at every node, until eventually a continuous score can be predicted. The random forest regression model takes the mean of the essay score predictions of all the decision trees constructed in order to make a prediction.

In decision tree classification, the data is split according to a condition at every node, changing the probabilities of different score classes occurring. The tree is traversed and the score class with the highest probability of occurring is returned. The random forest classification model will classify the essay to the highest occurring score class across all decision trees.

3.3 Support Vector Machine

Support vector machine is a supervised machine learning algorithm which works by plotting all the essays as a point or vector in an n-dimensional space according to their features, and finding a hyperplane to fit the data and perform regression or classification.

Support vector classification uses the hyperplane to separate the essay vectors into distinct score classes. An optimal hyperplane is constructed by maximising

the distance between data points of different classes. The model will make predictions by plotting an essay vector on the same plane and identifying the score class it belongs to.
Support vector regression uses the hyperplane as a line of best fit which helps predict continuous values for the essay score.

3.4 Naive Bayes

Naive Bayes is a supervised learning classifier which uses Bayes' theorem to make predictions. Given a set of input feature vectors $x_1...x_n$ and the class variable y, Bayes' theorem gives the following relationship.

$$P(y \mid x_1, x_2, x_3...x_n) = \frac{P(x_1 \mid y)P(x_2 \mid y)...P(x_n \mid y)P(y)}{P(x_1)P(x_2)...P(x_n)} \tag{2}$$

The denominator remains constant given an input set of features, hence:

$$P(y \mid x_1, x_2, x_3...x_n) \propto P(x_1 \mid y)P(x_2 \mid y)...P(x_n \mid y)P(y) \tag{3}$$

$$P(y \mid x_1, x_2, x_3...x_n) \propto P(y)\prod_{i=1}^{n} P(x_i \mid y) \tag{4}$$

This is the Naive Bayes probability model; it uses the input feature vectors to predict the probability of every essay score occurring. The Naive Bayes classifier combines this model with a decision rule. The decision rule is to classify the essay to the score which has the highest probability of occurring. This is represented as:

$$y = argmax_y \ P(y)\prod_{i=1}^{n} P(x_i \mid y) \tag{5}$$

There are three types of naive Bayes classifiers that are predominantly used: Multinomial, Gaussian, and Bernoulli, considering different data distributions.

3.5 Multinomial Logistic Regression

Logistic Regression is a supervised learning algorithm used for binary classification. The softmax function extends this to multinomial logistic regression which is useful in automated essay grading as the essay can be classified into more than 2 score classes. The softmax function is defined as:

$$softmax(z) = \frac{e^z}{\sum_{i=1}^{k} e^{z_i}} \tag{6}$$

Given an input vector z, the softmax function makes all values positive using e^z, and then normalises them so that their cumulative sum is 1. This gives a discrete probability distribution across k classes. The vector z is the combination of all the features and weights for an essay. The whole model can be represented as:

$$y = softmax(b_0 + b_1x_1 + b_2x_2 + ... + b_nx_n) \tag{7}$$

This gives a list of probabilities of the essay vector belonging to different score classes. The essay is then classified to the score class with the highest probability.

4 Methodology

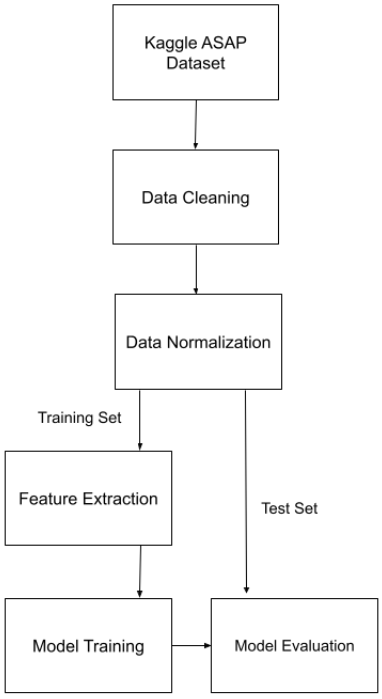

Figure 1: System Architecture

4.1 Dataset Description

In 2012, the William and Flora Hewlett Foundation sponsored the Automated Student Assessment Prize (ASAP) competition on Kaggle[1], seeking a fast, effective, and affordable solution to the automated essay grading problem. The openly available dataset provided with this problem contains 12,976 essays of approximately 150 to 500 words in ASCII text, written by students from grade 7 to 10 and scored by human assessors. Essay sets 1 to 6 of this dataset were

[1] https://www.kaggle.com/c/asap-aes

used to train and test the models in this paper, with a total of 10,684 essays, of which 8547 were used for training and 2137 were used for testing.

4.2 Data Cleaning

The essays from sets 1 through 6 of the Kaggle dataset were cleaned using the following steps:

- Personally identifying information like names, locations and numbers were replaced in the Kaggle dataset with strings like @PERSON1, @LOCATION1, and @NUM1. These do not effect an essay's score and hence, all words following @ were removed from the dataset.
- Stopwords are unimportant words in an essay such as 'a', 'an', and 'the'. These words are trivial and do not add anything meaningful to an essay. Hence, in order to allow the models to focus on more important words, these words were filtered out from all essays. A collection of stopwords was obtained using the NLTK library[2] in python.

4.3 Data Normalisation

The six essay sets chosen had different score ranges and hence, the essay scores had to be normalised before machine learning could be performed.

[2]https://www.nltk.org

Essay Set	Score Range
1	2-12
2	1-6
3	0-3
4	0-3
5	0-4
6	0-4

Table 1: Score Ranges of the 6 Essay Sets Being Used

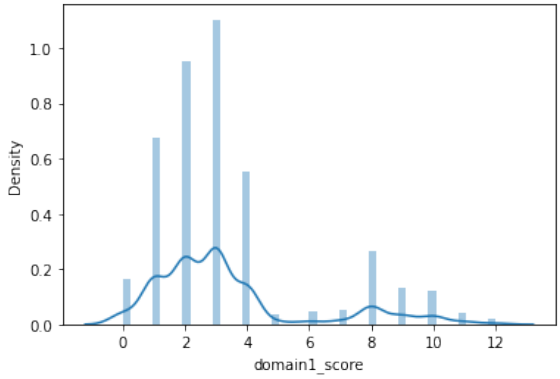

Figure 2: Score distribution before data normalisation

Table 1 shows the initial score ranges of the six essay sets. Figure 2 illustrates the score distribution across score classes before data normalisation was performed. The essay scores were normalised using min-max normalisation to transform them into a decimal between 0 and 1. The formula for min-max normalisation is:

$$x = \frac{x - min(x)}{max(x) - min(x)} \tag{8}$$

This value was multiplied by 10 and rounded off to the nearest whole number in order to normalise all the scores between 0 and 10.

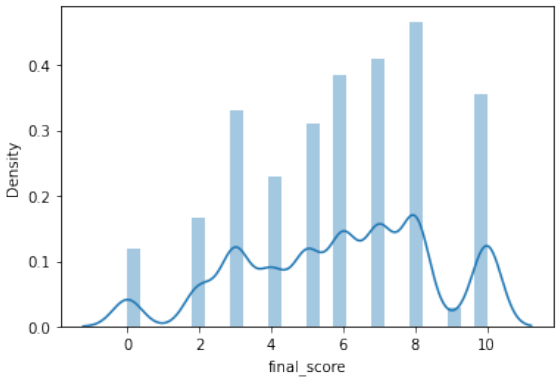

Figure 3: Score distribution after data normalisation

Figure 2 displays the score distribution after data normalisation was performed. All the scores were transformed to the 0 to 10 score range, allowing for comparisons between them.

4.4 Feature Extraction

Feature extraction is a vital element in building an effective automated essay grading system. Therefore, statistical features which could best correlate with essay quality were found. Eventually, the following features were selected, which remained consistent across all the machine learning models compared:

1) Number of words, characters, and sentences: These are basic features of any text document, however, they can have an impact on essay quality. I used the python NLTK library to split the essay and get the word count, sentence count, and character count of each essay.

2) Part Of Speech (POS) Count: A metric that could be used to evaluate the quality of writing is the count of each POS class present. I found the number of words in the syntactical classes of nouns, verbs, adjectives and adverbs for each essay using the POS Tags in the NLTK library.

3) Orthography: The number of spelling mistakes in an essay could be an impor-

tant feature for determining an essay's score. I used the pyspellchecker library[3] for this.

4.5 Hyperparameter Tuning

In order to achieve the highest accuracy for every model, the hyperparameters had to be tuned. This wasn't required for linear regression, logistic regression and naive Bayes as they don't have any critical hyperparameters. However, this was performed for random forest and support vector machine.

4.5.1 Random Forest

The random forest regression and classification models had the n-estimators hyperparameter. The value provided for this determines the number of decision trees that would be constructed. Higher number of trees would give better model performance, however, would take longer to execute. In order to find the optimal value for the n-estimators, I plotted its values from 0 to 250 against the mean squared error for random forest regression, and against accuracy for random forest classification.

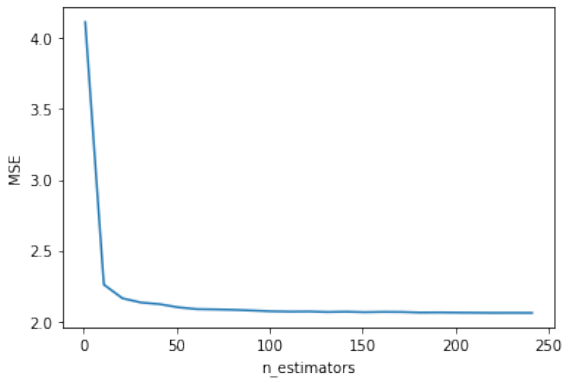

Figure 4: Random Forest Regression: n-estimators against mean squared error

[3]https://pyspellchecker.readthedocs.io

As evident from Figure 4, the mean squared error decreases as the number of regression decision trees constructed is increased. This is more pronounced in the beginning, and after 150 n-estimators, successive iterations do not offer significant reductions in the mean squared error. Therefore, the value of 150 was chosen for the n-estimators in the random forest regression model.

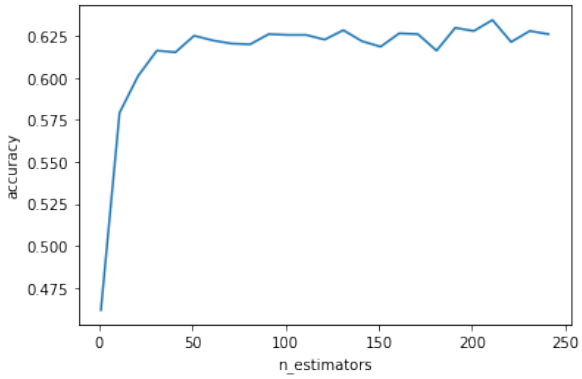

Figure 5: Random Forest Classification: n-estimators against accuracy

As evident from Figure 5, the accuracy for the random forest classifier increases rapidly in the beginning and then fluctuates, with 210 n-estimators giving the highest accuracy. Therefore, 210 was chosen for the value of n-estimators in the random forest classification model.

4.5.2 Support Vector Machine

Support Vector Machines have the C hyperparameter. The C value governs how correct the classification of data points should be. The larger the C value, the more accurately all the points will be classified during training. However, if the C value is too large, it would result in overfitting and hence, lower accuracy. In order to obtain the optimal C value, C values in multiples of 10 from 0.01 to 100000 were plotted against the mean squared error for support vector

regression, and against accuracy for support vector classification.

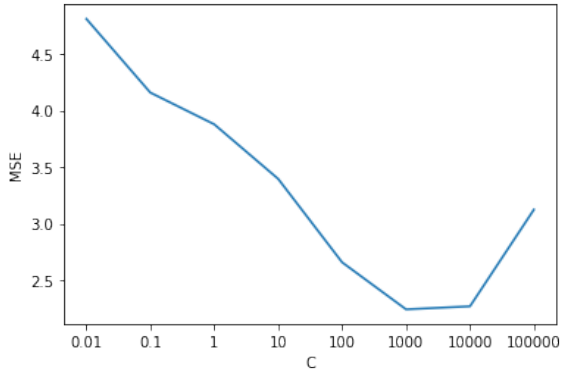

Figure 6: SVR: C against accuracy

As clear from Figure 6, the mean squared error in support vector regression gradually decreases until the C value of 1000, after which it starts to increase. Hence, the value of 1000 was chosen for C for support vector regression.

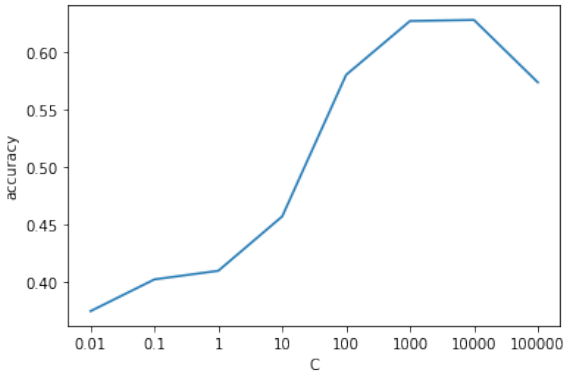

Figure 7: SVM: C against accuracy

As can be observed from Figure 7, the accuracy of the support vector classifier increases as the C value is increased up to 1000. However, when it is increased to 10000, the accuracy remains constant and at, 100000, the accuracy

decreases as a result of overfitting. Therefore, the value of 1000 was chosen for C for support vector classification as well.

4.6 Model Training

The selected machine learning models were trained using the scikitlearn library[4] in python with an 80-20 train test split. A random state of 42 was used in order to ensure that the training and testing splits remained consistent across all models for a fair comparison.

4.7 Evaluation

I used accuracy as the metric for evaluating both regression and classification models. Accuracy simply measures the percentage of the total number of essay scores the model predicted correctly. I computed this by counting the predicted values which matched with the values assigned by human raters, diving this value with the total number of essays in the test set, and multiplying by 100 to get a percentage. Regression models predicted a continuous value, hence, this value had to first be rounded off to the nearest whole number before it could be compared with the actual scores in the test data. Additionally, my rationale was that essay grading is subjective and hence, any predicted score within the range of one score bracket above or below the actual score was considered to be an accurate prediction.

5 Results

In this section, findings of the experiment are presented, comparing linear regression, support vector machine, random forest, naive Bayes, and multinomial

[4]https://scikit-learn.org

logistic regression for automated essay grading. The presented results are reproducible and the code for the same can be found on the accompanying Github repository[5].

5.1 Regression Models

Model	Accuracy
Linear Regression	52.36%
Support Vector Regression	95.60%
Random Forest Regression	95.60%

Table 2: Accuracy Scores for Regression Models

5.2 Classification Models

Model		Accuracy
Naive Bayes	Multinomial	55.55%
	Gaussian	56.29%
	Bernoulli	79.41%
Random Forest Classification		95.60%
Support Vector Classification		91.44%
Logistic Regression		81.94%

Table 3: Accuracy Scores for Classification Models

The results demonstrate that random forest regression, random forest classification, and support vector regression were the best performing machine learning models for this problem on this dataset, with an accuracy of 95.60%. Linear regression, along with multinomial and Gaussian naive Bayes seemed to report

[5]https://github.com/ishaan-arya/automated-essay-grading

quite low accuracies, while naive Bayes Bernoulli, support vector classification, and logistic regression seemed to give quite highly accurate predictions.

6 Analysis

From the presented research, it is evident that the selected machine learning models vary considerably for automated essay grading on the Kaggle dataset. Among regression models in Table 2, linear regression reported the lowest accuracy of 52.36%, possibly because this model assumes that features are independent of each other and is sensitive to outliers in the data. Random forest regression and support vector regression seemed to be very effective for this problem on this dataset, with both reporting an accuracy of 95.60%. This is probably because random forest chooses features during training and takes the mean of various decision trees, preventing overfitting and bias and giving a more reliable prediction. Support vector regression is also robust to outliers, and can generalise effectively, giving high testing accuracy.

Among the classification models in Table 3, naive Bayes reported average accuracies, with multinomial giving the lowest accuracy of 55.55%, Gaussian slightly outperforming it with an accuracy of 56.29%, and Bernoulli reporting a notably higher, 79.41% accuracy. In general, the accuracies for naive Bayes were lower than other classification models, possibly due to the assumption of independent predictor features which doesn't hold true for automated essay grading. Logistic regression also has this assumption and hence, it was only slightly better than naive Bayes Bernoulli, with an accuracy of 81.94%. Support vector classification seemed to be a more effective model, with an accuracy of 91.44%, as it avoids overfitting and holds great generalisation capabilities. Random forest classifier reported an accuracy of 95.60%, the same as random forest regression and support vector regression. This is probably because random forest regression and classification split the data into features in the same

way, which seems to be highly effective for automated essay grading.

7 Conclusion and Future Work

Through comparing different models for automated essay grading on the Kaggle dataset, this paper has demonstrated that differences between how the selected machine learning models make predictions have a significant impact on performance. By experimentally implementing these models on student-written essays and calculating their accuracies, this paper was able to present a comparison of the selected models and establish that the random forest regression, random forest classification, and support vector regression models delivered the best performance for this problem on this dataset, with an accuracy of 95.60%. In the future, this comparative study can be extended to other machine learning and deep learning models including BERT (Bidirectional Encoder Representations from Transformers) and CNN (Convolutional Neural Network), and the effect of other variables like features and hyperparameters could be more comprehensively analysed to help researchers build more robust and effective automated essay grading systems. These systems could potentially revolutionize the field of education by allowing for more consistent scoring, reducing costs, and saving students and educators valuable time. However, this technology still has a lot to prove and further research is required before it can become a reality in educational institutions around the world.

References

[1] Arshad Arafat, Mohammed Raihanuzzaman, et al. *Automated essay grading with recommendation.* PhD thesis, BRAC University, 2016.

[2] Yigal Attali and Jill Burstein. Automated essay scoring with e-rater® v. 2.0. *ETS Research Report Series*, 2004(2):i–21, 2004.

[3] Yigal Attali and Jill Burstein. Automated essay scoring with e-rater® v. 2. *The Journal of Technology, Learning and Assessment*, 4(3), 2006.

[4] Jill Burstein, Karen Kukich, Susanne Wolff, Chi Lu, Martin Chodorow, Lisa Braden-Harder, and Mary Dee Harris. Automated scoring using a hybrid feature identification technique. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 206–210, 1998.

[5] Michael J Curran, Peter Draus, George Maruschock, and T Maier. Student perceptions of project essay grade (peg) software. *Issues in Information Systems*, 14(1):89–98, 2013.

[6] Michael J Curran, Peter Draus, George Maruschock, and Tim Maier. Faculty perceptions of project essay grade (peg) software. *Issues in Information Systems*, 15(1), 2014.

[7] Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407, 1990.

[8] Noura Farra, Swapna Somasundaran, and Jill Burstein. Scoring persuasive essays using opinions and their targets. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*, pages 64–74, 2015.

[9] Harshanthi Ghanta. Automated essay evaluation using natural language processing and machine learning. Master's thesis, 2019.

[10] Shelby J Haberman and Sandip Sinharay. The application of the cumulative logistic regression model to automated essay scoring. *Journal of Educational and Behavioral Statistics*, 35(5):586–602, 2010.

[11] Zixuan Ke and Vincent Ng. Automated essay scoring: A survey of the state of the art. In *IJCAI*, volume 19, pages 6300–6308, 2019.

[12] Beata Beigman Klebanov and Michael Flor. Word association profiles and their use for automated scoring of essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1148–1158, 2013.

[13] Yali Li and Yonghong Yan. An effective automated essay scoring system using support vector regression. In *2012 Fifth International Conference on Intelligent Computation Technology and Automation*, pages 65–68. IEEE, 2012.

[14] Alen Lukic and Victor Acuna. Automated essay scoring. *Rice University*, 2012.

[15] Tim McGee. Taking a spin on the intelligent essay assessor. *Machine scoring of student essays: Truth and consequences*, 7992, 2006.

[16] Huyen Nguyen and Lucio Dery. Neural networks for automated essay grading, 2018.

[17] Ellis B Page. The imminence of... grading essays by computer. *The Phi Delta Kappan*, 47(5):238–243, 1966.

[18] Isaac Persing and Vincent Ng. Modeling thesis clarity in student essays. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 260–269, 2013.

[19] Peter Phandi, Kian Ming A Chai, and Hwee Tou Ng. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, 2015.

[20] Donald E Powers, Jill C Burstein, Martin Chodorow, Mary E Fowles, and Karen Kukich. Stumping e-rater: challenging the validity of automated essay scoring. *ETS Research Report Series*, 2001(1):i–44, 2001.

[21] Thomas Quinlan, Derrick Higgins, and Susanne Wolff. Evaluating the construct-coverage of the e-rater® scoring engine. *ETS Research Report Series*, 2009(1):i–35, 2009.

[22] Lawrence M Rudner and Tahung Liang. Automated essay scoring using bayes' theorem. *The Journal of Technology, Learning and Assessment*, 1(2), 2002.

[23] Kaveh Taghipour and Hwee Tou Ng. A neural approach to automated essay scoring. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 1882–1891, 2016.

[24] Salvatore Valenti, Francesca Neri, and Alessandro Cucchiarelli. An overview of current research on automated essay grading. *Journal of Information Technology Education: Research*, 2:319–330, 2003.

[25] Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. A new dataset and method for automatically grading esol texts. In *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, pages 180–189, 2011.