教育测量与评估双语期刊
CHINESE/ENGLISH
JOURNAL OF EDUCATIONAL MEASUREMENT AND EVALUATION

Volume 4
Issue 3 *Upcoming: Special issue on automated scoring of reading and writing tasks*
Article 1

2023

Comparing Performance of Feature Extraction Methods and Machine Learning Models in Essay Scoring

Lihua Yao
Northwestern University

Hong Jiao
University of Maryland

Recommended Citation

Yao, Lihua and Jiao, Hong (2023) "Comparing Performance of Feature Extraction Methods and Machine Learning Models in Essay Scoring," *Chinese/English Journal of Educational Measurement and Evaluation | 教育测量与评估双语期刊*: Vol. 4: Iss. 3, Article 1.
DOI: https://doi.org/10.59863/DQIZ8440
Available at: https://www.ce-jeme.org/journal/vol4/iss3/1

Comparing Performance of Feature Extraction Methods and Machine Learning Models in Automatic Essay Scoring

Lihua Yao [a] and Hong Jiao [b]

[a]Northwestern University, Feinberg School of Medicine, Department of Medical Social Sciences
[b]University of Maryland, Department of Human Development and Quantitative Methodology

Abstract
This study used Kaggle data, the ASAP data set, and applied NLP and Bidirectional Encoder Representations from Transformers (BERT) for corpus processing and feature extraction, and applied different machine learning models, both traditional machine-learning classifiers and neural-network-based approaches. Supervised learning models were used for the scoring system, where six out of the eight essay prompts were trained separately and concatenated. Compared with previous study, we found that adding more features such as readability scores using Spacy Textsta improved the prediction results for the essay scoring system. The neural network model, trained on all prompt data and utilizing NLP for corpus processing and feature extraction, performed better than other models with an overall test quadratic weighted kappa (QWK) of 0.9724. It achieved the highest QWK score of 0.859 for prompt 1 and an average QWK of 0.771 across all 6 prompts, making it the best-performing machine learning model that was tested.

Keywords
Automated Scoring;
Feature Extractions;
Natural Language Processing;
BERT

1 Introduction

Writing skills and higher-order thinking skills are often assessed through constructed response items, such as short or long essays, which allow test-takers to demonstrate their ability to articulate their thoughts and ideas in a more complex and nuanced way than can be captured through multiple-choice questions or other types of selected-response items. From an item response theory (IRT) model perspective, constructed response items that have more than two response categories are generally considered to have more information than multiple-choice items (Zwick 1990; Yao & Schwarz, 2006; Reckase, 2009). Traditionally, human raters have been used to score students' constructed responses. However, human rating poses several intrinsic potential problems. Firstly, it is expensive due to the need for subject matter experts to devote extensive time to scoring one response. Secondly, human rating is a judgmental process and is subject to various rater effects, such as rater severity, leniency, centrality, the Halo effect (Thorndike 1920), and other inconsistencies related to fatigue, rater expertise, stereotyping, and other qualitative backgrounds. Wind (2019) suggests that rater effects related to severity, centrality, and misfit may substantially impact student classifications and latent ability estimates, ultimately threatening the fairness of rater-mediated assessments.

Numerous approaches have been suggested in the literature to address rater effects in scoring constructed responses. One common approach is to use multiple raters to score each response, which can increase inter-rater consistency (Wolfe & McVay, 2012). However, achieving high inter-rater reliability is not always feasible. Regression-based methods, such as ordinary least squares (OLS) and weighted least squares (WLS), have also been explored to reduce rating errors (Raymon & Houston, 1990). Many researchers have proposed latent variable models, including the many-FACET Rasch model (Linacre, 1989, 2018), which model rater effects while accounting for the noise variance from raters. Other latent variable

Contact: Lihua Yao. Northwestern University, Feinberg School of Medicine, Department of Medical Social Sciences. Lihua.Yao@northwestern.edu.

models, such as the two-parameter partial credit models within the Bayesian item response theory modeling framework, have been proposed to account for various types of rater effects (e.g., Patz, Junker, Johnson, & Mariano, 2002; Wang & Yao, 2013). However, despite the advantages, these approaches may still be impacted by the regression towards the mean effect, and the subjectivity in human raters' ratings is still context-dependent and cannot be fully accommodated by the models. Additionally, the Bayesian approach to modeling rater effects has shown that the estimated rater scores tend to be regressed towards the mean of the priors.

Page (1966, 1968) conducted pioneering work on the Project Essay Grader (PEG) system, which has since inspired the development of many automated essay scoring systems. In 1998, Educational Testing Service developed the "e-rater" (Attali & Burstein, 2006), while Pearson Knowledge and Technologies developed the Intelligent Essay Assessor (IEA, Zupanc & Bosnic, 2015) in the same year. Other companies that developed automated scoring systems include Vintage Learning with IntelliMetric, CTB with Bookette, Pacific Metrics with CRASE, and the American Institute of Research with AutoScore. Recent advancements in automated essay scoring include the essay scoring engine of Measurement Incorporated, which won the Grand Prize in the Automated Scoring Challenge for the Nation's Report Card (NCES, 2022).

Automated essay scoring systems rely heavily on natural language processing to convert textual data into numerical values that can be analyzed using various machine learning techniques. This process involves extracting a range of features from essays, including length-based features, lexical features, embeddings, word category features, prompt-relevant features, readability features, syntactic features, argumentation features, semantic features, and discourse features, among others (Ke & Ng, 2019). While not all studies incorporate all of these features, effective feature engineering is a crucial element in developing accurate automated essay scoring systems.

In recent years, several researchers (e.g., Ke & Ng, 2019; Zupanc & Bosnic, 2015) have conducted surveys on the state of the art in automated essay scoring. Zupanc and Bosnic (2015) provided an overview of the progress made in automated essay scoring by examining the history, challenges, and 21 existing systems in this field. Ke and Ng (2019) focused on the corpora used for training, the approaches employed (including supervised, weakly supervised, and reinforcement learning), features extracted, and evaluation criteria. Ramesh and Sanampudi (2022) conducted a review of current automated essay scoring systems and categorized them into four modeling approaches: regression-based, classification models, neural networks, and ontology-based. More recent systems, particularly those developed since 2016, have predominantly employed neural networks. Notably, out of the 21 systems reviewed by Zupanc and Bosnic (2015), only two used neural network approaches.

Though different machine learning algorithms such as support vector machine (SVM, e.g., Ke et al., 2019; Persing & Ng, 2013), random forest (RF, Kumar et al., 2019; Mathias & Bhattacharyya, 2018a; b), XGBoost (Salim et al., 2019), Naive Bayes (NB), K-Nearest Neighbors (KNN), and stacking ensemble learning methods have been explored for developing automated essay scoring, the performance of these supervised machine learning models was vulnerable to the statistical features extracted (Ramesh & Sanampudi, 2022). These features are often extracted text features from different sources including text responses, scoring rubrics, writing prompts, and content standards. The types of features and the number of features extracted all impact the accuracy of automated essay scoring system. Automated essay scoring has been researched for over 50 years. At present, it is one of the routine psychometric analysis in operation for many large-scale tests. Despite the prevalence of automated scoring in assessment practice, most automated scoring engines used in large-scale tests remain black boxes due to the intellectual property owned by different testing companies who have put numerous intellectual and financial resources in developing such a system over decades. Zupanc and Bosnic (2016) also noticed that the technical details of automated essay scoring are often missing in research or technical reports as the commercial agencies supported the development of such a system considered it as proprietary information.

1.1 Development of Automated Scorer

The development of automated scorer involves three primary steps: 1) obtaining a training dataset containing student essays, 2) identifying relevant features or predictor variables, and 3) training predictive models using the training data. In this particular study, the authors leveraged recent technological advances to investigate the impact of various factors in each step of the automated essay scoring process. Specifically, for **step 1** (obtaining training data), we explored the

performance of training data *concatenated* and *separately* across different prompts or scoring rubrics. To explore the impact of different corpus processing and feature extraction techniques in **step 2**, we utilized two methods. The first method involved *NLP*, where we went through each step of the process. The second method utilized *BERT* (Devlin et al., 2018), a pre-trained model. We also considered the impact of feature space and dimensions on the scoring system, and included features such as the readability index to enhance the feature space and improve prediction results. Lastly, since the performance of machine learning models is data-dependent, we explored different models in **step 3** (training the models), including hyperplane for separation, ensemble machine learning, and neural networks and deep learning. Different machine learning models in sklearn such as support vector machine (SVM), Logistic Regression (LRG), Decision Tree, K-Nearest Neighbors (KNN), Random Forest (RF), Gradient Boosting(GB), artificial neural networks (ANN) in sklearn, and deep learning such as Feed-forward neural networks (FNN), and Long Short-Term Memory (LSTM) approach in Keras were applied.

1.2 Studies Using the ASAP Dataset

While many researchers have utilized the public *ASAP* dataset from the 2012 Hewlett Foundation Automated Essay Scoring challenge, we also chose to use the Kaggle dataset for our study due to its unique and diverse characteristics. This dataset offers a wider range of essay prompts, scoring rubrics, and score levels, which can provide new and valuable insights into the automated essay scoring process. Below are a few examples of models and results using the ASAP data set from our literature review from as early as year 2012 to 2022. Mahana et al. (2012), with the linear regression with a polynomial basis function, yielded an average QWK of 0.73. Taghipour et al. (2016), with the model of convolution layer before feeding the embeddings to the recurrent Long Short Term Memory units (LSTM) layer (CNN+LSTM), yielded an average QWK of 0.726. Dong, Zhang, and Yang (2017), with LSTM-CNN, yielded an average QWK of 0.764, with highest QWK of 0.822 for prompt 1. Nagaraj et al.(2018), with the combined models of LSTM and Deep Neural Network Model (DNN) and Word2Vec with 100 dimensions, yielded an overall QWK of values 0.9721 for all 8 prompts. Uto and Okano (2020), with integrates handcrafted essay-level features into a DNN-AES model, yielded an average QWK of 0.749. Uto, Xie, and Ueno (2020), with BERT and essay-level features, yielded QWK of 0.85 for prompts 1, and 0.88 for prompts 4 and 5. Eluwa J et al.(2022), with recurrent unit technique, yielded an average QWK of 0.88. Ormerod (2022), with the DeBERTa Large model, a pre-trained trans-former-based language model utilizing disentangled attention and an adversarial training mechanism, yielded an average QWK score of 0.797. Firoozi et al.(2022), with fine-tuned glove on an LSTM, yielded an average QWK score of 0.79. With neural networks, using 6 essay sets, Nguyen and Dery (2017) obtained a QWK score of 0.94, with word vectors of 300 dimensions, using a 2-layer neural network that trains word vectors together with the weights of learning rate $= 0.0001$ and regularization $L2 = 0.0001$.

To improve inter-rater agreement, it is more effective to train the model using all essays from all prompts or writing levels. This approach not only increases the training size and improves prediction accuracy, but also enhances agreement and consistency across all reading and writing levels. Our literature review revealed that two papers (Nguyen & Dery, 2017; Nagaraj et al., 2018) used combined data training from multiple prompts, while most others trained the model separately for each prompt. In this study, we investigated the effects of training the data together versus separately for different levels.

1.3 Features Used for Automated Scoring

For NLP and BERT processing in extracting feature space, we added features of readability scores using Spacy Textsta. There have been several studies that have used the readability index for automatic essay scoring. One example is a study by Chen, Chen, and Lu (2011) that used the Flesch Reading Ease formula to evaluate the readability of essays written by second language learners. The study found that the readability score was positively correlated with the quality of the essays, as assessed by human evaluators. Another example is a study by Attali and Burstein (2006) that used the Flesch-Kincaid Grade Level formula to evaluate the readability of a large corpus of student essays. The study found that the readability score was a useful predictor of essay quality, and that it was positively correlated with measures of text coherence and fluency. There have also been several studies that have used the readability index in combination with other features, such as grammar and style, to develop more sophisticated automatic essay scoring systems. For example,

Burstein, Chodorow, Leacock, and Mark (1999) used a combination of readability features, grammatical features, and word frequency features to develop an automatic essay scoring system that was able to predict human scores with high accuracy.

To summarize, the objective of this study was to examine the elements that impact the automated essay scoring by examining diverse techniques for corpus processing, feature extraction, and machine learning models. The outcomes of this study could assist researchers and developers in creating and enhancing automated essay scoring systems.

The paper follows next by three parts: *Methods* that covers the study design, the data, data corpus processing, and machine learning models and evaluation criteria, *Results*, and *Discussion*.

2 Methods

In this part, we will talk about study design, the data set that was used, and then explain each step in the data processing for NLP and the parameters used in each step, followed by the pre-trained transformer BERT, followed by models for machine learning in training the data, and the evaluation criteria.

2.1 Study Design

This study developed an essay scorer through a four-step process, which involved data pre-processing, feature extraction, model training, and model performance evaluation. To perform corpus processing and feature extraction, two approaches were utilized: NLP and BERT, as described in the work of Devlin et al. (2018). In addition to the features typically produced by NLP and BERT packages, the study identified the readability index as a critical feature for automatic essay scoring.

First, this study explored to train an automated scorer using all essays across multiple prompts. In general, it was expected that the combined training data would increase the training sample size leading to higher predicted score accuracy for score categories with small sample sizes in its original prompt. Training an automatic essay scoring system to handle essay scoring with different levels and score categories together has several advantages, such as reducing workload, improving efficiency, and better utilizing available data by training the system on all the available relationships between the different levels and score categories. This approach leads to a more robust and accurate model, as the system learns from the full range of data. The key to achieving this is having a large and diverse dataset with examples of essays at different levels and scores and training a model that can learn the relationship between the text of the essays and their scores.

The common approach to creating automated essay scoring models using ASAP data is to construct prompt-specific models. Nonetheless, two studies - as observed by Nagaraj et al. (2018) and Nguyen & Dery (2017) - developed prompt-free automated scorers utilizing all essays from 6 or 8 prompts in the ASAP dataset. To assess the influence of training data jointly versus separately for various levels, the study utilized identical feature selection methods and models to both concatenated data for all prompts and data by prompt separately.

2.2 Data Set

In this study, NLP and BERT and different machine learning and deep learning models were applied to the ASAP dataset. The dataset includes 8 essay sets, each generated from a single prompt. Prompt 1 in the dataset differs from the other prompts, as it asks students to write an essay in response to a specific reading passage and the scoring rubric evaluates both the quality of the writing and the student's ability to analyze and respond to the passage. The score range for prompt 1 is 2-12. The remaining prompts in the dataset focus more on evaluating the student's ability to write effectively and clearly. These prompts are scored on a 1-6 scale, which is a common measure used in writing assessments to provide a measure of writing quality. Every essay in the dataset was evaluated by two independent human raters, and the final score assigned to the essay was obtained by adding the scores given by each rater. The resulting score is stored in the dataset as domain1 score for all essays, and for the purpose of this study, it will be referred to as the true score. Although essays from prompt 2 have an additional human rated score named domain2 score, we did not consider it in this study because other prompts did not have similar scores. The dataset contains essays written by students in grades 7 to 10, and the average length of each essay ranges from 150 to 550 words.

Figure 1 and Figure 2 display the relationship between the essay scores, as indicated by domain1 score, and the average and total word length for the 8 essay sets in the dataset, with the last one representing scores for domain2 score. Notably, there is a strong positive correlation between essay scores and total word length, which is expected given that longer essays may demonstrate a more thorough and thoughtful analysis. Additionally, there are some weaker correlations with average word length. These observations inform our feature selection process, as we consider including both total and average word length as features in our model training.

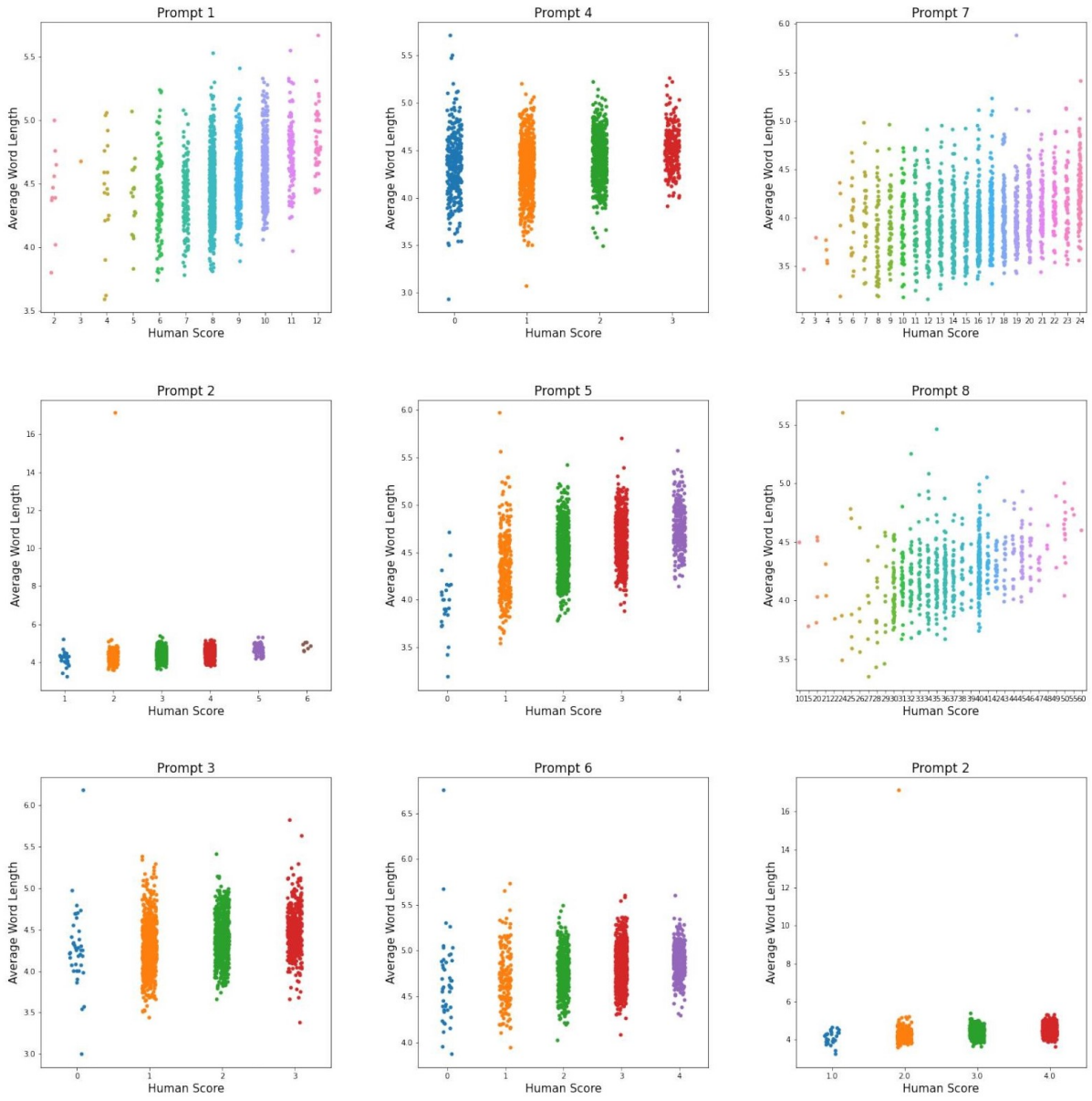

Figure 1: Average Word Length Versus Scores for the Eight Essay Set

For prompts 7 and 8, the maximum score points are 24 and 60 respectively. This means that in the training process, we will need to classify essays in these prompts into one of the 24 or 60 categories. Considering the machine training time and also a concern of the sample sizes for each of the score points or classes for these two prompts, we only conducted analysis for prompts 1-6 in this paper. To account for the differences between prompt 1 and the other prompts, three different types of datasets were used for training the automated essay scoring models. The first dataset included

all prompts 1-6 combined (referred to as ALL). The second dataset included each prompt trained separately (referred to as Sep), and the third dataset included prompts 2-6 trained together and prompt 1 trained separately. This approach was taken to evaluate the impact of combining prompts in the training process and to assess the effectiveness of training prompt-specific models. In total, the dataset used for training consisted of 10,684 essays.

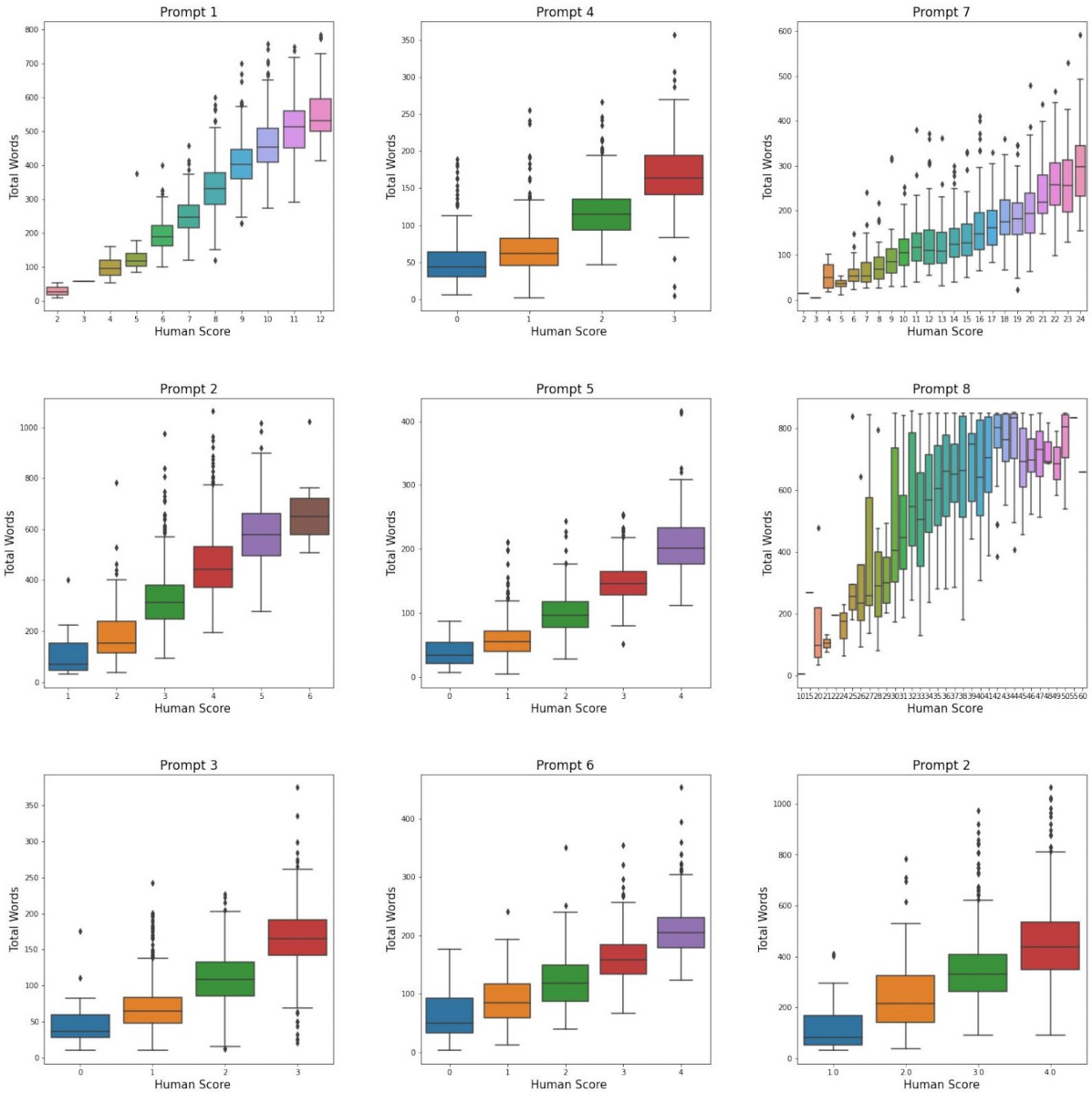

Figure 2: Total Word Length Versus Scores for the Eight Essay Set

The score distribution for each of the 6 prompts in the train and validation data sets from the Kaggle website is presented in Table 1. The score distribution is unevenly distributed, which could impact the results of the machine learning model. One potential solution to address this issue is to use SMOTE (Chawla, et al., 2002), a technique that oversamples the minority classes. The validation data set, which had a size of 3,544, was utilized as the test data set and processed in the same way as the training data set. For reference, an example essay can be found in the Appendix.

Table 1: Train Data and Validation Data Summary

	Score	Train		Validation	
Prompt1		Count	Percentage	Count	Percentage
	2	10	1	2	0
	3	1	0	0	0
	4	17	1	2	0
	5	17	1	5	1
	6	110	6	27	5
	7	135	8	60	10
	8	687	39	195	33
	9	334	19	157	27
	10	316	18	105	18
	11	109	6	31	5
	12	47	3	5	1
Total		1783		589	0
Prompt2	1	24	1	3	0
	2	153	8	46	8
	3	763	42	279	46
	4	778	43	253	42
	5	75	4	19	3
	6	7	0	0	0
Total		1800		600	
Prompt3	0	39	2	5	1
	1	607	35	173	30
	2	657	38	244	43
	3	423	25	146	26
Total		1726		568	
Prompt4	0	311	18	67	11
	1	636	36	229	39
	2	570	32	213	36
	3	253	14	77	13
Total		1770		586	
Prompt5	0	24	1	3	0
	1	302	17	80	13
	2	649	36	233	39
	3	572	32	182	30
	4	258	14	103	17
Total		1805		601	
Prompt 6	0	44	2	4	1
	1	167	9	42	7
	2	405	22	141	24
	3	817	45	327	55
	4	258	14	103	17
Total		180		600	
All Total		10684		3544	

2.3 The Feature Space Using NLP

In general, NLP has two main phases in application: first is the data processing, and second is the algorithms. During the data preparation, each essay text corpus was preprocessed: terms within the corpus are normalized in the following steps: a) Convert to lowercase or upper case; b) Stopword removal; c) Stemming and lemmatization; d)Feature extractions and tokenization. The unstructured text is transformed into a structured form; the feature space is created for model training.

For normalizing the corpus/essays, the following steps were applied: 1. convert to lowercase 2. stop words processing: nltk.corpus import stopwords; 3. *Lemmatize* use *WordNetLemmatizer().lemmatize*; 4. *Stemming: Snowball Stemmer("english")*; 5. realize Bigram, trigram from *gensim.models*; 6. part-of-speech tagging: *allowed_postags* = [*'NOUN'*, *'ADJ'*, *'VERB'*, *'ADV'*]. Below are explanations of the computer terms or languages.

Step 1: Texts are converted into a lower or upper case.

Step 2: *Stopwords* "the", "of", "to" are common but do not contain meanings. NLK module import stopwords is used to view and filter stop words. For each essay, check each word, and if the word has length smaller than 2 or is a STOPWORDS, then remove it.

Step 3: *Lemmatization* is the process of replacing a word with its root or head word called lemma. The aim is to reduce inflectional forms to a common base form (Khyani et al., 2021). For example "am," "are", "is", "was", "were", would all be treated the same as 'be'. The function, WordNetLemmatizer() was used to lemmatize words.

Step 4: *Stemming* is the process of reducing a word to its word stem (i.e. flying to fly). By combining lemmatization and stemming, it is possible to obtain the accuracy of lemmatization while retaining the speed and simplicity of stemming. Combining lemmatization and stemming can lead to improved performance on certain tasks, such as text classification or information retrieval, by reducing words to their root form while preserving their meaning.

Step 5: N-grams analyses are often used to see which words often show up together. Combinations of two words or three words are referred to, i.e., Bigrams/Trigrams. After Step 3, each essay is converted to a list of words. Bigrams/Trigrams was applied and the list now not only contain a single meaningful word, but also contains two or three meaningful words.

Step 6: Only the nouns, adjectives, verbs, adverbs are included to improve topic interpretability.

The next step after the normalization of corpus/ essays is to create feature space for the model training. The most commonly used method in creating features is "Document Term Matrix," (*DTM*), in which documents are treated as rows (i.e., observations) and the items of text are treated as columns (i.e., features or terms). This kind of structure is very much similar to the data structure we used to apply item response theory to the testing data, where each row represents each test taker and each column represents each item in the test. Instead of being the response of the examiner to the item in each cell, the number in the cell is the number of appearances of the term in the input document. That is, the element $TF(ij)$ for the *i*th row, *j*th column in this matrix is the frequency of the appearance of the *j*th term in the *i*th document. Another popular method is Term Frequency-Inverse Document Frequency values ($TF-IDF$) (Silge & Robinson, 2018). $TF-IDF$ generates values by first calculating a Term Frequency (TF) for the number of times each term occurs within a document. Next, each TF value is multiplied by an Inverse Document Frequency (IDF) value that assigns higher weight for elements that are rarely found across documents. The Document Frequency is computed by the number of documents that the term appears in divided by the total number of documents. Therefore, if a term is infrequent, and only appears, for example in one document, then IDF will be higher.

After normalizing the text corpus, *CountVectorizer* $TF-IDF$ was applied to converted the tockenized words into vectors of length 1,000. That is, for each essay, there was 1,000 columns after the tokenization. The number 1,000 was selected because it exceeded both the number of features in Huyen Nguyen and Lucio Dery's 2017 study, as well as the number of features in BERT, which was 768. This decision was based on the authors' previous experience and study.

Creating essays is a complex process, therefore, readability index, similar to summary statistics, will increase the performance of the prediction if they are added to the feature space (Zhou & Jiao 2022). *SpacyTextstat* is a python package that calculates statistics from text to determine readability, complexity, and grade level of a particular corpus.

Readability scores for each essay were created, and normalized, and we adopted and computed the following index: *'FleschScore','smog_index','difficult_words','avg_sentence_length','spache_readability', 'dale_chall_readability_score'*. For demonstration purposes, we will show how *'FleschScore'* and *'dale_chall_readability_score'* were computed in the package.

'FleshScore' is called the Flesch reading-ease test. A higher *'FleschScore'* score indicates that the materials are easier to read, while a lower number indicates that the passages are more difficult to read. For example, a *'FleschScore'* of 100-90, indicates a school level of 5th, and a score of 80-70 indicates a school level of 7th. The formula for the Flesch reading-ease score (FRES) test is:

$$206.853 - 1.015 \times \frac{totalwords}{totalsentences} - 84.6 \times \frac{totalsyllables}{totalwords} \tag{1}$$

The formula for calculating the raw score of *'dale_chall_readability_score'* is given below:

$$0.1579 \times \frac{difficultwords}{words} \times 100 + 0.0496 \times \frac{words}{sentences} \tag{2}$$

In the computation, we uses a list of 3,000 words that groups of fourth-grade American students could reliably understand. Here the difficult words are any words that are not on the list. If the percentage of difficult words is above 5%, then add 3.6365 to the raw score to get the adjusted score, otherwise the adjusted score is equal to the raw score. A *'dale chall readability score'* of 5.0-5.9 indicates that the corpus can be understood easily by an average of 5th or 6th-grade students, and a score of 6.0–6.9 indicates that the passages is easily understood by an average 7th or 8th-grade student.

The six readability indexes computed from package *SpacyTextstat*, together with average word length, average essay length, and essay prompt numbers were added into the feature space for the training process. When training the machine learning model using data from all 6 essay prompts, the prompts were treated as a single feature, and their numerical values were normalized to account for any variation in the order of prompts. This feature was combined with the 1,000 columns generated from the tokenization process, resulting in a total of 1,009 features or columns created for each essay. The definitions and indices for these features can be found in Table 2. On the other hand, when training the model using data from each prompt separately, there were 1,008 features in total, which did not include the feature for the essay prompt. All of the pre-processing steps for the text data, including normalization, CountVectorizer, and TF-IDF, were applied to both the training and test data sets. This ensured that the text data in the test set was processed in the same way as the training data, and thus allowed for a fair evaluation of the machine learning model's performance on unseen data. It's important to apply the same pre-processing steps consistently to all data sets to ensure accurate and reliable results.

Table 2: Features of Readability and Their Definitions

Spacy Textstat	Explannation of the statistic index
FleschScore	How difficult a passage in English is to understand
smog_index	How many years of education the average person to have to understand a text
difficult_words	Level of difficulty of the word
avg_sentence_length	Average sentence length
spache_readability	Readability test for writing in English works best on texts that are for children up to fourth grade.
dale_chall_readability_score	Difficulty level of understanding of the passage
avg_word_length	Average word length
total_words	Total number of words
avg_sentence_length	Average sentence length

2.4 Feature Space Using BERT Data Processing

BERT is a powerful pre-trained transformer-based machine learning model developed by Google (Devlin et al., 2018) for natural language processing tasks. It was pre-trained on a large corpus of unlabeled text, including Wikipedia and book corpus, with over 3.3 billion words. BERT takes into account the context of a word, resulting in 768-dimensional embeddings. In this study, BERT was used to extract features from the essays, along with readability indices, resulting in a feature space of 777 dimensions. Previous research by Rodrigez, Jafari, and Ormerod (2020) also used BERT on the same ASAP dataset and achieved an average QWK of 0.752, with their highest average QWK of 0.76 using a combination of LSTM and CNN models.

2.5 Machine Learning Models

After the pre-processing of either NLP or BERT, a feature space is obtained with 1,009 dimensions for NLP and 777 dimensions for BERT. Machine learning and deep learning models are trained using the training data, and hyperparameters are optimized using GridSearchCV from the sklearn package with a default 5-fold cross-validation. For each fold, a different subset of the data is used as validation data, and the remaining data is used for training. The model is evaluated on each fold, and the results are aggregated to select the best combination of hyperparameters. The sample size for each fold ranges from 340-360, which is a reasonable size. A higher cross-validation, which means a larger training size, generally leads to better results but also requires more computation time.

Since the quality and quantity of training data greatly impact the accuracy and performance of predictive models, three types of models are explored for the ASAP dataset: hyperplane separation, ensemble, and deep learning models. The modeling techniques applied include SVM, LRG, Tree, KNN for hyperplane separation models, RF, GB for assembling models, and ANN, FNN, LSTM for deep learning models. These models are implemented using packages such as sklearn and Keras.

- **Support vector machine (SVM)**: a supervised machine learning method used for classification and regression. The parameters used is 'rbf', Radial Basis Function for the kernal and four C values for the strength of the regulation from 1, 10, 50, 100.
- **Logistic regression (LRG)**: a classification algorithm. L2(Ridge regularization) penalty term, default solver 'lbfgs'[1] , and 20 regularization strength from 0.01 to 5 is implemented.
- **Decision tree (Tree)**: a supervised learning approach for classification and prediction. Seven numbers from 3 to 30 were chosen for *max_depth*.
- **The k-nearest nerghbors (KNN)**: a very simple and easy to understand algorithm; we used 10 values of neighbors and 5 cross validation.
- **Random forest (RF)**: a supervised ensemble learning method for classification and regression. The ensemble is through bootstrapping (Breiman, 1996). Ten *n_estimators* from 10 to 500 were used.
- **Gradient Boosting (GB)**: a class of algorithms for classification and prediction. The ensemble is Boosting (Friedman 2001) method that is constructed sequentially; it is different from RF. We used *GradientBoostingClassifier* from *sklearn*. Three *n_estimators* values 10, 30, and 50 were used.
- **ANN**: it is neural network that mimics the way nerve cells work in the human brain. We used *MLPClassifier* from *sklearn*. Cross run of two layer sizes([(150,100,50), (120,80,40)]), two solver (['sgd', 'adam'])[2], two activation (['tanh', 'relu']), two alpha ([0.01,0.3, 1]), and two learning rate (['constant','adaptive']) were conducted. Both tanh and relu are non-linear activation functions[3].
- **FNN**: an artificial neural network, named feed-forward Neural Network. We used *Sequential* imported from *keras.models* and five layers with Dense 2,000 and activation='relu', which stands for rectified linear activation unit. The last layer used 'sigmoid' as the activation. The difference between this multi-class classification model and the other models is that the output value, which is the scores for each essay, need to be converted or reshaped

[1] algorithm is to minimized the negative log-likelihood loss function.

[2] sgd is a simple and efficient optimization algorithm and adam is more advanced and robust and adaptive.

[3] $relu(x) = max(0,x), tanh = (e^x - e^{-x})/(e^x + e^{-x})$

into a matrix format of binary data; it is called one-hot encoding. For instance, if the essay score is 8, then the column 8 of the matrix is 1, and the other columns have a value of 0. Since the maximum score point for all essays in prompts 1-6 is 12, the matrix has a total of 13 columns.

- **LSTM**: recurrent Long Short Term Memory. This is a deep learning model. For this model, the input data needs to be reshaped from 2D to 3D. Three *epochs* and three *batch_size* were applied[4].

The four hyperplane models, two assembling models, and three deep learning models are widely used and can be easily implemented using existing packages, allowing readers to replicate the process. Python code for these models can be found on the author's GitHub website[5].

2.6 Evaluation Criteria

QWK provides a more robust and nuanced measure of inter-rater agreement; it is computed using the following formula: LetO_{ij} and E_{ij} be the observed and expected number of essays that have a rating of i (actual) and received a predicted value j. This E is calculated as the outer product between the actual rating's histogram vector of ratings and the predicted rating's histogram vector of ratings, normalized such that E and O have the same sum.

$$\omega_{ij} = \frac{i-j}{N-1} \tag{3}$$

$$QWK = 1 - \frac{\sum_{ij} \omega_{ij} Q_{ij}}{\sum_{ij} \omega_{ij} E_{ij}} \tag{4}$$

We also evaluated *precision*, *recall* or *sensitivity*, *accuracy*, *F-score*, and *specificity*. These metrics are typically calculated based on a confusion matrix as shown in Table 3; they are computed as below.

$$Precision = \frac{TP}{FP+TP} \tag{5}$$

$$Recall = \frac{TP}{FN+TP} \tag{6}$$

$$Accuracy = \frac{TP+TN}{TP+FN+TN+FP} \tag{7}$$

$$Specificity = \frac{TN}{TN+FP} \tag{8}$$

$$F = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{9}$$

Table 3: Confusion Matrix

True	Predicted	
	0	1
0	TN	FP
1	FN	TP

The accuracy of the model on the test data, which is also known as validation data on the Kaggle website, is calculated

[4]FNN is a feedforward network that processes input data in a single pass, while LSTM is a recurrent network that processes input data sequentially, allowing it to capture temporal dependencies in the data. FNNs are generally used for tasks that do not involve sequential data, while LSTMs are used for tasks that require modeling temporal dependencies, such as natural language processing, speech recognition, and time series forecasting.

[5]https://github.com/yaolihua081

to determine the test score. The evaluation output includes the train score, test score, and predicted score for each essay, encompassing both the train and test data. The criteria were calculated using functions from the sklearn library.

3 Results

To evaluate the performance of the models trained using all prompts data and training each prompt separately, we computed the overall QWK and QWK for each prompt and compared the results. Table 4 provides a summary of the train score, test score, train data QWK, and test data QWK for both NLP and BERT feature extraction methods using the training data that contains all six prompts. Overall, NLP outperformed BERT, and the ANN model achieved the highest score.

Table 4: Scores and QWK for ALL Models Trained on Data with Six Prompts

Model	*Train_Score*	*Test_score*	*Train_QWK*	*Test_QWK*
LRG	0.674	0.658	0.963	0.966
KNN	0.639	0.638	0.959	0.962
SVM	0.778	0.672	0.977	0.969
Tree	0.719	0.642	0.969	0.967
RF	1	0.645	1	0.961
GB	0.64	0.652	0.96	0.963
ANN	0.69	0.682	0.97	0.972
FNN	0.73	0.667	0.975	0.97
LSTM	0.668	0.634	0.966	0.966
BERT-SVM	0.923	0.608	0.992	0.962
BERT-LRG	0.704	0.629	0.968	0.96
BERT-Tree	0.577	0.619	0.951	0.964
BERT-KNN	0.566	0.468	0.934	0.926
BERT-RF	0.999	0.6	0.9999	0.945
BERT-GB	0.777	0.669	0.977	0.955
BERT-ANN	0.762	0.65	0.975	0.967
BERT-FNN	0.853	0.598	0.982	0.957

Table 5 presents the confusion matrix for each of the six prompts for the test data when training all six prompts together. We observed that for prompt 1, the predicted scores were 2 and 6-12, with score points 3, 4, and 5 missing. This result was expected since the validation data had no score point of 3, and the training data had only one case of score point 3 and only 17 cases of score points 4 and 5.

Table 6 shows the confusion matrix for the training data for prompt 1 when using the ANN model to train all prompts together. It is observed that score points 3, 4, and 5 are missing from the prediction, and most of the scores of 3, 4, and 5 were predicted with values of 6.

Table 7 presents the confusion matrix for each of the 6 prompts on the test data obtained from training each prompt separately. The analysis revealed that for prompt 1, some score points (3 and 5) were still missing in the prediction, and in two cases with score point 4, values of 6 were predicted. These findings are consistent with the results obtained from training on all 6 prompts together.

The results for each of the 6 prompts for training using data from all prompts and each prompt separately are shown in Figure 3 and Figure 4, respectively. These figures provide a visual representation of the relative performance of each model and method.

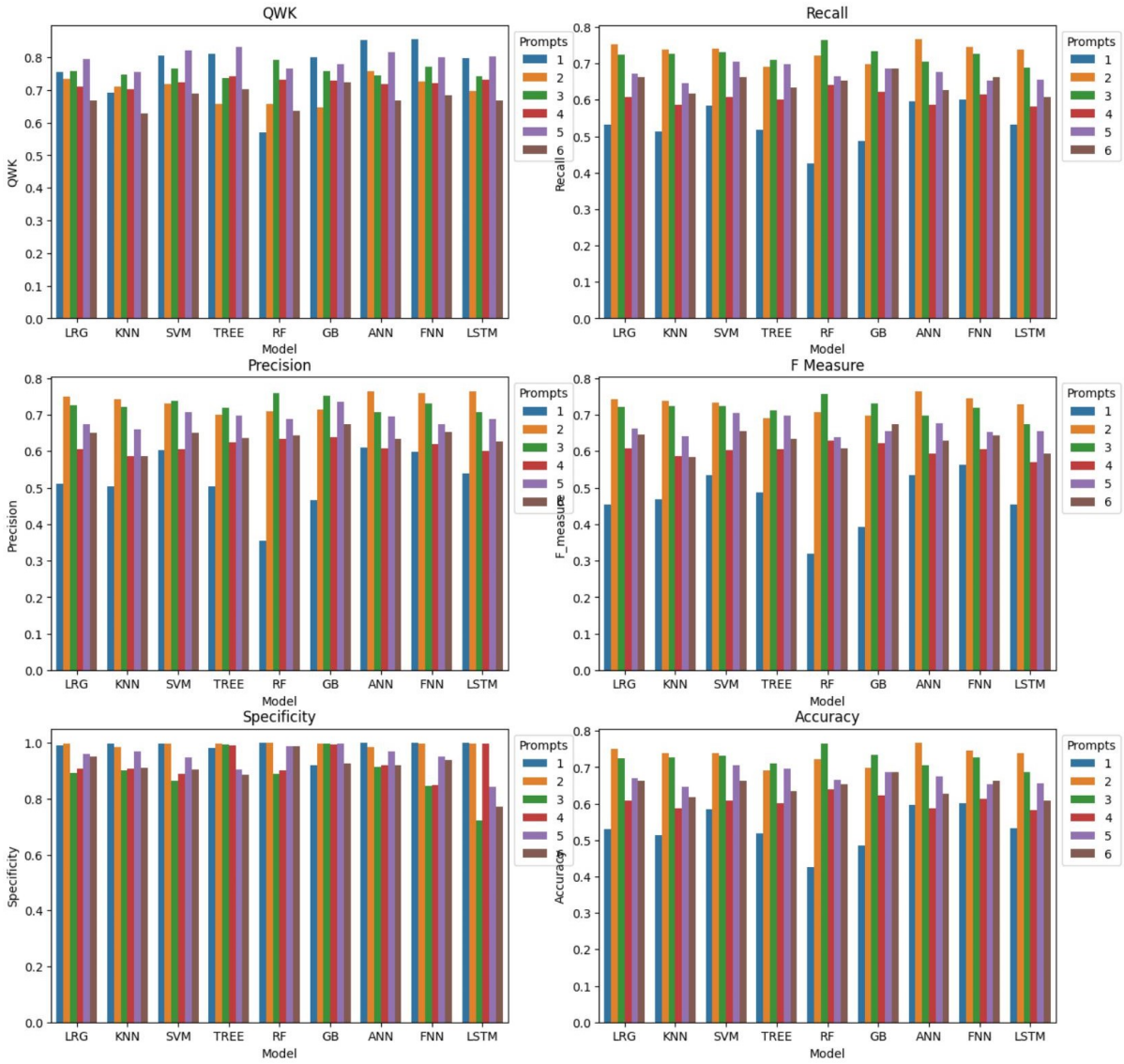

Figure 3: QWK, Recall, Accuracy, F Measure, Specificity, and Accuracy for Each Prompt by Training All Data Together

The results obtained from training the best model on all prompts, prompts 2-6, and each prompt separately are presented in Table 8. The findings indicate that, for prompts 1, 2, 3, and 6, the performance of the model trained on all data together is superior to that of the model trained on each prompt separately. However, for prompts 4 and 5, the model trained on each prompt separately yields better results than the model trained on all data together. The best QWK for prompt 1-6 are 0.859(Sep), 0.759(ALL), 0.796(ALL), 0.779(Sep), 0.845(Sep), and 0.724(ALL), respectively. The ANN model trained on all data has an average QWK of 0.771 across all 6 prompts, while the ANN model trained on separate data has an average QWK of 0.761.

It is worth noting that in addition to the QWK, precision, accuracy, F-measure, and specificity were also computed. However, it is important to keep in mind that the evaluation criteria may vary depending on the specific purpose. For instance, in the context of essay scoring, greater emphasis is placed on the QWK metric.

The overall QWK scores for all prompts trained on prompts level data were computed for nine different models, namely LRG, KNN, SVM, Tree, RF, GB, ANN, FNN, and LSTM. The QWK scores for these models were 0.971, 0.971, 0.97, 0.97, 0.969, 0.969, 0.97, 0.96, and 0.951, respectively. It should be noted that training the model on all data together

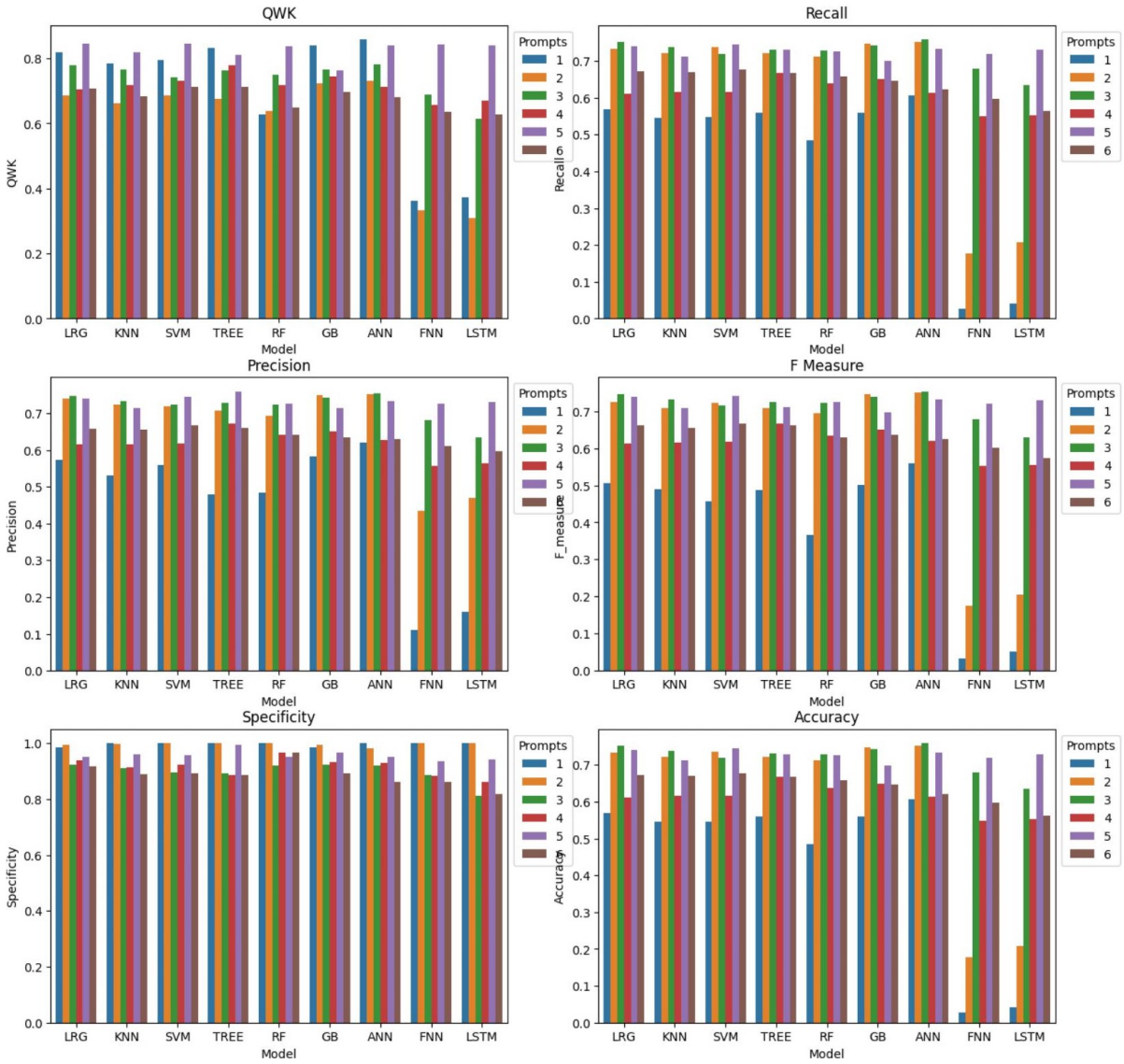

Figure 4: QWK, Recall, Accuracy, F Measure, Specificity, and Accuracy for Each Prompt by Training Each Prompt Separately

or separately may not result in significant differences in QWK scores. Table 8 shows that there are more noticeable differences in other evaluation metrics for some prompt. The detailed classification differences can be found in Table 4 and Table 6.

Please note that "micro averaging" was used in the computation in Table 8. The precision and recall are always the same when using the "micro averaging" scheme; it does not distinguish between different classes. "Macro averaging" and "Weighted averaging" can be applied which might yield slightly different values; "weighted averaging" was used for those values in Figure 3 and Figure 4.

Table 5: Confusion Matrix Table for ANN Across All 6 Prompts Using Test Data from Training All 6 Prompts

Prompt1	2	6	7	8	9	10	11	12
2	2	0	0	0	0	0	0	0
4	0	2	0	0	0	0	0	0
5	0	5	0	0	0	0	0	0
6	0	24	1	2	0	0	0	0
7	0	16	1	43	0	0	0	0
8	0	2	0	185	5	3	0	0
9	0	0	0	75	46	36	0	0
10	0	0	0	2	15	87	1	0
11	0	0	0	0	2	29	0	0
12	0	0	0	0	0	3	1	1
Prompt2	1	2	3	4	5			
1	3	0	0	0	0			
2	2	35	9	0	0			
3	0	2	220	57	0			
4	0	0	48	204	1			
5	0	0	0	14	5			
Prompt3	1	2	3					
0	5	0	0					
1	149	24	0					
2	52	158	34					
3	2	34	110					
Prompt4	0	1	2	3	4			
0	31	35	1	0	0			
1	37	151	40	1	0			
2	3	56	119	35	0			
3	0	1	22	52	2			
Prompt5	0	1	2	3	4			
0	1	2	0	0	0			
1	2	52	25	1	0			
2	0	21	186	26	0			
3	0	0	39	128	15			
4	0	0	0	31	72			
Prompt6	0	1	2	3	4			
0	1	2	1	0	0			
1	2	14	22	4	0			
2	2	12	68	57	2			
3	0	1	29	253	44			
4	0	0	2	25	59			

Table 6: Confusion Matrix Table for Prompt 1 using ANN on Training Data for All 6 Prompts

Prompt 1	2	6	7	8	9	10	11	12
2	10	0	0	0	0	0	0	0
3	1	0	0	0	0	0	0	0
4	2	15	0	0	0	0	0	0
5	1	15	0	1	0	0	0	0
6	1	75	0	34	0	0	0	0
7	0	23	6	106	0	0	0	0
8	0	11	0	600	48	28	0	0
9	0	0	0	169	88	77	0	0
10	0	0	0	48	55	210	3	0
11	0	0	0	4	5	96	4	0
12	0	0	0	0	0	41	5	1

Table 7: Confusion Matrix Table for ANN Across All 6 Prompts Using Test Data from Training Each Prompt Separately

Prompt1	2	4	6	7	8	9	10	11	12
2	1	1	0	0	0	0	0	0	0
4	0	0	2	0	0	0	0	0	0
5	0	0	5	0	0	0	0	0	0
6	0	0	24	2	1	0	0	0	0
7	0	0	18	17	25	0	0	0	0
8	0	0	1	2	179	11	2	0	0
9	0	0	0	0	74	43	39	1	0
10	0	0	0	0	4	13	86	2	0
11	0	0	0	0	0	2	23	6	0
12	0	0	0	0	0	0	2	2	1
Prompt2	1	2	3	4	5				
1	2	1	0	0	0				
2	0	33	13	0	0				
3	0	2	218	59	0				
4	0	0	54	191	8				
5	0	0	0	12	7				
Prompt3	1	2	3						
0	5	0	0						
1	150	22	1						
2	46	171	27						
3	2	34	110						
Prompt4	0	1	2	3					
0	35	31	1	0					
1	49	136	44	0					
2	10	41	139	23					
3	0	1	26	50					
Prompt5	1	2	3	4					
0	3	0	0	0					
1	62	18	0	0					
2	25	186	21	1					
3	0	31	132	19					
4	0	0	42	61					
Prompt6	0	1	2	3	4				
0	2	1	1	0	0				
1	3	21	15	3	0				
2	0	23	66	49	3				
3	0	1	48	230	48				
4	0	0	5	27	54				

Table 8: Best Results by Prompt for All Models

Prompt	Model	Recall	Accuracy	F Measure	Specificity	QWK	Data Trained
1	FNN	0.601	0.601	0.601	1	0.856	ALL
1	ANN	0.606	0.606	0.606	1	0.859	Sep
2	ANN	0.778	0.778	0.778	0.998	0.759	ALL
2	LRG	0.747	0.747	0.747	0.991	0.738	Prompt 2-6
2	ANN	0.752	0.752	0.752	0.982	0.73	Sep
3	RF	0.764	0.764	0.764	0.89	0.793	ALL
3	SVM	0.762	0.762	0.762	0.906	0.796	Prompt 2-6
3	ANN	0.759	0.759	0.759	0.92	0.783	Sep
4	TREE	0.601	0.601	0.601	0.992	0.741	ALL
4	GB	0.648	0.648	0.648	0.997	0.754	Prompt 2-6
4	TREE	0.667	0.667	0.667	0.886	0.779	Sep
5	ANN	0.73	0.73	0.73	0.961	0.839	ALL
5	GB	0.72	0.72	0.72	0.998	0.838	Prompt 2-6
5	SVM	0.745	0.745	0.745	0.956	0.845	Sep
6	GB	0.687	0.687	0.687	0.926	0.724	ALL
6	GB	0.688	0.688	0.688	0.926	0.704	Prompt 2-6
6	SVM	0.677	0.677	0.677	0.893	0.714	Sep

4 Discussion

Different data processing and different machine-learning models were applied to the ASAP training data for essay scoring. Over the last decade, many papers investigated models for automatic essay scoring using ASAP data set. Many studies have trained their automated essay scoring models separately for each of the eight prompts in the ASAP dataset. However, when it comes to inter-rater agreement, it may be more beneficial to train the model using all essays from all prompts or all writing levels. This approach increases the training size, leading to more accurate predictions, and enhances the agreement and consistency across all reading and writing levels. Additionally, incorporating larger and diverse datasets into the training process helps the system learn the relationships between features and scores more effectively. Upon comparing Tables 5 and 7, it becomes evident that training the model using all data together resulted in more accurately predicted scores.

In our literature review, we found two papers that trained model using all prompts. Using data for 6 prompts, Huyen and Lucio (2017) tested different dimensions of essay vectors, varying from 50, 100, 200, and 300 on a 2-layer neural network. Compared to the model trained with dimension 200, with a learning rate of 0.002 and a regularization rate of 0.0001, the model trained with dimension 300 together with a learning rate of 0.0001 and a regularization rate of 0.0001 optimized Kappa from 0.916 to 0.944. Using all 8 prompts, Nagaraj et al.(2018), with the combined models of LSTM and Deep Neural Network Model (DNN) and Word2Vec with 100 dimensions, yielded an overall QWK of values 0.972. In our study, we used data from 6 prompts out of concern of training time using our regular computer. To investigate the factors that affect the results, we varied the corpus processing and machine-learning models. Overall, training data together, the ANN model demonstrated the best performance among all models applied to the data, with an overall test quadratic weighted kappa of 0.972; our overall QWK is higher or comparable with others that using the combined data.

To compare our results with previous studies using the same ASAP dataset but trained separately, we calculated the QWK for each essay prompt. Our best QWK is 0.859 for prompt 1 and an average of 0.771 for all 6 prompts for ANN. For QWK in prompt level, our QWK is higher or comparable to most of the papers we found. For example, Firoozi et

al.(2022), with fine-tuned glove on an LSTM, yielded an average QWK score of 0.79, with QWK of 0.832, 0.713, 0.699, 0.835, 0.826, and 0.834 for prompt 1-6, respectively. In this study, the QWK for each of the six prompts had best values of 0.859, 0.759, 0.796, 0.779, 0.839, 0.724, respectively. Because of concern of computing time with all different models, we only applied a few set of parameters for each model; fine turning parameters of each model, especially our wining model ANN would possibly increase the accuracy and QWK more. For cross validation GridSearchCV , cv = 5 is used; higher values would improve the performance, but computer time is higher. The calculation of QWK is dependent on the scale of the scores used, meaning that the weights assigned to different levels of agreement may vary based on the number and distribution of categories in the scale. Therefore, when comparing QWKs between different raters or judges or studies, it is crucial to ensure that the scales used are consistent to enable meaningful comparisons.

There are several potential factors that could cause misalignment between human scores and predicted scores in automatic scoring systems. For example, the data quality will likely have an impact. The complexity of the machine learning model can also affect its performance. If the model is too simple, it may not be able to capture the complex relationships between the input features and the target scores. On the other hand, if the model is too complex, it may overfit to the training data and perform poorly on new, unseen data. The human bias in the scoring can also be reflected in the annotated data used to train the machine learning model. The students' writing skills are multi trained; no single feature can do it all. In applying NLP, existing packages from gensim and nltk organized the unstructured essay or corpus into clean and structured data. Different readability scores were computed to enhance the structure to access more dimensions of the writing skill in the essay. Because of the complexity of the essay, different machine learning algorithms may function differently. More research looking into factors such as the number of features/dimensions, applying different N-grams in corpus processing, adding other readability index, data argumentation using Blending Ensemble Learning (Zhou & Jiao 2022) that affect the prediction is warranted. We used 1,000 as the maxfeatures in NLP feature selection using $TF - IDF$. This 1,000 feature dimensions plus other features such as the readability and essay prompts did help in improving the scores. The use of the readability index for automatic essay scoring has shown promise as a tool for evaluating the quality and fluency of written text. In general, more feature space require more sample size. Further investigation into the effect of the number of features and the effect of the readability index should be conducted. In general, to reduce the misalignment or the false positive or false negative errors in automatic scoring systems, one needs to experiment with different models, fine turn hyperparameters, make adjustment of each of the factors mentioned above or even make adjustment of the training data if it is imbalanced.

The study only used essay prompts 1-6. We are not sure about the effect of adding other essay prompts. For future study, simulation with varying training sizes and score points will be conducted to examine the effect of training size. For less score points, the number of classes to be classified into are less, therefore, the accuracy is better because of larger size of training data for each class. Andersen et al. (2022) scored short responses with two levels of scores. Using Twitter data, Yao et al.(2020) demonstrated three levels of scoring for short responses or sentences. It is expected that applying NLP for short response answers with less possible score points is promising. Short response answers in an assessment, especially with more than two score levels, modeled by two-parameter partial credit model or graded response model have shown to have much more information than multiple choice items that had scores 0 or 1, therefore, scores for even short answers will greatly increase the precision of latent estimates or shorten the test length (Yao & Schwarz, 2006). Short answer or long essay, automatically scoring is promising in the future assessment in different fields such as in testing or medical fields

BERT, a pre-trained transformer was applied for data processing before the training. There were 768 dimensions after BERT transformer applied to the essays. With the readability index, the feature space were of dimension 777. The results from this data BERT processing was not better than routine NLP data processing. The performance of machine learning or deep learning models and the impact of how the data is processed is data driven. Each model is not fine turned because of time issue. Therefore, the relative performance of models may not be as expected. Conclusions from this study is not representative of the general performance of those models.

Several new transformer models, including GPT-2, GPT-3, T5, and RoBERTa, have been developed and can be used for automatic essay scoring. A recent study by Zhang et al. (2021) found that a pre-trained GPT-2 model was able to achieve high accuracy and reliability in scoring essays written by Chinese students on a standardized test of English

writing proficiency, outperforming other state-of-the-art AES systems. These promising results suggest that transformer models have the potential to significantly advance the field of automatic essay scoring.

In the future, the authors plan to extend their research into the health care field, where natural language processing has shown promise in aiding health care professionals in categorizing patient conditions and making appropriate intervention choices. Automated scoring of patient notes or social media posts could potentially improve clinical decision making, streamline medical evaluations and diagnoses, and aid in training. Patient notes that express positive or negative feelings could even contribute to the diagnosis of certain diseases such as dementia. By automating technical tasks, physicians could have more time to focus on patient care, leading to an overall improvement in patient experience.

References

[1] Albawi, S., Mohammed, T. A., & Al-Zawi, S. (2017). *Understanding of a convolutional neural network.* 2017 International Conference on Engineering and Technology (ICET), 1–6. https://doi.org/10.1109/ICEngTechnol.2017.8308186.

[2] Andersen, N., & Fabian Z. (2021) ShinyReCoR: A Shiny Application for Automatically Coding Text Responses Using R. *Psych* 3, 422–46. https://doi.org/10.3390/psych3030030.

[3] Attali, Y., & Burstein, J. (2006). Automatically grading the content of student essays. *Journal of Educational Technology Development and Exchange, 1*(1), 1−18.

[4] Burstein, J., Chodorow, M., Leacock, C., & Mark, A. (1999). Automated essay scoring with e-rater. *Journal of Technology, Learning, and Assessment, 2*(2), 1−35.

[5] Chary, M., Saumil, P., Alex Manini, Edward, B.,& Michael R., A Review of Natural Language Processing in Medical Education. *Western Journal of Emergency Medicine* 20, 78–86.

[6] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002) "SMOTE: Synthetic Minority Over-sampling Technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321−357.

[7] Chen, L., Ru Z., Chee Wee Leong, Blair L., Feng, G., & Ehsan, M. Hoque (2017). *Automated Video Interview Judgment on a Large-Sized Corpus Collected Online*. In 2017 Seventh International Conference on Affective Computing and Intelligent Interaction (ACII), 504–9. San Antonio, TX: IEEE. https://doi.org/10.1109/ACII.2017.8273646.

[8] Chen, W. Y., Chen, Y. J., & Lu, L. (2011). Automated essay scoring using the Flesch reading ease formula. *Journal of Educational Technology Development and Exchange, 4*(1), 1−12.

[9] Cohen, J. (1968). Weighted kappa: Nominal scale agreement with provision for scaled disagreement or partial credit. *Psychological Bulletin, 70*(4), 213–220.

[10] Devlin, J., Chang, Ming-Wei., Lee, K., & Toutanova, K. (11 October 2018). *BERT: Pretraining of Deep Bidirectional Transformers for Language Understanding*. arXiv:1810.04805v2.

[11] Eluwa J., Kuyoro S., Awodele O., & Ajayi A. (2022). Essay Scoring Model Based on Gated Recurrent Unit Technique. *International Journal of Scientific Research in Science, Engineering and Technology*, 323–330. https://doi.org/10.32628/IJSRSET229257.

[12] Firoozi, T, Mohammadi H,& Gierl,M. (2022). *Using Active Learning Methods to Strategically Select Essays for Automated Scoring.* https://doi.org/10.1111/emip.12537.

[13] Flor, M., & Hao, J. (2021): Text mining and automated scoring. In: von Davier, A.A., Mislevy, R.J., Hao, J. (eds.) *Computational Psychometrics: New Methodologies for a New Generation of Digital Learning and Assessment. Methodology of Educational Measurement and Assessment.* Springer, Cham. https://doi.org/10.1007/978-3-030-74394-9-14.

[14] Grant, D (1952). *AN EXPLORATORY STUDY OP HALO EFFECT IN RATING. The Ohio State University, Thesis*.

[15] Haller, S., Aldea, A., Seifert, C., & Strisciuglio, N. (2022). *Survey on Automated Short Answer Grading with Deep Learning: From Word Embeddings to Transformers*. https://doi.org/10.48550/ARXIV.2204.03503.

[16] Ke, Y., & Ng, H. (2019). Automated essay scoring using machine learning algorithms. *Journal of Educational Technology Development and Exchange, 2*(1), 1−12.

[17] Khyani, Divya, & B S, Siddhartha. (2021). An Interpretation of Lemmatization and Stemming in Natural Language

Processing. *Shanghai Ligong Daxue Xuebao/Journal of University of Shanghai for Science and Technology*. 22. 350−357.

[18] Kumar, A., Sharma, P., & Singh, R. (2019). Ensemble Learning Approach for Predictive Modeling Using Random Forest. *Journal of Big Data Analytics in Healthcare, 4*(2), 1−11.

[19] Leacock, Claudia, & Martin Chodorow. (2003). C-rater: Automated Scoring of ShortAnswer Questions. *Computers and the Humanities*. 37, 389–405. https://doi.org/10.1023/A:1025779619903.

[20] Li, B., & Yao, J. (2011). Automated essay scoring using Multi-classifier Fusion. *Communications in Computer and Information Science*, 233, pp. 151–157.

[21] Linacre, J. M. (1989). *Many-faceted Rasch measurement*. Chicago, IL: MESA Press.

[22] Linacre, J. M. (2018). *A user's guide to FACETS Rasch-model Computer Programs* (version 3.81.0) Retrieved from www.winsteps.com.

[23] Mathias, A., & Bhattacharyya, P. (2018). An empirical evaluation of random forest for stock price prediction. *Expert Systems with Applications*, 96, 168−183.

[24] Manvi, M., & Mishel, J. (2012). Automated Essay Grading Using Machine Learning. *Journal of Technology Research*, 3, 1−10.

[25] Medsker, L. R., & Jain, L. C. (2001). *Recurrent neural networks. Design and Applications*, 5, 64−67.

[26] Nagaraj, A., Sood, M., & Srinivasa, G. (2018). *Real-Time Automated Answer Scoring*. 2018 IEEE 18th International Conference on Advanced Learning Technologies (ICALT), 231–232. https://doi.org/10.1109/ICALT.2018.00122.

[27] Nguyen, H., & Dery, L. (2017). *Neural Networks for Automated Essay Grading*. The Hewlett Foundation: Automated Essay Scoring. Retrieved at https://cs224d.stanford.edu/reports/huyenn.pdf.

[28] NCES (2022). *Four Teams Win Top Prize in Automated Scoring Challenge for The Nation's Report Card.*

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

[30] Page, E.B. (1968). The use of the computer in analyzing student essays. *International Review of Education, 14*(3), 253−263.

[31] Patz, R. J., Junker, B. W., Johnson, M. S., & Mariano, L. T. (2002). The Hierarchical Rater Model for Rated Test Items and its Application to Large-Scale Educational Assessment Data. *Journal of Educational and Behavioral Statistics, 27*(4), 341–384.

[32] Persing, N., & Ng, H. (2013). *Support Vector Machines for Text Classification*. In Machine Learning for Text-Based Information Retrieval (pp. 93−108). Springer, Berlin, Heidelberg.

[33] Raymond, Mark R., & Houston, Walter H. (1990). *Detecting and correcting for rater effects in performance assessment*. American College Testing Program.

[34] Ramesh, D., & Sanampudi, S. K. (2022). An automated essay scoring systems: a systematic literature review. *Artificial Intelligence Review*. 55, 2495−2527. https://doi.org/10.1007/s10462-021-10068-2. Epub 2021 Sep 23. PMID: 34584325; PMCID: PMC8460059.

[35] Reckase, M. D. (2009). *Multidimensional item response theory*. Springer.

[36] Rich, C. S., Schneider, M. C. , & D'Brot, J. M. (2013). Applications of automated essay evaluation in West Virginia. In M. D. Shermis and J. Burstein, (Eds). *Handbook of Automated Essay Evaluation: Current Applications and New Directions*, pp. 99–123. New York: Routledge.

[37] Salim, A., Ahmad, N., & Zainal, A. (2019). Gradient Boosting Machine for Credit Card Fraud Detection. *International Journal of Computer Science and Information Security, 17*(4), 148−154.

[38] Sarker, A., Klein, A.Z., Mee, J., Harik, P., & Gonzalez-Hernandez, G.,(2019). An Interpretable Natural Language Processing System for Written Medical Examination Assessment. *Journal of Biomedical Informatics* 98: 103268. https://doi.org/10.1016/j.jbi.2019.103268.

[39] Silge, J., & Robinson, D. (2018). *Analyzing word and document frequency: TF-IDF*. In Text Mining with R: A Tidy Approach . Retrieved from https://www.tidytextmining.com/tfidf.htm

[40] Stefanie A. W. (2019). Examining the Impacts of Rater Effects in Performance Assessments. *Applied Psychological Measurement*, 43, 159–171. First online on doi: 10.1177/0146621618789391.

[41] Firoozi, T, Bulut, O., Epp, C. D., Naeimabadi, A., Barbosa, D. (2022). The effect of fine-tuned word embedding

techniques on the accuracy of automated essay scoring systems using neural networks. *Journal of Applied Testing Technology. 23*(1), 21−29.

[42] Taghipour, K., & Ng, H. T. (2016). *A Neural Approach to Automated Essay Scoring.* Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 1882–1891. https://doi.org/10.18653/v1/D16-1193.

[43] Trivedi, M. (2017). Self-Driving Cars. *Computer, 50*(12), 18–23. https://doi.org/10.1109/MC.2017.4451204.

[44] Uto, K., & Okano, T. (2020). *Robust Neural Automated Essay Scoring Using Item Response Theory*. Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, 2491−2497.

[45] Uto, K., Xie, B., & Ueno, K. (2020). *Neural Automated Essay Scoring Incorporating Handcrafted Features*. Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 4725−4735.

[46] Wang, Z., & Yao, L (2013). *The Effects of Rater Distributions and Rater Severity on Students' Ability Estimation for Constructed-Response items*. Research Report, ETS RR−13−23. http://search.ets.org/researcher/.

[47] Wolfe, E. W., & McVay, A. (2012). Applications of latent trait models to identifying substantively interesting raters. *Educational Measurement: Issues and Practice*, 31, 31–37.

[48] Yao, L., Li, J., Alam, H., & Melnokov, O. (2020). *An Evaluation of Tweet Sentiment Classification Methods*. International Conference on Computational Science and Computational Intelligence (CSCI), 298–303. Las Vegas, NV, USA: IEEE, 2020.https://doi.org/10.1109/CSCI51800.2020.00057.

[49] Yao, L., & Schwarz, R.D. (2006). A multidimensional partial credit model with associated item and test statistics: an application to mixed-format tests, *Applied Psychological Measurement*. 30, 469−492.

[50] Yu, D., & Deng, L. (2015). *Automatic Speech Recognition*. Springer London. https://doi.org/10.1007/978-1-4471-5779-3.

[51] Zeng, Y., Zhang, R., & Lim, T. J. (2016). Wireless communications with unmanned aerial vehicles: Opportunities and challenges. *IEEE Communications Magazine, 54*(5), 36–42. https://doi.org/10.1109/MCOM.2016.7470933.

[52] Zhang, T., Schoene, A. K., Ji, S., & Ananiadou, S. (2022). Natural Language Processing Applied to Mental Illness Detection: A Narrative Review. *Npj Digital Medicine* 5, 46. https://doi.org/10.1038/s41746-022-00589-7.

[53] Zhang, Z., Li, Y., Li, D., Chen, X., & Chen, Q. (2021). *Automated Essay Scoring with Pre-trained Language Models: An Empirical Study on Chinese Students' English Writing Proficiency*. IEEE Access, 9, 62111−62119. DOI: 10.1109/ACCESS.2021.3073416

[54] Zhao, S., Zhang, Y., Xiong, X., Botelho, A., & Heffernan, N. (2017). *A memory-augmented neural model for automated grading*. Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale. 189–192.

[55] Zhou, T., & Jiao, H., (2022). Data Augmentation in Machine Learning for Cheating Detection in Large-Scale Assessment: An Illustration with the Blending Ensemble Learning Algorithm. *Psychological Test and Assessment Modeling*.

[56] Zhu, W., & Sun, Y. (2020). *Automated essay scoring system using multi-model Machine Learning*. Proceedings a the International Conference on Machine Learning Techniques and NLP (MLNLP 2020). DOI: 10.5121/csit.2020.101211.

[57] Zou, J., Han, Y., & So, S.-S. (2008). Overview of Artificial Neural Networks. In D. J. Livingstone (Ed.), *Artificial Neural Networks*, 458, pp. 14–22. Humana Press. https://doi.org/10.1007/978-1-60327-101-1-2.

[58] Zupanc, K., & Bosnic, Z. (2016). Advances in the field of automated essay evaluation. *Journal of Educational Technology Development and Exchange, 9*(1), 1−16.

[59] Zwick, R. (1990). Do multiple-choice tests have more construct validity than open-ended items? *Educational Researcher, 19*(1), 5−14.

Appendix: A Sample Essay

One essay from prompt 1 with essay id 515 below:

"Dear, local newpaper @CAPS1 it's true technology is blinding everyone. More and more people are being sucked into electronics and less people are caring about our planet. Their are so many great things out there that could be put at risk. People @MONTH1 say that if you spend allot of time on the computer they would be able to find cure's for the deseases out there. Those are just lies to stay lasy and waste their life on the computer. What is going to happen to all of the sports in the world. The players are going to get old and they will retire and if no one is going to take their place then the games would be cancled. I am a big fan of football and I don't want to see it ge cancled. Some of my friends are saying they are addicted to the computer and I always hope they get out of that habit because what's going to happen when they get older they are going to need a @CAPS2. If they stay addicted they won't want a @CAPS2 and they what they won't have money, they can't pay bills, and they would lose their house. Why do people say they could find cures for things by looking on the computer we all know that's not true. They need to learn about them first not by just looking on the computer. They @MONTH1 not want to but they would have to look in books and do some heavey thinking before they can find the cures. I hope the ways of life change really fast because things can get really bad. If we do stop this way of life then things could get better and then our planet @CAPS3 will stay around longer and we could make the a better place."