A Neural Network approach to Automated Essay Scoring:

A Comparison with the Method of Integrating Deep Language Features using Coh-Metrix

by

Eunjin Shin

A thesis submitted in partial fulfillment of the requirements for the degree of

Master of Education

in

Measurement, Evaluation and Cognition

Department of Educational Psychology
University of Alberta

Abstract

Automated essay scoring (AES) has emerged as a secondary or as a sole marker for many high-stakes educational assessments due to remarkable advances in feature engineering using natural language processing, machine learning, and deep learning algorithms. The purpose of the study was to compare the effectiveness and the performance of two AES frameworks, each based on machine learning with deep language features and deep learning algorithms. More specifically, support vector machines (SVMs) in conjunction with Coh-Metrix features were used for a traditional AES model development and the convolutional neural networks (CNNs) approach was used for deep learning model development. Then, the strengths and weaknesses of the models under different circumstances (e.g., types of scoring rubric, length of essay, and essay type) were tested. The results were evaluated using the Quadratic Weighted Kappa (QWK) score and compared with the agreement between the human raters. The results indicated that the CNNs model performs better, producing more comparable results to the human raters than the Coh-Metrix + SVMs model. Moreover, our best models could achieve state-of-the-art performance in most of the essay sets with a high average QWK score.

Key words: automated essay scoring, convolutional neural networks, deep learning, coh-metrix, deep language features

Acknowledgements

This work would have not been possible for me to accomplish without an enormous amount of help and support from the caring and inspiring people around me. First and foremost, I would like to deeply thank my supervisor Dr. Mark J. Gierl who has continuously supported me and patiently guided me towards the completion of my study. Also, I would like to extend thanks to my committee members, Dr. Maria Cutumisu and Dr. Hollis Lai for their time and support.

Moreover, I would like to thank all CRAMErs who have inspired me and encouraged me to constantly grow as a better researcher. Especially, I am indebted to Dr. Guo, Zhang, and Ricioppo. Your friendship and priceless advice have been such a huge support on my journey.

Last but not least, this journey would have not even begun without unequivocal love and support from my family. I want to express my deepest gratitude to my parents who have taught me to pursue the right things and believed in me and my decisions. You guys have shown me what perseverance and dedication looks like. Also, special thanks go to my partner who has been wholeheartedly present throughout this process. I truly feel blessed to have you in my life.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Automated essay scoring (AES) is no longer a foreign concept to anyone in educational assessment. Originally developed to assist a traditional human marking system, where lack of consistency and scoring subjectivity were problematic (Zhang, 2013), AES frameworks have not only demonstrated high prediction accuracy, but also provided exciting benefits. Such benefits include improving the consistency of scoring, reducing time for score reporting, encouraging cost and time efficient scoring systems, and the possibility of providing instant feedback to students on their performance (Gierl, Latifi, Lai, Boulais, & De Champlain, 2014).

With the surging trend of using computers and technology in educational assessment, employing an AES system as a secondary or as a sole marker has recently resurfaced in many high-stakes exams (Shermis, 2010). For example, the Australian Education Ministry attempted to employ AES to grade written essays in their national standardized literacy assessment (the National Assessment Program - Literacy and Numeracy; NAPLAN) either as a sole marker or in conjunction with separate scores from a human marker (ACARA NASOP Research Team, 2015). Even though the ministry eventually decided to abandon the plan due to much dispute and skepticism from practitioners, it was still a valuable attempt to raise public awareness of the promise for essay scoring as well as to understand what the common concerns and perspectives are toward employing AES in high-stakes exams.

Some of the common concerns in adopting AES involved doubts in prediction accuracy, transparency, and interpretability of the scoring algorithms (Zaidi, 2016). To overcome such concerns, researchers have attempted to introduce improved AES frameworks, which evolved to produce more accurate prediction by incorporating deep learning algorithms or by utilizing deep language features (e.g., world knowledge, text easability) to ensure the model captures content of the essay and the construct of interest (e.g., Ng et al., 2014; Dong, Zhang, & Yang, 2016; Taghipour & Ng, 2016; Latifi, 2016).

Background to Problem

An AES system receives written essays and then assigns a numeric score to them reflecting the quality of the response based on the content, grammar, and organization (Taghipour & Ng, 2016). Project Essay Grade (PEG; Page, 1967) was the first attempt to establish an AES framework to reduce and overcome the drawbacks of human marking systems. PEG used a regression model with surface features, such as the document length, word length, and punctuation. Following PEG, many of the early AES frameworks focused on implementing supervised machine learning algorithms, such as regression (Page, 1994; Attali & Burstein 2004), preference ranking (Yannakoudakis, Briscoe, & Medlock., 2011; Chen & He, 2013), and classification (McNamara, Crossley, Roscoe, Allen, & Dai, 2015) using pre-identified language features.

AES systems with machine learning algorithms were typically accompanied by pre-defined language features to capture determinative information for accurate scoring. Some of the commonly selected features include nonlinguistic and linguistic features, such as total number of words per essay, sentence length, word length, the total number of grammatical errors, the types of grammatical errors, and the kinds of grammatical constructions that appeared in an essay (Kaplan, Randy, Wolff, Burstein, Lu, Rock, & Kaplan, 1998). These so-called surface-level features, in conjunction with appropriate machine learning algorithms, could produce comparable scoring results to human raters.

However, regardless of their wide usage and popularity, AES systems with machine learning algorithms were often criticized and challenged due to lack of rationale and direct connections with how human raters typically process and score responses and the inability to identify other features that define writing quality (Attali, 2013; Perelman, 2014). To overcome such limitations, recent advances in computational linguistic and natural language processing have introduced a more rational approach for extracting language features. Deep language features are theoretically and empirically driven language aspects that are more directly related to higher-order associations of essay quality (Latifi, 2016). For example, Coh-Metrix (McNarmara, Graesser, McCarthy, & Cai, 2014) is one of the tools that provides an analysis of a text based on surface-level and deep language features for in-depth text analyses. Xu and Liu (2016) introduced an AES framework using the Coh-Metrix indices and eight additional deep language features to score Chinese ESL students' writing. The results indicated that high-quality (or high-score) essays were significantly correlated with cohesion at the sentence and paragraph levels, syntactic complexity, and surface-level grammatical error. Also, according to Latifi (2016), Coh-Metrix features in conjunction with machine learning algorithms (i.e., Sequential Minimal Optimization and Random Forest; SMO & RF) could achieve very consistent and accurate prediction results comparable to current state-of-the-art results with average quadratic weighted kappa (QWK) scores of 0.69 and 0.65, respectively.

However, identifying and selecting these features can be challenging because doing so requires extensive knowledge of grammar and spelling as well as the deeper features such as semantics, discourse, and pragmatics (Dong, Zhang, & Yang, 2016). Moreover, even though the accuracy of many traditional AES systems based on machine learning is greatly dependent on these hand-selected features, only limited information about feature engineering has been released. While a few commercial AES vendors (e.g., eRater, Project Essay Grade, PaperRater) have disclosed information about their algorithms, they are often limited to general descriptions

(Wilson & Andrada, 2016). The critical elements such as included or excluded features and the weights assigned to each feature are often concealed as proprietary information. To overcome such problems related to feature engineering, AES frameworks have evolved to employ a deep neural networks model, which are capable of learning features automatically in an end-to-end manner utilizing deep learning algorithms. This new approach enables the models to make a direct prediction of essay scores without depending heavily on hand-crafted features and producing very accurate prediction results (e.g., Williams & Zipser, 1989; Miklov et al., 2010; LeCun et al., 1998; Kim, 2014; Zhao, Zhang, Xiong, Botelho, & Heffernan, 2016).

Purpose of Current Study

Machine learning and especially deep learning approaches in AES have shown promising prediction results (e.g., Yannakoudakis et al., 2011; Chen and He, 2013; Taghipour & Ng, 2016; Dong, Zhang, & Yang, 2017; Zhang, Zhang, Xiong, Botelho, & Heffernan, 2017). However, much concern still remained around both approaches as the machine learning AES algorithms are heavy dependent on features selected by humans (or feature engineering), while the deep learning AES algorithms are often perceived to only work well when there is a large training sample available. Even though a few recent studies have demonstrated deep learning AES frameworks that seem to have better prediction results compared to the previous machine learning approaches (Nguyen & Dery, n.d), no study has been conducted to compare the behaviours of the two approaches thoroughly.

Hence, the purpose of the current study is to compare the effectiveness of the machine learning and deep learning AES frameworks. More specifically, a support vector machines (SVM) model using Coh-Metrix features was implemented to compare its performance against a convolutional neural networks (CNNs) model. The following research questions will be addressed in the current study:

1) Do the deep learning AES frameworks produce more accurate prediction results compared to the machine learning AES systems with deep language features?
2) How does the model behaviour change in particular circumstances (i.e. type of rubric used for scoring, the length of essay scored, types of essay prompt)?

Chapter 2: Literature Review

In this chapter, I introduce essential information that is required to understand the AES frameworks proposed in the study. First, to understand the nature of language features introduced in the machine learning based AES model, Coh-Metrix features and indices are described. Also, theoretical background on deep learning algorithms, such as artificial neural networks (ANNs), multilayer perceptrons (MLPs), backpropagation, convolutional neural networks (CNNs), and several neural word embedding techniques are explained. Finally, the model validation metric adopted in the study is presented.

Overview of Coh-Metrix Language Features

Coh-Metrix is computerized essay-scoring that was first developed to understand natural language using over 200 features from a text corpus (Graesser, McNamara, Louwerse, & Cai, 2004). Coh-Metrix provides 108 numeric indices of the linguistic and discourse representations of a text and the values that could be used in various text analyses in order to investigate the cohesion of the text and the coherence of the mental representation of the text (Graesser, McNamara, Louwerse, & Cai, 2004; Graesser & McNamara, 2011). Coh-Metrix features not only encompass deep language features, but also surface-level descriptive features. Coh-Metrix 3.0 allows a user to analyze a corpus with web tools for a relatively small corpus consisting of less than 15,000 words per text. Otherwise, a free text analysis service is provided for larger corpora as requested.

Coh-Metrix 3.0 indices are categorized into 11 groups, which are Descriptive, Text Easability Principal Component Scores, Referential Cohesion, Latent Semantic Analysis (LSA), Lexical Diversity, Connectives, Situation Model, Syntactic Complexity, Syntactic Pattern Density, Word Information, and Readability. First, in terms of surface-level features, Descriptive indices include 11 features such as the number and the length of total words, sentences, paragraphs in a text. Descriptive indices can also be used to check the dataset and interpret patterns of data. Therefore, using Coh-Metrix features, I not only focus on deep language features, but also on surface-level language features, which are frequently highlighted in the traditional AES frameworks.

Second, Text Easability and Referential Cohesion are the two categories closely related to measuring cohesion. Cohesion is one of the most critical recurring concepts in Coh-Metrix. Coh-Metrix defines cohesion as text characteristics that help readers mentally connect ideas in the text. Text Easability indices are based on the recently updated Coh-Metrix easibility components, which provide a more complete picture of text ease and difficulty based on the linguistic characteristics of text (Graesser, McNamara, & Kulikowich, 2011). The indices include narrativity, syntactic simplicity, word concreteness, referential cohesion, deep cohesion, verb cohesion, connectivity, and temporality. More specifically, the deep cohesion index measures the degree to which the text contains causal and intentional connectives and logical relationships within the text. Similarly, verb cohesion reflects the degree of occurrence for overlapping verbs in the text, assuming more frequently repeated verbs will facilitate and enhance the understanding of the context for readers. Referential Cohesion refers to overlap in content words between local sentences or co-references. It includes five indices that measure the degree of overlap in different dimensions, such as noun overlap, argument overlap, stem overlap, content word overlap, and anaphor overlap between pairs of sentences. For example, in anaphor overlap, a pair of sentences has an anaphor overlap if the later sentence contains a pronoun that refers to a pronoun or noun in an earlier sentence.

Third, Coh-Metrix features emphasize the standing of each element in a text as well as the relationships among them, within different levels in the text. For example, lexical diversity refers to the variety of unique words that occur in a text in relation to the total number of words. It is measured using a type-token ratio, which is computed by dividing the number of unique words by the number of tokens of these words. More specifically, if the number of word types is equal to the total number of words, then the lexical diversity would be at a maximum, meaning that all the words in a sentence are different. Connectives refer to the inter-relationships among the words and sentences in a text. Connectives provide evidence about text organization (Cain & Nash, 2011). The indices measure the degree to which different types of connectives (i.e., logical, causal, temporal, and additive connectives) are used. An incidence score for each connective is provided. Syntactic complexity refers to the degree of complexity in composition of a sentence. Words are first categorized into part-of-speech categories (e.g., noun, verbs, and adjectives) or into phrases (noun-phrases, verb-phrases, and prepositional-phrases) to reveal the syntactic structure of a sentence. Coh-Metrix captures the complexity with several indices such as the number of words located before the main verb, the proportion of intersection tree nodes between all sentences, and across paragraphs in syntactic tree. Syntactic pattern density refers to the density of particular elements in a text. The indices include incidence scores, pattern density for noun, adverbial phrases, and prepositions. For example, if a text has a higher noun and verb phrasal density, it is more likely to be informationally dense with complex syntax. Latent Semantic Analysis (LSA; Landauer et al., 2007) provides measures of semantic overlap between sentences or between paragraphs. It includes eight indices computed using mean scores and standard deviations of LSA cosines for adjacent units of all sentence and paragraph pairs.

Other unique indices like readability, situation model, and word information, are also available. Readability is a rather traditional concept, which has been computed using more than 40 readability formulas (Klare, 1974). Coh-Metrix uses the two most common formulas, which are the Flesch Reading Ease Score (FRES) and the Flesch Kincaid Grade Level (FKGL). Situation model refers to the level of mental representation of a text where more implicit words are used. It includes indices such as the total incidence score of causal verbs, causal particles in text, causal cohesion, intentional cohesion, temporal cohesion, and overlaps between verbs. Word information refers to the degree to which words are assigned to syntactic categories. It includes indices such as the familiarity, concreteness, imagability, and meaningfulness of the words in a text.

Automated Essay Scoring with Deep Learning Algorithms

AES systems that use deep learning algorithms have the benefit of directly extracting features in an input text without any prior knowledge. Because they are capable of learning features automatically in an end-to-end manner, it does not require extensive knowledge in linguistics to determine which features to include in the prediction model (Williams & Zipser, 1989). More specifically, with lower layers learning basic features in essays and upper level layers learning more high level and abstract features, deep neural networks can automatically learn critical features from essays and therefore make accurate predictions (Lee et al., 2009).

Previous studies have demonstrated that deep learning AES frameworks can produce more robust results than the traditional models based on machine learning algorithms across different domains. Many different algorithms were used to demonstrate the robustness of results such as the recurrent neural networks approach (Williams & Zipser, 1989; Miklov et al., 2010;

Dong & Zhang, 2016) and convolutional neural networks (LeCun et al., 1998; Kim, 2014; Dong & Zhang, 2016).

To demonstrate the power of the AES systems, a competition on automated essay scoring called Automated Student Assessment Prize (ASAP) was organized in 2012 by Kaggle and sponsored by the Hewlett Foundation. The competition used a quadratic weighted kappa (QWK) score to measure the similarity between the human scores and the predicted scores, and the winning team demonstrated a kappa score of 0.81. Even though the winning team's algorithm was later known to utilize some hand-picked features in conjunction with machine learning algorithms, many studies were proposed to replicate or improve the kappa score using deep learning algorithms.

For example, Aklikaniotis, Yannakoudakis, and Rei (2016) implemented a single-layer long short-term memory (LSTM) approach, which is a special case of recurrent neural networks (RNNs). The results indicated that with the score-specific word embedding (SSWE), the LSTM approach could score the essays in a human-like manner, outperforming other state-of-the-art systems without any prior knowledge of the grammar or the domain of the text. Taghipour and Ng (2016) implemented and compared several deep learning approaches such as LSTM, CNNs, and a hybrid of LSTM and CNNs. Their best model, LSTM, could achieve a QWK of 0.76 on average with no prior feature engineering. Dong, Zhang, and Yang (2017) also compared LSTM and CNNs. The results indicated that their LSTM-CNN model with attention pooling could reach an average QWK of 0.76. Moreover, Zhang, Zhang, Xiong, Botelho, and Heffernan (2017) proposed a memory-augmented neural model for automated grading and their best model could achieve state-of-the-art performance on seven out of eight essay sets with a very high average QWK score of 0.78.

Overview of Artificial Neural Networks (ANNs)

Inspired by the web-like structure of the human brain, artificial neural networks (ANNs) process and store information intended to mimic how the human brain processes information. A human brain contains an enormous amount of nerve cells and neurons and each nerve cell is connected to many other cells creating a web-like design. Each cell collects input from all other connecting neural cells, and when it reaches its threshold, it signals to all the connecting cells.

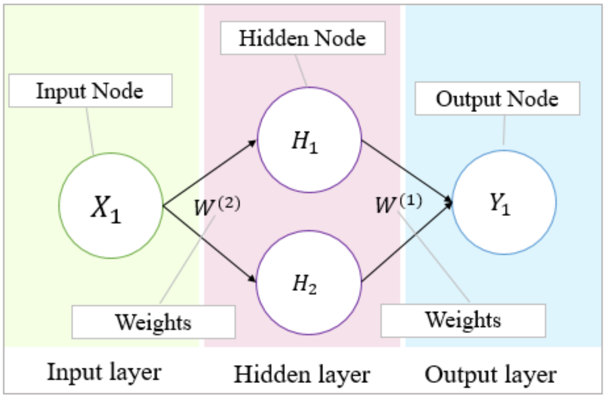


Figure 1. Simple representation of a basic neural network with one hidden layer.

Typically, ANNs are composed of multiple layers that include an input layer, a hidden layer, and an output layer. The input layer picks up the signals and passes them on to the next layer followed by an output layer which delivers the results. Each layer could encompass several nodes (i.e., $X_1, H_1, H_2, and\ Y_1$). When multiple input values are passed to a neuron, the neuron processes its values using the following three processes: weighting the input variables, calculating the scores, and applying the activation functions. First, when an input signal comes in, it gets multiplied by an assigned weight value, $W^{(2)}\ and\ W^{(1)}$. Then, the modified (or weighted) input values are summed up to a single value. Last, the result of the neuron's calculation is turned into an output signal value using the assigned activation transfer function so it could be matched with the outcome variables, Y_1. A schematic view of this model is presented in Figure 1.

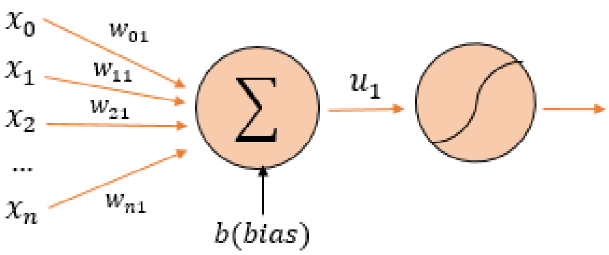


Figure 2. Simple representation of a neuron with the sigmoid activation function and a bias unit.

Moreover, a bias unit can be added to enable more flexible learning after the input variables are weighted and summed. A bias unit is simply an additional neuron added to each layer that is independent from the previous input layer. A bias unit plays a significant role in producing flexible learning, as it can shift the layer to model a data space apart from the origin. Also, it allows the layer to pass non-zero inputs for the next layer when all the inputs are zeros.

For example, assume we are trying to do a simple binary classification with a set of $n + 1$ inputs $\{x_0, x_1, \ldots, x_n\}$ and the pre-labeled outcome values {0, 1} and we aim to understand how the first hidden node (H_1) processes the information. Consider a set of $n + 1$ inputs $\{x_0, \ldots, x_n\}$ and given the inputs, a neuron first computes a weighted linear combination using the inputs and a bias. A weight parameter in the first hidden node, w_{i1}, is defined for each input and we can compute the summed value, u_1 as in formula (1) below. Then, $f(x)$, a non-linear activation function, such as the sigmoid activation function [see formula (2)] is applied to get a predicted output value, as in formula (3). A schematic view of this model is presented in Figure 2.

$$u_1 = \sum_{i=0}^{n} w_{i1}\, x_i + b_1, \tag{1}$$

$$f(x) = \frac{1}{1+e^{-x}}, \text{ and} \tag{2}$$

$$f(u_1) = f(\sum_{i=0}^{n} w_{i1}\, x_i + b_j). \tag{3}$$

One of the most important sources of information that must be learned in these formulas is the weights w_{i1}. The weights can be learned using an optimization algorithm called ‘back propagation.’ Backpropagation is often used in network optimization due to its simplicity and efficiency. Backpropagation algorithms can be broken down into two parts: feed-forward and backward-propagation. In feed-forward, the training inputs are introduced and provided to each layer accordingly. Next, the error is calculated and propagated layer by layer from the output to the input in a reversed order. Meanwhile, the weights and biases for each node and layer are updated according to the defined error function. In the current study, we used categorical cross entropy as the error (or loss) function and aimed to minimize the error between the labeled outputs and the predictions.

Overview of Multilayer Perceptrons (MLPs) in Deep Learning

A simple logistic regression can only solve problems when the variables are linearly separable. To solve more complex problems, we could use ANNs instead by adding a hidden layer to so that the networks can learn non-linear representations as well. However, to ensure the model to learn more abstract representations, it is important to construct a deep network model by providing more hidden layers. Multilayer perceptrons (MLPs) are feed-forward neural networks with hidden layer(s). In MLPs, the hidden nodes are fully connected to each other which permits passing information from the input or the previous layer to the output. Successive model layers can learn deeper intermediate representations, thereby increasing the prediction or classification accuracy. For example, three hidden layers with varying sizes of nodes were stacked to produce 2d outputs in Figure 3.

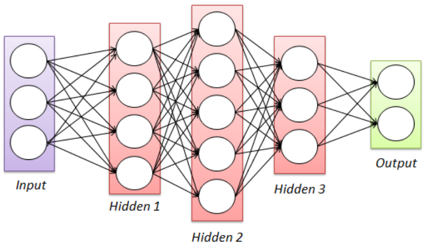


Figure 3. Conceptual representation of deep neural networks.

Overview of Convolutional Neural Networks (CNNs)

Convolutional neural networks can be described as a special case of a multilayer perceptron. CNNs were inspired by a cat's visual cortex and how the cells in the cortex are sensitive to small sub-regions of the visual field (Hubel & Wiesel, 1962). The cells would work as filters in the visual input and extract recognizable and critical features. Inspired by visual cortex functions, CNNs have been traditionally used in image recognition and image processing and recent studies have demonstrated successful implementation for using CNNs in AES (e.g., LeCun et al., 1998; Kim, 2014; Dong & Zhang, 2016).

A CNN typically consists of one or more convolutional layers, pooling layers, and fully connected output layers. It takes an array of data as input and aims to produce feature maps using kernels. Kernels are the filters that contain shared weights. Kernels are applied to the input to produce feature maps. In addition to their unique architecture, CNNs are also renowned for their efficient learning characteristics. For example, neurons are not fully connected, but rather selectively connected in CNNs unlike typical MLPs. This is called sparse interaction and it can be achieved by making the kernel size smaller than the input. For example, in Figure 4, not every input value in the first layer is connected to each feature map value as we chose the kernel size, three, which is smaller than the input size, nine.

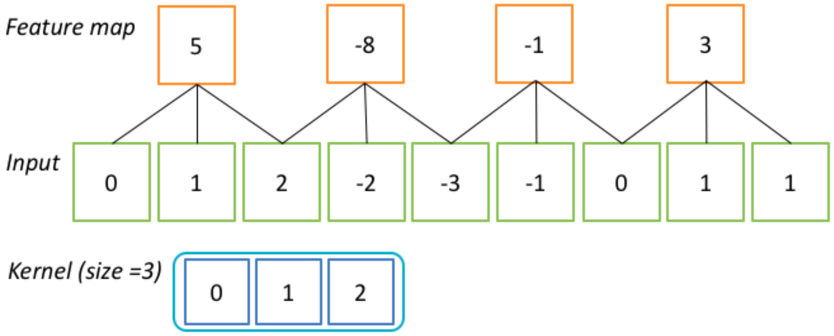


Figure 4. Conceptual representation of 1d discrete data convolution.

Sparse interactions among the neurons allow the network to efficiently describe complicated interactions among many variables. For instance, when processing an image, the input image typically contains millions of pixels. However, by limiting the number of connections each output can hold, we do not necessarily need to learn and store all the parameters connecting the entire set of input. Many studies have demonstrated the efficiency of reducing the number of connections in CNNs while achieving comparable accuracy with a little to no loss of information (e.g., Changpinyo, Sandler, & Zhmoginov, 2016; Liu, Wang, Foroosh, Tappen, & Pensky, 2015).

Parameter sharing is the process of using the same parameter for more than one function in a model. This process can be achieved by using only one shared set of parameters for the kernel that is used at every position of the input. Sharing weights has benefits because it can significantly reduce the number of parameters that must be learned. For example, in Figure 5, the kernel consists of shared parameters (i.e., $w_0, w_1, w_2, and\ w_3$). As the kernel moves across the input image, the values in the feature map are computed with matrix-multiplication among the weights and the input values.

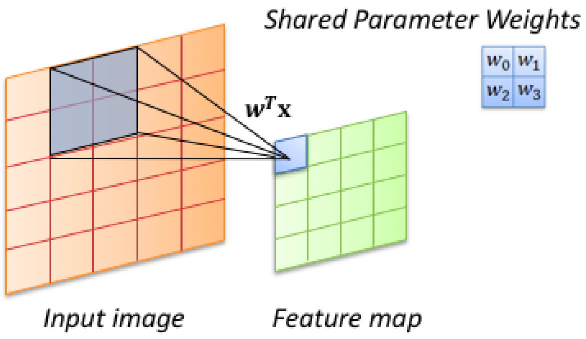


Figure 5. Conceptual representation of 2d discrete data convolution.

Typically, CNNs have a unique architecture where convolutional layers and pooling layers appear alternatively, followed by a fully connected dense layer to produce outcomes. By using multiple layers and large numbers of filters, the CNN architecture can provide vast amounts of representational power to solve complicated tasks. Consider a simple image recognition problem where the input value is a 2d image with K height and Kweight where the goal is to recognize the image and classify it into predefined categories (see Figure 6). In this case, convolutional layers take several features as input and produce M feature-maps as output. More specifically, the convolutional layer consists of two stages, where the layer performs several parallel convolutions to produce a set of values using kernels. Then, nonlinear activation functions are applied to the output to compute values for the feature maps.

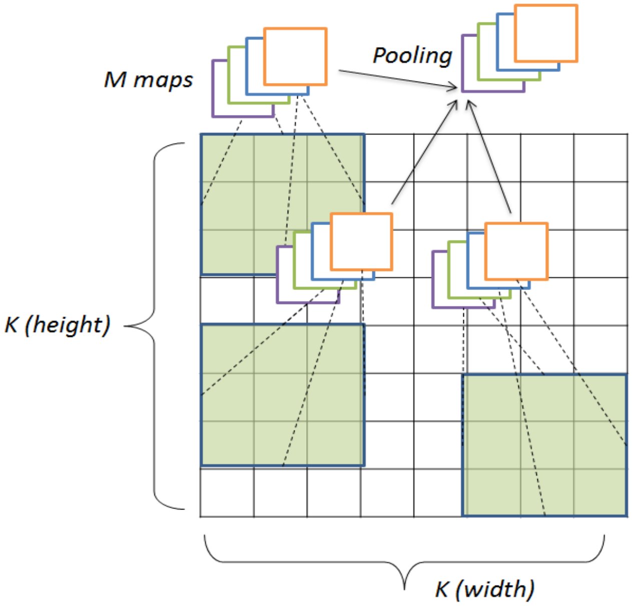


Figure 6. Conceptual representation of convolutional and pooling layers in CNNs.

Pooling is then used to reduce the dimensionality of the convolutional responses. In other words, the pooling layer downsamples the feature maps to reduce the number of parameters and networks required for learning. For example, in case of image recognition, if the goal is to recognize whether an image contains a dog, then we do not need to know the exact location of the dog's face. Rather, we care more about whether certain features exist, not where they exist. Pooling can help the model learn to be more robust to the noise and generalizable, thereby achieving invariance against minor local changes in the dataset. Several different pooling strategies can be used such as spatial pooling, average pooling, and max pooling. In our study, max pooling was used in the model structure. In max pooling, the response for each block is taken to be the maximum value over the block responses. In a case where the convolutional response map is a 4-by-4 grid and we pool over the four 2-by-2 grid blocks, we will be able to get the pooled response as presented in Figure 7 using max-pooling.

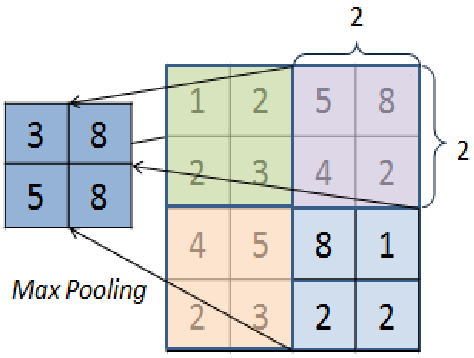


Figure 7. Example of max pooling in CNNs.

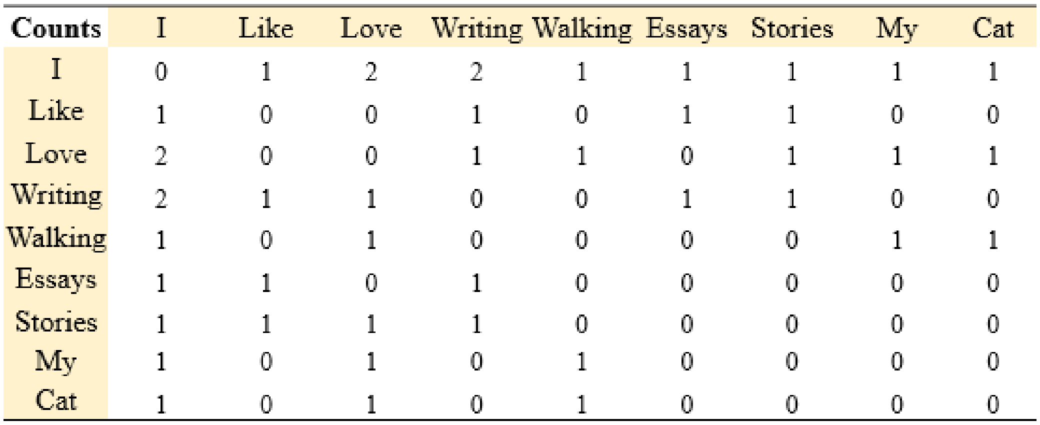
After alternating convolutional and pooling layers, the features are fed into a fully connected layer, which is also called a dense layer. Multiple dense layers can be stacked, where the last dense layer depends on the format of the output and the types of problems the model was attempting to learn. For instance, the output in the current study is a varying size of essay scores, which range from 0 to 60. As a result, the last dense layer should contain 61 neurons to produce outcomes for a multiclass classification.

Overview of Neural Word Embedding

Word embedding is a technique for representing a word as a real number vector while preserving its meanings, semantic relationships, and alternative meanings by using distances among the words. Words that are closely related (i.e., similar meanings, similar locations in a sentence) should be located in a close vector space while words that are far apart should be more distant from one another. For example, ‘location’ and ‘destination’ are semantically related words, so the reasonable embedding space would represent them as vectors that are not far apart. Inspired by one of the most successful ideas in statistical natural language processing, word embedding has been critical to improving the performance of various natural language processing tasks, such as syntactic parsing (Socher, Bauer, Manning & Ng, 2013) and sentiment analysis (Socher, Perelygin, Wu, Chuang, Manning, Ng, & Potts, 2013). Word embedding is also an essential procedure in AES as many machine learning algorithms and almost all deep learning algorithms are incapable of processing strings or plain text in their raw form. Rather, they require text to be processed and represented as numeric input to perform accurate predictions.

Word2vec (Mikolov, Kombrink, Burget, Cernocky, & Khudanpur, 2011) is a commonly used word embedding technique, and it is often considered the de facto standard for pre-trained word embedding in text analysis. Word2vec attempts to learn geometrical representation of words from their word co-occurrence information using predictive models. More specifically, Word2vec uses the two predictive models, which are the continuous bag-of-words (CBOW) and continuous skip-gram (CSG) to compute the probability of a target word to occur in a certain context that is defined by neighbouring words. While the CBOW model learns the embedding by predicting the current word based on its context, the CSG model learns by predicting the surrounding words given a current word. Using the two methods, Word2vec learns the relationships between the target word and context word one-by-one as it moves through every target word in a corpus. As a result, the learning could take a significant amount of time when a text includes a large number of words.

To improve the learning efficiency, GloVe (Pennington, Socher, & Manning, 2014) uses statistical information (i.e., word counts) achieved from a count-based model to produce vector representations of words. As the name suggests, the model uses information about how frequently words appear together in a corpus and stores this information in a matrix form. For example, in Figure 8, the matrix shows how the co-occurrence information can be represented for the following example sentences: “I like writing essays.”, “I love writing stories.”, and “I love walking my cat.” Then, a dimensionality reduction is conducted on the matrix. Dimensionality reduction aims to extract the most meaningful information from the original matrix while minimizing the loss of critical information. The final product should have a lower-dimensional matrix and it should consist of vector representations of each word in each row.



Counts	I	Like	Love	Writing	Walking	Essays	Stories	My	Cat
I	0	1	2	2	1	1	1	1	1
Like	1	0	0	1	0	1	1	0	0
Love	2	0	0	1	1	0	1	1	1
Writing	2	1	1	0	0	1	1	0	0
Walking	1	0	1	0	0	0	0	1	1
Essays	1	1	0	1	0	0	0	0	0
Stories	1	1	1	1	0	0	0	0	0
My	1	0	1	0	1	0	0	0	0
Cat	1	0	1	0	1	0	0	0	0

Figure 8. Example of the words co-occurrence matrix.

Previous studies have attempted to compare the efficiency of Word2vec and GloVe embedding in learning vector representation. Some studies have demonstrated that Word2vec could learn vector representation more efficiently regardless of the language type compared to GloVe (e.g., Naili, Chaibi, Hajjami, & Ghezala, 2017; Berardi, Esuli, & Marcheggiani, 2015). However, many recent studies have demonstrated that GloVe yields better performance with less overfitting in various text analyses (Dhingra, Liu, Salakhutdinov, & Cohen, 2017). Moreover, GloVe showed additional surprising benefits, such as reduced learning time and adaptability in both a small and a large corpus, as well as a smaller vector size (Le & Mikolove, 2014).

Therefore, we decided to use the pre-trained GloVe model (Pennington, Socher, & Manning, 2014) as our embedding. Several dimensions (e.g. 50d, 100d, and 300d) of pre-trained weights are available for GloVe embedding and in the current study we used the 300d weights from both Wikipedia 2014 and Gigaword 5.

Evaluation Metrics and Quadratic Kappa Score

Model validation in AES depends on comparing the similarity between the model performance and human raters (Attali, 2013; Chung & Baker, 2003; Williamson, Xi, & Breyer, 2012). In this comparison, human judges are considered the 'gold standard' and function as the explicit criterion for evaluating the performance of AES frameworks (Latifi, 2016). Various validity coefficients have been adopted as evaluation metrics in previous studies to measure correlation or agreement. These measures include kappa score, quadratic kappa score (QWK), Pearson's correlation, Spearman's correlation, and Kendall's Tau (Taghipour & Ng, 2016).

In the current study, we adopted QWK as our main evaluation metric, which was the official evaluation metric of the AES competition, where the dataset of the current study originated. Kappa score and QWK are often used as consistency measures (Cohen, 1960). The kappa score provides a chance-corrected index and is computed based on the ratio of the proportion of times the agreement is observed to the maximum proportion of times that the agreement is made while correcting for chance agreement (Siegel & Castellen, 1988). It ranges from one, when agreement is perfect, to zero when agreement is not significantly better than chance. Conventionally, a kappa score greater than 0.80 is considered good agreement and a score greater than 0.60 is considered moderate agreement.

Kappa, however, does not account for the degree of disagreement. Therefore, a weighted kappa score is used to overcome this problem when ordered categorical data is used. Where i

represents a human-rated score and j represents a machine-rated score, and N is the number of possible ratings, a weight matrix W can be constructed as follows:

$$W_{i,j} = \frac{(i-j)^2}{(N-1)^2}. \tag{3}$$

Then a matrix O is constructed so that $O_{i,j}$ represents the number of essays that receive a rating i by the human and a rating j by the machine. An expected count matrix E is computed as the outer product of histogram vectors of the two ratings. The matrix is normalized so that the sum of elements in E and O are the same. QWK can be calculated as follows:

$$QWK = 1 - \frac{\sum_{i,j} W_{i,j}\, O_{i,j}}{\sum_{i,j} W_{i,j}\, E_{i,j}}. \tag{4}$$

Chapter Summary

In the chapter, I provided a general overview of essential concepts required for the development and evaluation of AES frameworks. Traditional machine learning and deep learning AES frameworks place distinctive emphases on pre-identified features and modeling algorithms, respectively. Coh-Metrix features and indices can successfully represent determinative features required for accurate prediction with a wide range of language features based on computational linguistics. Such indices encompassed both surface-level (i.e., Descriptive indices) and deep language features (i.e., Text Easability, Principal Component Scores, and Referential Cohesion).

By way of comparison, deep learning AES frameworks focus on modeling prediction algorithms. A comprehensive review of previous relevant studies has demonstrated how the Long Short-term Memory (LSTM) approach and the Convolutional Neural Networks (CNNs) approach could be used in AES. It is also noted that CNNs is one of the special cases of neural network approaches that could be successfully implemented for essay grading with the help of neural word embedding techniques such as Word2vec and GloVe embedding.

Chapter 3: Method

In this chapter, I describe the dataset used in the current study, the model development and architecture, and the evaluation measures used for model validation.

Dataset for Automated Essay Scoring

The dataset used in the study was collected and released as part of a competition on AES called ‘Automated Student Assessment Prize’ (ASAP), which was organized by Kaggle and sponsored by the Hewlett Foundation. The dataset consisted of eight essay sets. Each essay set consisted of different types of topics that were persuasive, narrative, or source-dependent. The responses were written by students in grades 7 to 10 and they ranged from 150 words to 650 words. All of the responses were transcribed so they could be evaluated using AES procedures. Student responses were rated by either two or three human raters.

In essay set 1, students in grade 8 were provided a short prompt and required to write a short letter to a local newspaper stating their opinion on the topic with the goal of persuading the readers to agree with them (see Appendix A for more detail). The resolved score for the first essay set ranged from 2 to 12. Even though the score category was not large, challenges in model training arose due to the score distribution. While most of the essay scores ranged from 6 to 10, not enough response samples were provided for lower range score categories. For example, only one out of 1783 essays was given score 3. Also, there were only 10 essays assigned score 2 (see Figure 9)

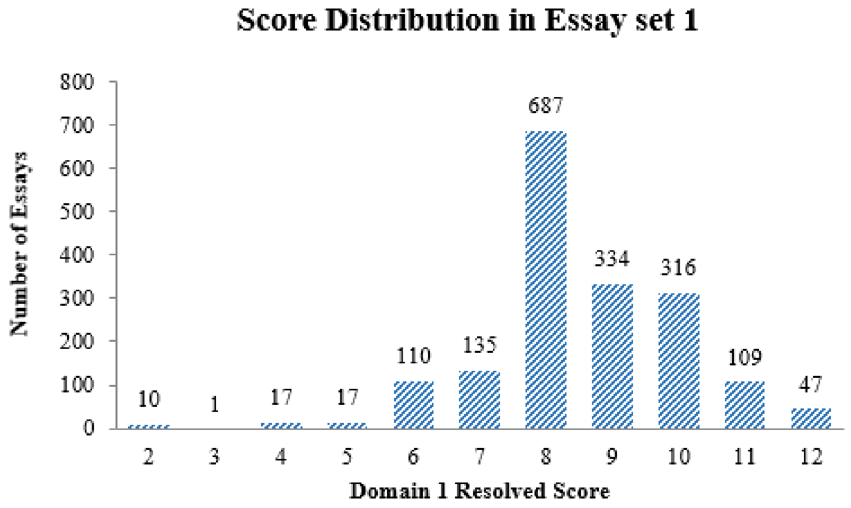


Figure 9. Score distribution of essay set 1.

In essay set 2, grade 10 students were provided a short prompt and required to write a short persuasive essay to a newspaper reflecting their views on censorship in libraries (see Appendix A for more detail). Unlike other essay sets, it was scored using both domain 1 and 2 scoring rubrics. The first rubric was designed to score students' writing applications while the second rubric was designed to score students' language conventions. The score for writing applications ranged from 1 to 6, while the score for writing conventions ranged from 1 to 4. Again, the score distribution of writing applications did not have enough representative samples given for certain score categories, such as score 6 in this case (see Figure 10).

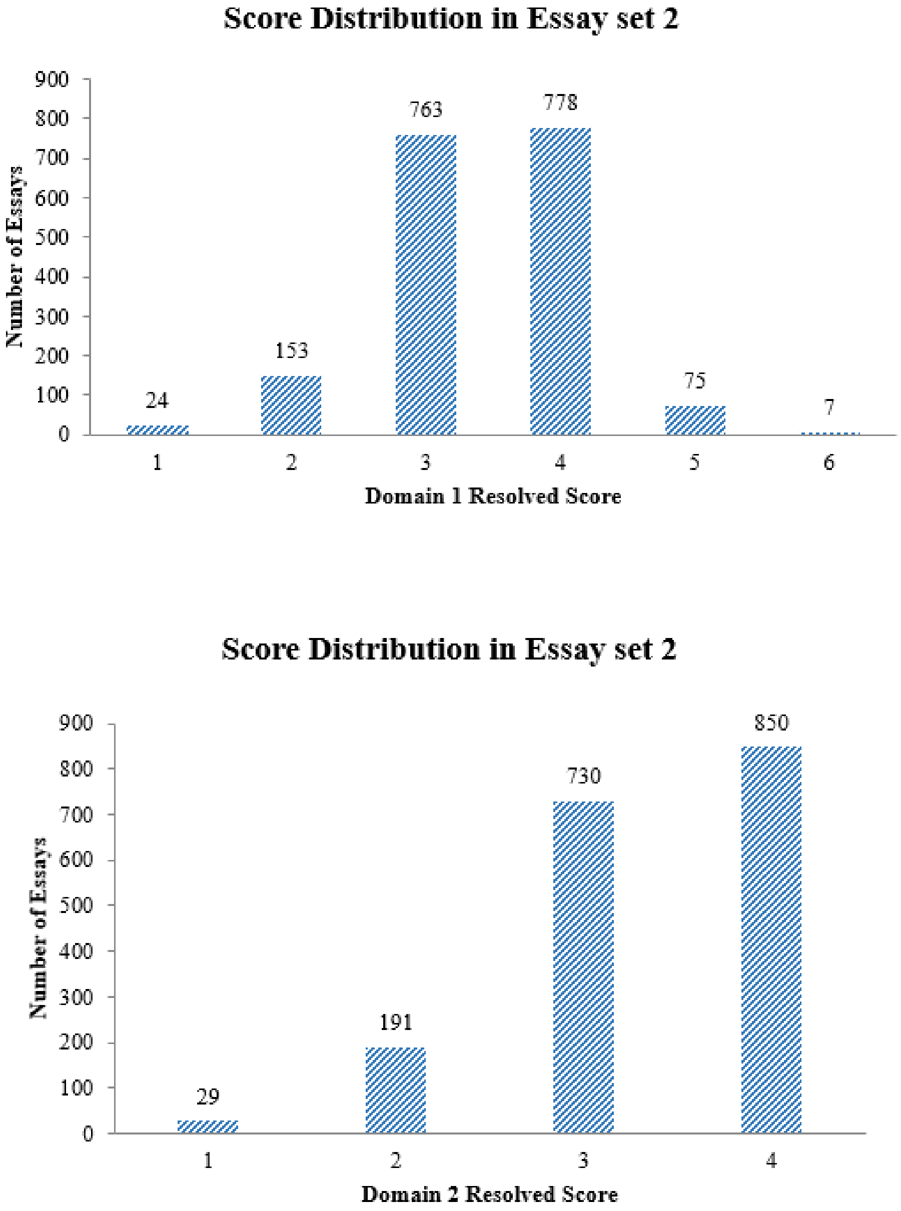


Figure 10. Score distribution of essay set 2a and 2b.

In essay set 8, grade 10 students were provided a short prompt and required to write a story related to laughter (see Appendix A for more detail). The essays were scored by three human-raters and the resolved score ranged from 0 to 60. Similar challenges arose in model training due to a relatively large score range (i.e., 0 to 60) and the score distribution. While most of the essay scores ranged from 29 to 49, not enough representative sample responses were given for lower (i.e., 1 to 29) and upper ranges of the score categories (i.e., 49 to 60; see Figure 11 for more information).

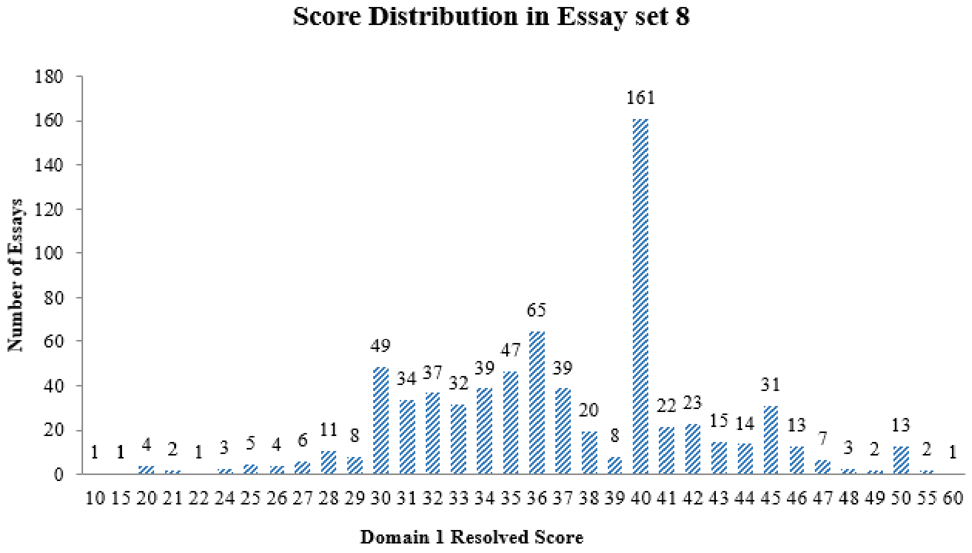


Figure 11. Score distribution of essay set 8.

Only the training dataset was released originally, so in the current study, I partitioned the released training dataset into training, testing, and validation datasets to prevent overfitting and evaluate the final accuracy. Validation set is often used as a criterion to select the best models among the runs with different hyper-parameters and to prevent overfitting. For example, when the validation error starts increasing, even if the training error keeps decreasing, we can assume the model started overfitting the training dataset. The testing set is used to evaluate the final model performance after the best model is selected based on the validation set. In the current study, 10% of the original training set was assigned for validation, 10% of the remaining training set was assigned for testing, and the remaining responses were used for training (see Table 1).

When the responses are blindly distributed for training, testing, and validation, it is possible for the model to encounter responses with unlearned score categories in model evaluation. For example, only one essay response had score 10 in essay set 8. If the response was assigned as part of the testing set, the model could not have a chance to learn generalizable patterns for the score category 10, thus it could result in lowering the model accuracy. To avoid this problem, we attempted to evenly distribute every score category for training, testing, and validation by checking whether certain response categories were included in the training set after the random assignment. If not, we randomly shuffled the dataset until the requirement was satisfied.

Table 1. Descriptive Statistics of the ASAP Dataset

	Essay Set								
	1	**2a**	**2b**	**3**	**4**	**5**	**6**	**7**	**8**
Essay type	Persuasive			Source Dependent			Narrative		
Average word length	350	350	350	150	150	150	150	250	650
Domain 1 score	2-12	1-6	-	0-3	0-3	0-4	0-4	0-24	0-60
Domain 2 score	-	-	1-4	-	-	-	-	-	-
Grade	8	10	10	10	10	8	10	7	10
Training N	1443	1458	1458	1397	1434	1461	1458	1270	585
Testing N	161	162	162	156	160	163	162	142	65
Validation N	179	180	180	173	178	181	180	157	73
Total N	1783	1800	1800	1726	1772	1805	1800	1569	723

Machine Learning Model Development

Model 1: Coh-Metrix Features and Support Vector Machine (SVM). The first model was developed using the entire 108 Coh-Metrix features in conjunction with a machine learning classification algorithm called a support vector machine (SVM). An SVM produces classifications by locating a hyper-plane, which is intended to distinguish the identified classes as clearly as possible. For example, three hyper-planes (A, B, and C) are introduced to classify the stars from circles in Figure 12. In this case, it is very intuitive to choose the plane A to separate the sets most accurately.

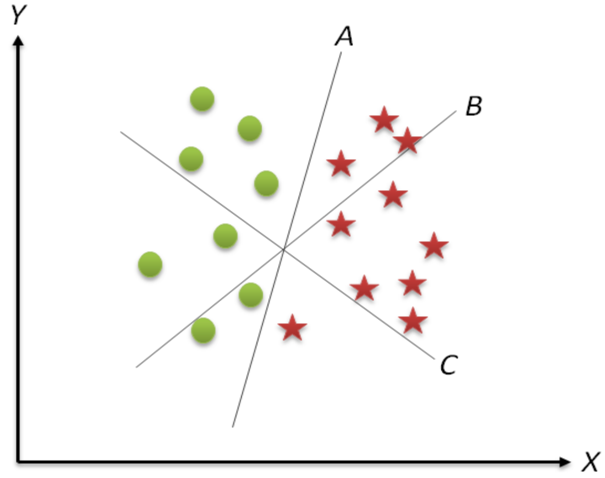


Figure 12. Conceptual representation of SVMs classification.

SMVs are often used in classification problems such as pattern recognition, image classification, text and hyper-text categorization, and automated essay scoring (Joachims, 1997; Burges, 1998; Tong & Koller, 2002). Just like other machine learning algorithms for AES, SVMs require pre-selected features such as word length, word level, spelling errors, sentence length, and sentence level (Chen & He, 2013). Recent studies have demonstrated the significance of utilizing Coh-Metrix features in conjunction with machine learning algorithms such as SVMs to improve the performance in various text analyses (Xu & Liu, 2016, Latifi, 2016; see Table 2 more information about the Coh-Metrix features).

Table 2. *Coh-Metrix Features*

Feature Name	N	Feature Name	N
Descriptive	11	Syntactic Complexity	7
Text Easability Principle Scores	16	Syntactic Pattern Density	8
Referential Cohesion	12	Word Information	22
LSA	8	Readability	3
Lexical Diversity	4		
Situation Model	8	**Total**	**108**

Using full Coh-Metrix features with an SVM, Latifi (2016) produced very consistent and accurate prediction results that were comparable to current state-of-the-art on the same dataset. In his study, a polynomial kernel SVM could produce the best results when combined with Coh-Metrix features, with an average quadratic weighted kappa score of 0.68. Inspired by the effectiveness of this method, we used similar machine learning algorithms while experimenting with several hyper-parameters such as the type of kernel transformation, and degree of regularization and a slack variable for improved accuracy. More information about the hyper-parameters is presented in Table 3.

Table 3. *Coh-Metrix + SVM Model Hyper-parameters*

Parameter Name	Parameter Value
Degree (Slack Variable; C)	1, 2
Kernel	Linear, Polynomial, Radial basis function (Rbf)
Regularization parameter	1, 10, 100
Epochs	20
Batch size	128

Deep Learning Model Development

Data Processing and GloVe Embedding. Prior to transforming the essay responses into vectors for word embedding, critical pre-processing steps were conducted to decrease the noise in the model learning and predictions (see Figure 13). For example, all the words were converted to lower cases and lemmatized using the Python NLTK library (Bird et al., 2009). Lemmatization is the process of grouping the words together so that they can be analyzed as a single item based on their dictionary form. Non-alphabetic words and numbers (e.g. @, #, %, 0-9) were eliminated while punctuations were kept and treated as separate words. Then, the cleaned responses were tokenized. Tokenization is the process of breaking down a text into individual words (or tokens). Each token was assigned a unique numeric index so that the index matched the location of the word in an embedding matrix.

After every essay response was converted into a different size of row vectors, each row vector was padded with zeros to keep the vector-size even for the entire essay responses. This step was necessary as CNNs only take inputs of the same length. If the first essay contains only 100 words while the second essay response contains 120 words, then the first essay set should be padded with 20-zeros to make the vector-size even between the two responses. Finally, the word embedding weight matrix was constructed for the unique words located in the essay sets using the Stanford's publicly available GloVe 300-dimensional embeddings, which were trained on six billion words from Wikipedia 2014 and Gigaword 5 (Pennington et al., 2014).

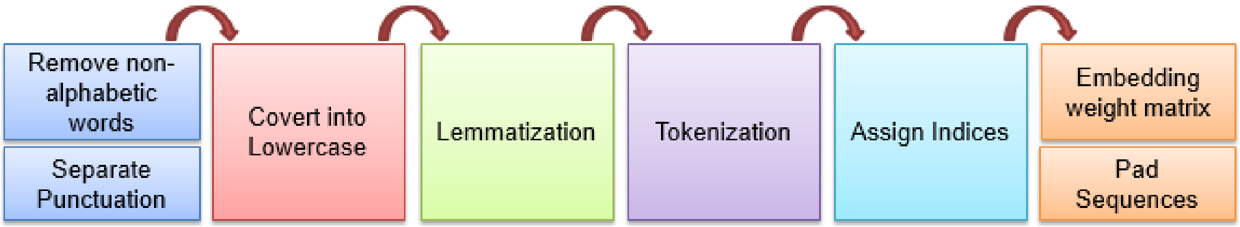


Figure 13. Data processing and embedding procedures.

Model 2: Convolutional Neural Networks (CNNs). We implemented CNNs using Keras. Keras is a modular neural network library written in Python that runs with Tensorflow or Theano backends. In terms of the overall model frameworks, the CNNs were composed of multiple layers starting from an embedding layer with GloVe pre-trained weights, three-layer convolutional networks, and dense layers. The embedding layer serves as a lookup table so each essay response that was encoded with numeric indices could be mapped onto a continuous vector space. In the three-layer convolutional networks, each layer of the CNN consisted of a convolutional layer and a max-pooling layer. The output of the last pooling layer was flattened into a 1d feature vector that represents each essay response. The 1d feature vector was then fed into fully-connected dense layers. A dense layer is a typical neural networks layer where every neuron is connected to every other neuron. More information about the overall model frameworks is described in Figure 14.

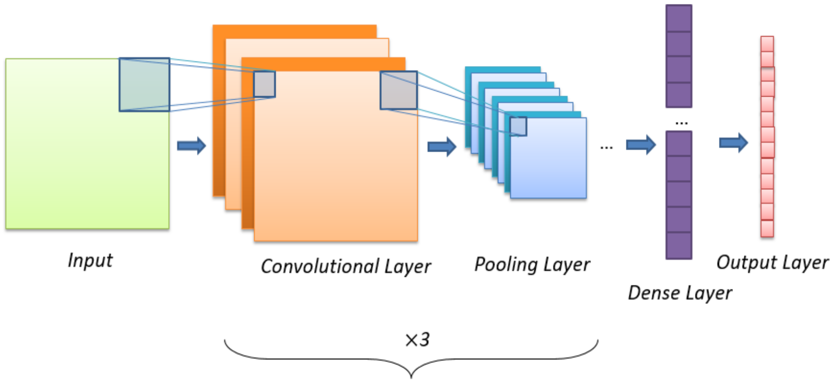


Figure 14. Conceptual representation of CNNs model architecture in the current study.

More specifically, the padded input was a matrix with a dimension equal to the number of essays × maximum sequence length. The embedding layer receives integer inputs and maps them to the pre-trained weight of the corresponding index. The output of the layer will have a shape of the number of essays × maximum sequence length × embedding dimension. For example, if the input response ‘I had a very very busy day’ was encoded as [1, 2, 3, 4, 4, 5, and

8], the embedding layer (or lookup layer) will assign corresponding embedded weights for each token. A dropout was added to the output of the embedding so that the learned pre-trained embedding is more generalizable. Dropout is a regularization technique where randomly selected neurons are ignored during training (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014). As some of the dropped neurons temporarily stop passing information to the next neurons, the network becomes less sensitive to the specific weights of neurons. Because we have embedded all the tokenized words that we could locate from the GloVe pre-trained word embedding, we decided to introduce dropout to prevent the networks from being overly sensitive to some of the unique and unrepresentative words.

Feature maps were generated using kernel weights and the input. The size of kernels was adjusted as part of hyper-parameters to select the best model. Also, it is conventional to apply a non-linear activation function before the pooling layer to introduce some nonlinearity in the model leaning. We chose the rectifier linear units (ReLU) activation function in each convolutional layer as the non-linear activation function. ReLU activations are one of the simplest non-linear activation functions due to the efficiency of calculating their derivatives. More specifically, the function has a very simple structure as the activation is set at a threshold of zero, meaning it always returns zero as an output when inputs smaller than zeros are introduced (see Figure 15).

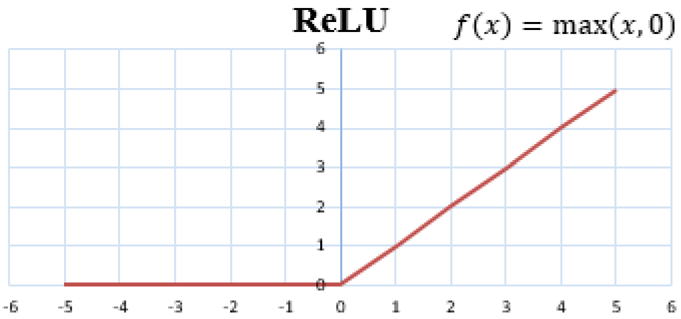


Figure 15. Rectified Linear Units (ReLU) activation function.

After the non-linear activation function was applied to the feature maps, max-pooling was conducted. For the first CNNs model, three convolutional layers were stacked alternatively with max-pooling layers before it was flattened to be fed into a fully connected dense layer with 100 neurons. The final dense layer had 13 neurons to predict the score category. The softmax activation function was required in order to provide comparable categorical results for the current dataset. The softmax activation function is a generalization of the sigmoid activation function, where the outputs are mapped to range from zero to one. For example, in logistic regression, the sigmoid activation function is used as the activation function to conduct binary classification. Unlike the sigmoid function, softmax is used for multiclass classification (or multinomial logistic regression) where more than two output categories are provided. As the current dataset included responses scored in various categories (e.g., 2 to 12 in essay set 1), it was important to choose the softmax activation for the last output layer. More information about the hyper-parameters and architecture can be found in Tables 4 and 5.

Table 4. *CNN Model Hyper-parameters*

Layer	Parameter Name	Parameter Value
Embedding	Embedding dimension	300
Dropout	Dropout rate	0.20, 0.50
CNN	Number of filters	50, 100, 200
	Kernel size	2, 3, 4, 5
Dense	Number of neurons	50, 100, 200
Model Compile	Epochs	20
	Batch size	128

Table 5. *CNN Model Architecture for essay set 1*

Layer	Output Shape	Parameter #
Embedding	(None, 874, 300)	2,565,300
Dropout	(None, 874, 300)	0
Convolutional	(None, 870, 70)	75,050
Pooling	(None, 435, 50)	0
Convolutional	(None, 431, 100)	25,100
Pooling	(None, 215, 100)	0
Convolutional	(None, 211, 200)	100,200
Pooling	(None, 105, 200)	0
Flatten	(None, 21000)	0
Dense	(None, 100)	2,100,100
Activation	(None, 100)	0
Dense	(None, 13)	1,313
Activation	(None, 13)	0
Total parameters: 4,867,063		
Trainable parameters: 2,301,763		
Non-trainable parameters: 2,565,300		
Train on 1443 samples, validate on 161 samples		

Last, for model training, the objective of the both Coh-Metrix + SVM and CNNs was to minimize a conventional loss function for classification problems described as the categorical cross-entropy. A cross-entropy shows the distributional differences between the two targets. Where k is the number of inputs, y is the output and $\hat{y}$ is the predicted output, categorical cross-entropy is defined as follows:

$$E(y, \hat{y}) = \sum_{i=1}^{k} y_i \log(\hat{y}_i)\,. \qquad (4)$$

In the study, to minimize the objective function for the CNNs, I used the Adam optimization algorithm. The Adam optimization algorithm is one of the prevalently adopted parameter optimization algorithms in deep learning, especially in computer vision and natural language processing. Adam is commonly preferred over the traditional stochastic gradient descent optimization due to many benefits. Such benefits include computational efficiency, less memory requirements, and good performance in optimizing non-stationary objectives. In addition, Adam is also known to escape from saddle points and local minima significantly faster than other optimizers. Therefore, it was suitable to adopt Adam in training the CNNs for more efficient learning.

Evaluation and Comparison of the Prediction Model

To compare and evaluate the prediction accuracy in the proposed models, quadratic weighted kappa score (QWK) was used as an agreement measure. QWK was the official agreement measure in the Automated Student Assessment Prize (ASAP) competition, where the dataset of the current study originated from. Also, most of the studies that developed AES systems using the competition dataset reported QWK as one of their main evaluation criteria (Dong & Zhang, 2016, Tahgipour & Ng, 2016).

Chapter Summary

In the chapter, it was noted that a machine learning AES framework can be modeled by extracting pre-identified features, locating an appropriate learning algorithm, and adjusting model hyper-parameters. Likewise, a deep learning based AES framework requires pre-processing the responses, embedding words to assign initial weights, constructing an appropriate learning algorithm, and adjusting model hyper-parameters as it learns the patterns and makes predictions.

A review of previous relevant studies has demonstrated successful implementations of learning algorithms such as support vector machines (SVMs) and convolutional neural networks (CNNs) in essay scoring. Also, it is noted that constructing powerful learning algorithms require a fair amount of exploration with adjusting modeling details such as tuning hyper-parameters (i.e., kernel size, kernel transformation, and regularization parameters).

Finally, quadratic weighted kappa (QWK) can be used for evaluation. Not only can it be used to compare the results of the two proposed frameworks, but it also lets us compare the results of our best model with the results of the previous studies. In short, the chapter introduced a step-by-step guide for developing machine learning and deep learning AES systems.

Chapter 4: Results

In this chapter, I present and compare the findings of the two proposed AES frameworks, Coh-Metrix + SVM and CNNs. First, the information about final model specifications is presented. More specifically, model hyper-parameters selected for the best accuracy results are introduced. Second, prediction results of the Coh-Metrix + SVM and CNNs are presented and compared. To make valid comparisons, the two models are evaluated using well-accepted AES evaluation criteria (Williamson, Xi, & Breyer, 2012) and compared based on the type, length, and scoring rubrics of the essay sets.

Part One: Model Specifications for the Best Performing Models

In the Coh-Metrix + SVM model, I experimented with several different types of kernels and regularization parameters to produce the best results. Previous research demonstrated good prediction performance using the polynomial kernel SVM (Latifi, 2016). Therefore, we chose the same model as one the options while producing several other models with different hyper-parameter selections to locate the best model. More information can be found in Table 6.

Table 6. *Final Selection of Hyper-parameters of the best SVM models*

Parameter Name	Parameter Value
Degree, (Slack variable; C)	1
Kernel	Polynomial
Regularization parameter	1.0
Epochs	20
Batch size	128

In the CNNs model, some of the hyper-parameters that produced the best accuracy varied across the essay sets and the final model was selected based on comparing their performances given different hyper-parameters. More specifically, the embedding dimension, number of neurons in the dense layer, epoch size, and the batch size remained consistent, while dropout rated varied in essay set 2a and 8. Also, for essay set 2b, 3, and 8, the sigmoid activation function was used instead of ReLU in the last layer. In addition, for the last essay set instead of using 3 convolutional layers, one additional convolutional layer with the sigmoid activation function was added for more accurate prediction results. More information about hyper-parameters can be found in Table 7.

Table 7. *Final Selection of Hyper-parameters of the best CNNs models*

Layer	**Parameter**	**Essay set**								
	Name	**1**	**2a**	**2b**	**3**	**4**	**5**	**6**	**7**	**8**
Embedding	Dimension	300	300	300	300	300	300	300	300	300
Dropout	Rate	0.5	0.2	0.5	0.5	0.5	0.5	0.5	0.5	0.3
CNN	Filters	50 100 200	50 100 200	50 100 200	50 100 200	50 100 200	50 100 200	50 100 200	50 100 200	50 100 200 200
	Kernel size	5	5	5	5	5	5	5	5	5
	Activation	ReLU	ReLU	Sigmoid	Sigmoid	ReLU	ReLU	ReLU	ReLU	Sigmoid
Dense	Neurons	100	100	100	100	100	100	100	100	100
Model	Epoch	15	15	15	15	20	20	20	20	30
Compile	Batch size	128	128	128	128	128	128	128	128	128

Part Two: Model Prediction Results and Agreement Measures

According to Williamson, Xi, and Breyer (2012), QWK can be evaluated based on two criteria-based guidelines. First, the score should be bigger than 0.70, so that it accounts for at least more than half of the variance in human-rated scores. Second, the absolute difference between the human-human and human-machine agreement should not be bigger than 0.10.

Table 8 presents the prediction outcomes of the proposed models using a quadratic weighted kappa (QWK) score. In terms of the conformity to the first criterion, five out of nine essay sets had QWK exceeding 0.70 in the Coh-Metrix + SVM model, while six out of eight essay sets showed QWK exceeding 0.70 in the CNNs model. Moreover, for the sets where both the models could not conform to the criterion, still, the CNNs model produced very close to the criterion score.

Table 8. *Model Comparison using Quadratic Weighted Kappa score*

Model	**Essay set**								
	1	**2a**	**2b**	**3**	**4**	**5**	**6**	**7**	**8**
Coh-Metrix + SVM	**0.81**	0.60	0.54	0.69	**0.70**	**0.76**	**0.75**	**0.71**	0.62
CNNs	**0.80**	0.69	0.68	**0.77**	**0.76**	**0.82**	**0.78**	**0.74**	0.48
Human Raters	0.71	0.78	0.72	0.81	0.86	0.74	0.77	0.68	0.63

Notes. The bold numbers indicate that the score has satisfied the first criterion (QWK > 0.70).

Table 9 presents the outcome of the agreements between the human-raters and the proposed models based on quadratic weighted kappa (QWK) score. In terms of the conformity to the second criterion, five out of nine essay sets showed QWK absolute difference smaller than 0.10 in the Coh-Metrix + SVM model, while the CNNs model conforms to criterion on all of the essay sets except set 8. Also, for essay set 8, the absolute difference was not tremendously large.

In short, the CNNs model showed better performance based on the two-criterion based guidelines. Moreover, the CNNs model could obtain a very high overall average QWK score, 0.72 that is better than the Coh-Metrix + SVM model, 0.68 and more comparable to the average human-rater's agreement, 0.74.

Table 9. *Model Comparison using the Absolute Difference with the Human-rater Agreement*

Model	**Essay set**								
	1	**2a**	**2b**	**3**	**4**	**5**	**6**	**7**	**8**
Coh-Metrix + SVM	**0.10**	0.18	0.18	0.12	0.16	**0.02**	**0.02**	**0.03**	**0.01**
CNNs	**0.09**	**0.09**	**0.04**	**0.04**	**0.10**	**0.08**	**0.01**	**0.06**	0.15

Notes. The bold numbers indicate that the score has satisfied the second criterion (Absolute difference ≤ 0.10).

In addition, Table 10, 11 and 12 present the outcomes based on different circumstances. First, when compared based on the types of essay they scored, the Coh-Metrix + SVM model showed its strength in scoring the narrative type essays, while the CNNs showed comparable results to the human-raters in scoring persuasive and source-dependent essay types. Second, when compared based on the average length of essay they scored, the Coh-Metrix + SVM model produced more accurate results in scoring essay sets with an average of 650 words, while the CNNs showed the best accuracy in scoring essay sets that ranged from 150, 250, and 350 words. Last, when compared based on the scoring rubric applied, the CNNs model consistently showed better accuracy than the Coh-Metrix + SVM model.

Table 10. *Model Results Comparison Based on the Types of Essays*

Model	Types of Essay		
	Persuasive (1, 2a)	**Source-dependent** (3, 4, 5,6)	**Narrative** (7,8)
Coh-Metrix + SVM	0.71	0.73	0.67
CNNs	0.74	0.78	0.61
Human Raters	0.75	0.79	0.66

Table 11. *Model Results Comparison Based on the Average Length of Essay*

Model	Average Essay length (words)			
	350 (1, 2a, 2b)	**150** (3, 4, 5, 6)	**250** (7)	**650** (8)
Coh-Metrix + SVM	0.65	0.73	0.67	0.62
CNNs	0.72	0.78	0.74	0.48
Human Raters	0.74	0.79	0.68	0.63

Table 12. *Model Results Comparison Based on the Type of Scoring Rubric*

Model	Types of Scoring Rubric	
	Writing Application (1, 2a, 3, 4, 5, 6)	**Language Convention** (2b, 7, 8)
Coh-Metrix + SVM	0.72	0.62
CNNs	0.77	0.64
Human Raters	0.78	0.68

Chapter Summary

The results of the current study demonstrated that the CNNs model outperformed the Coh-Metrix + SVM model based on the two-criterion based guidelines and produce a higher average QWK score. Moreover, the CNNs model could produce better results in significantly more categories when compared based on different circumstances (i.e., types of essay, average length, and the scoring rubric).

Chapter 5: Discussion

Unlike human raters who process essay texts cognitively using linguistics and writing knowledge to analyze not only the content but also various styles of writing and creativity, the automated essay scoring (AES) frameworks grade papers by either utilizing pre-defined features and learning the patterns using the model or extracting and learning the features and patterns simultaneously to make accurate predictions (Zhang, 2013). Recently, AES systems have evolved with two frameworks: a machine learning AES with carefully hand-engineered features and a deep learning AES algorithm. The hand-engineered features often refer to the surface-level features as well as deep language features. Previous studies have demonstrated the power of utilizing deep language features for accurate prediction in AES (Latifi, 2016). Also, deep learning AES frameworks have gained popularity because they provide the important benefit of learning and extracting features while learning the patterns in a parallel manner.

The purpose of the present study was to investigate and compare the behaviours and performance of two AES approaches. More specifically, I implemented a machine learning AES using SVM with language features extracted from Coh-Metrix and a deep learning AES system using the CNNs approach. According to the previous studies, only about 30% of errors made in essays are currently detected using a state-of-the-art error detection system, and a large number of missed errors are long-distance errors such as sequential or time-series information (Ng et el., 2014). Therefore, I selected algorithms that could effectively capture sequential information. For example, Coh-Metrix features were originally designed to investigate the cohesion and coherence of the text because previous studies assert that CNNs could effectively model sentence level coherence (Dong, Zhang, & Yang, 2017). The current study was designed to address following questions:

(1) Do the deep learning AES frameworks produce more accurate prediction results compared to the machine learning AES systems with deep language features? I developed two AES systems, one based on a machine learning classification algorithm called support vector machines (SVMs) in conjunction with language features extracted using Coh-Metrix and the second one based on a deep learning algorithm called convolutional neural networks (CNNs). Using a dataset from the previous essay scoring competition (i.e., the ASAP), the two models were trained and compared based on their prediction accuracy. Coh-Metrix features were acquired by requesting a text analysis service online from the Coh-Metrix website. The performance of the frameworks was then evaluated using an agreement score – quadratic weighted kappa score. When comparing the performance, I also adopted standard-based criteria by Williamson, Xi, and Breyer (2012) to make more thorough and comprehensive comparisons. The results suggested that the deep learning algorithm could perform very similarly to human raters and outperform a machine learning algorithm.

(2) How does the model behaviour change in particular circumstances (i.e., type of rubric used for scoring, the length of essay, type of essay prompt)? The models were further evaluated according to several additional criteria to better understand the strengths and weaknesses of the model solutions. First, their QWK scores were compared based on the type of essay set attempted to score. Second, prediction results were compared based on the average length of essay set. Third, the results were compared based on the types of scoring rubric. For example, three different types of essay prompts were given, which are persuasive, source-dependent, and narrative essays. Also, the average length of the essays varied across the prompts. For example, there were 350 words in essay set 1 and 2, 150 words in essay set 3, 4, 5, and 6, 250 words in essay set 7, and 650 words in essay set 8 on average. Scoring longer responses

accurately typically requires more learning in the models. Also, deep learning algorithms often require larger sample sizes than machine learning algorithms to train the networks to recognize patterns (Cho, Lee, Shin, Choy, & Do, 2015). Therefore, I attempted to better understand if either model performs better under these circumstances. The results suggested that the deep learning algorithm could perform better in persuasive and source-dependent essay prompts while the Coh-Metrix + SVM model showed better performance in the narrative essay prompt. Also, in terms of the average essay length, the CNNs model performed better than the Coh-Metrix + SVM model except in essay 8, where the average length was 650. Last, when compared based on the types of rubric, the CNNs model consistently showed better performance.

Conclusion

The results demonstrated the effectiveness of using a deep learning algorithm without any additional help from feature engineering to produce accurate prediction results. In short, the CNNs model achieved higher QWK scores in most of the essay sets except set 8. It achieved an average QWK score of 0.72 while the Coh-Metrix + SVM model achieved 0.68. CNNs produced better performance compared to the machine learning approach with deep language features. I also noticed several surprising benefits of the proposed CNNs model.

To begin, AES frameworks with deep learning algorithms do not require any feature engineering, which often entails extensive knowledge in language conventions and linguistics. Even though Coh-Metrix features can provide some bountiful and adequate theoretical supports to understand and interpret the prediction mechanism, based on my experience, obtaining the extracted features still required a significant amount of time. Also, it was stated that the quality of the extracted Coh-Metrix features could significantly depend on the quality of data pre-processing before submitting corpus for extraction. Therefore, it is important to consider the reliability and the subjectivity of the extracted features.

Also, the presented CNNs model could produce results that are comparable to previously proposed models despite the fact that it required a relatively simple embedding technique and architecture. Even though it is hard to make a direct comparison with other results due to the difference in testing samples, our CNNs model could achieve comparable results with state-of-the-art performance in the current dataset except in essay set 8. When compared to previous research where deep learning algorithms were implemented (Dong, Zhang, & Yang, 2017; Zhao et al., 2017), our model still achieved comparable accuracy despite its simplicity. Further, when compared to the models where Coh-Metrix features were used with a machine learning algorithm, our model consistently achieved better results except in essay set 8. More information can be found in Table 13.

Table 13. *Model Results Comparison with Previous Research and State-of-the-art*

Model	**Essay set**									**Ave.**
	1	**2a**	**2b**	**3**	**4**	**5**	**6**	**7**	**8**	
CNNs	0.80	0.69	0.68	0.77	0.76	0.82	0.78	0.74	0.48	0.73
The-State of-Art	0.77	0.70	0.66	0.71	0.77	0.80	0.74	0.76	0.67	0.73
LSTM-CNN-ATT	0.82	0.68	-	0.67	0.81	0.80	0.81	0.80	0.71	0.76
Memory Networks	0.83	0.72	-	0.72	0.82	0.83	0.83	0.79	0.68	0.78
Coh-Metrix + SMO	0.76	0.64	0.64	0.69	0.73	0.79	0.66	0.72	0.59	0.68
Coh-Metrix + RF	0.71	0.59	0.60	0.67	0.72	0.78	0.70	0.71	0.42	0.65

Notes. LSTM-CNN-att (Dong, Zhang, & Yang, 2017); MN (Zhao et al., 2017);
Coh-Metrix +SMO, Coh-Metrix+ RF (Latifi, 2016); The-State-of-Art (Shermis, 2014)

Limitations of the Study and the Directions for Future Research

Even though the study was designed and structured to minimize potential error with results and further interpretations, the following two limitations should be carefully considered for future research: First, the performance comparison based on particular circumstances (e.g., essay average length, scoring rubric, and essay type) could not be taken at face values. For example, when compared based on the type, average length, and rubrics of essay, the CNNs model overall showed better performance, while the second model achieved better accuracy in predicting scores for narrative essays and the lengthiest essay set. However, it is could be due to the poor results in essay set 8, where not enough sample size was provided for a deep learning algorithm to train and recognize generalizable patterns. As the comparisons were made to rather understand the behaviours in a more comprehensive manner, further research is required to investigate whether and how these circumstances significantly affect the behaviours of deep learning AES frameworks.

Second, compared to the results of state-of-the-art AES frameworks, our best model (or CNNs model) could not produce comparable accuracy in essay set 8. Deep learning often requires a relatively large dataset for generalizable learning and only 585 samples were used for model training. Moreover, to produce more accurate results, deep learning often requires a more complex architecture for a small sample size-based learning. Some of the previous studies could produce better results (e.g., QWK 0.71) in essay set 8 using a more complex model structure with specialized embeddings (e.g., SSWE, ATT; Dong, Zhang, & Yang, 2017; Zhao et al., 2017). Therefore, it will be important for future research to investigate whether a relatively simple deep learning architecture could still provide generalizable results with a small training set.

References

ACARA NASOP research team. (2015). An evaluation of automated scoring of NAPLAN persuasive writing. *Acara Australian Curriculum Assessment and Reporting Authority, 30.*

Alikaniotis, D., Yannakoudakis, H., & Rei, M. (2016). Automatic text scoring using neural networks. arXiv preprint arXiv:1606.04289. http://dx.doi.org/10.18653/v1/P16-1068

Attali, Y. (2013). Validity and Reliability of Automated Essay Scoring. In M.D. Shermis & J.C. Burstein (Eds.), Handbook of Automated Essay Evaluation: current application and new directions (pp. 181-198). New York: Psychology Press. http://dx.doi.org/10.4324/9780203122761.ch11

Attali, Y., & Burstein, J. (2004). Automated Essay Scoring with E-raters ® V. 2.0. ETS Research Report Series, 2004(2).

Bird, S., Klein, E., & Loper, E. (2009). Natural language processing with Python: analyzing text with the natural language toolkit. Sebastopol, CA: O'Reilly Media.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2), 121-167.

Cain, K., & Nash, H. M. (2011). The influence of connectives on young readers' processing and comprehension of text. Journal of Educational Psychology, 103(2), 429.

Chen, H., & He, B. (2013). Automated essay scoring by maximizing human-machine agreement. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing (pp. 1741-1752).

Cho, J., Lee, K., Shin, E., Choy, G., & Do, S. (2015). How much data is needed to train a medical image deep learning system to achieve necessary high accuracy? arXiv preprint arXiv:1511.06348.

Chung, G. K. W. K., & Baker, E. L. (2003). Issues in the reliability and validity of automated scoring of constructed responses. In M. D. Shermis, & J. C. Burstein (Eds.), Automated essay scoring: A cross disciplinary perspective (pp. 23–40). Mahwah, NJ: Lawrence Erlbaum Associates.

Cohen, J. (1960). A coefficient of agreement for nominal scales. Educational and psychological measurement, 20(1), 37-46.

Dong, F., Zhang, Y., & Yang, J. (2017). Attention-based Recurrent Convolutional Neural Network for Automatic Essay Scoring. In Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017) (pp. 153-162).

Gierl, M. J., Latifi, S., Lai, H., Boulais, A. P., & Champlain, A. (2014). Automated essay scoring and the future of educational assessment in medical education. Medical education, 48(10), 950-962.

Graesser, A. C., & McNamara, D. S. (2011). Computational analyses of multilevel discourse comprehension. Topics in cognitive science, 3(2), 371-398.

Graesser, A. C., McNamara, D. S., & Kulikowich, J. M. (2011). Coh-Metrix: Providing multilevel analyses of text characteristics. Educational researcher, 40(5), 223-234.

Joachims, T. (1998, April). Text categorization with support vector machines: Learning with many relevant features. In European conference on machine learning (pp. 137-142). Springer, Berlin, Heidelberg.

Kaplan, R. M., Wolff, S., Burstein, J. C., Lu, C., Rock, D., & Kaplan, B. (1998). Scoring essays automatically using surface features. ETS Research Report Series, 1998(2).

Kim, Y. (2014). Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882.

Klare, G. R. (1974). Assessing readability. Reading research quarterly, 62-102.

Landauer, T., McNamara, D. S., Dennis, S., & Kintsch, W. (2007). LSA: A road to meaning. Mahwah, NI: Lawrence Erlbaum Associates.

Latifi, S. M. F. (2016). Development and Validation of an Automated Essay Scoring Framework by Integrating Deep Features of English Language (Doctoral dissertation, University of Alberta).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278-2324.

Lee, H., Grosse, R., Ranganath, R., & Ng, A. Y. (2009, June). Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the 26th annual international conference on machine learning (pp. 609-616). ACM.

McNamara, D. S., Crossley, S. A., Roscoe, R. D., Allen, L. K., & Dai, J. (2015). A hierarchical classification approach to automated essay scoring. Assessing Writing, 23, 35-59.

McNamara, D. S., Graesser, A. C., McCarthy, P. M., & Cai, Z. (2014). Automated evaluation of text and discourse with Coh-Metrix. Cambridge University Press.

Mikolov, T., Karafiát, M., Burget, L., Černocký, J., & Khudanpur, S. (2010). Recurrent neural network based language model. In Eleventh Annual Conference of the International Speech Communication Association.

Mikolov, T., Kombrink, S., Burget, L., Černocký, J., & Khudanpur, S. (2011, May). Extensions of recurrent neural network language model. In Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on (pp. 5528-5531). IEEE.

Ng, H. T., Wu, S. M., Briscoe, T., Hadiwinoto, C., Susanto, R. H., & Bryant, C. (2014). The CoNLL-2014 shared task on grammatical error correction. In Proceedings of the Eighteenth Conference on Computational Natural Language Learning: Shared Task (pp. 1-14).

Nguyen, H., & Dery, L. Neural Networks for Automated Essay Grading.

Page, E. B. (1967). Grading essays by computer: Progress report. Proceedings of the 1966 Invitational Conference on Testing (pp. 87-100). Princeton, NJ: Educational Testing Service.

Page, E. B. (1994). Computer grading of student prose, using modern concepts and software. The Journal of experimental education, 62(2), 127-142.

Pennington, J., Socher, R., & Manning, C. (2014). Glove: Global vectors for word representation. In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Perelman, L. (2014). When "the state of the art" is counting words. Assessing Writing, 21, 104-111.

Sharma, S. (2017, September 06). Activation Functions: Neural Networks – Toward Data Science. Retrieved from https://towardsdatascience.com/activation-functions-neuralnetworks-1cbd9f8d91d6

Shermis, M. D. (2010). Automated essay scoring in a high stakes testing environment. In Innovative Assessment for the 21st Century (pp. 167-185). Springer, Boston, MA.

Shermis, M. D. (2014). State-of-the-art automated essay scoring: Competition, results, and future directions from a United States demonstration. Assessing Writing, 20, 53-76.

Siegel, S. C., & Castellan, J. NJ (1988). Nonparametric statistics for the behavioural sciences. New York, McGraw-Hill.

Socher, R., Bauer, J., & Manning, C. D. (2013). Parsing with compositional vector grammars. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 455-465).

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In Proceedings of the 2013 conference on empirical methods in natural language processing (pp. 1631-1642).

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. The Journal of Machine Learning Research, 15(1), 1929-1958.

Taghipour, K., & Ng, H. T. (2016). A neural approach to automated essay scoring. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (pp. 1882-1891).

Tong, S., & Koller, D. (2001). Support vector machine active learning with applications to text classification. Journal of machine learning research, 2(Nov), 45-66.

Williams, R. J., & Zipser, D. (1989). A learning algorithm for continually running fully recurrent neural networks. Neural computation, 1(2), 270-280.

Williamson, D. M., Xi, X., & Breyer, F. J. (2012). A framework for evaluation and use of automated scoring. Educational measurement: issues and practice, 31(1), 2-13.

Xu, W., & Liu, M. (2016). Using Coh-Metrix to Analyze Chinese ESL Learners' Writing. International Journal of Learning, Teaching and Educational Research, 15(5), 16-26.

Yannakoudakis, H., Briscoe, T., & Medlock, B. (2011, June). 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-Volume 1 (pp. 180-189). Association for Computational Linguistics.

Zaidi, A. H. (2016). Neural Sequence Modelling for Automated Essay Scoring.

Zhang, L., Xiong, X., Zhao, S., Botelho, A., & Heffernan, N. T. (2017, April). Incorporating rich features into deep knowledge tracing. In Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale (pp. 169-172). ACM.

Zhang, M. (2013). Contrasting automated and human scoring of essays. R & D Connections, 21(2).

Zhao, S., Zhang, Y., Xiong, X., Botelho, A., & Heffernan, N. (2017, April). A Memory-Augmented Neural Model for Automated Grading. In Proceedings of the Fourth (2017) ACM Conference on Learning@ Scale (pp. 189-192). ACM.

Appendix A: Essay Sets

A1: Essay Set #1

Prompt

More and more people use computers, but not everyone agrees that this benefits society. Those who support advances in technology believe that computers have a positive effect on people. They teach hand-eye coordination, give people the ability to learn about faraway places and people, and even allow people to talk online with other people. Others have different ideas. Some experts are concerned that people are spending too much time on their computers and less time exercising, enjoying nature, and interacting with family and friends.

Write a letter to your local newspaper in which you state your opinion on the effects computers have on people. Persuade the readers to agree with you.

A2: Essay Set #2

Prompt
Censorship in the Libraries "All of us can think of a book that we hope none of our children or any other children have taken off the shelf. But if I have the right to remove that book from the shelf -- that work I abhor -- then you also have exactly the same right and so does everyone else. And then we have no books left on the shelf for any of us." --Katherine Paterson, Author Write a persuasive essay to a newspaper reflecting your vies on censorship in libraries. Do you believe that certain materials, such as books, music, movies, magazines, etc., should be removed from the shelves if they are found offensive? Support your position with convincing arguments from your own experience, observations, and/or reading.

A3: Essay Set #3

Source

ROUGH ROAD AHEAD: Do Not Exceed Posted Speed Limit

by Joe Kurmaskie

FORGET THAT OLD SAYING ABOUT NEVER taking candy from strangers. No, a better piece of advice for the solo cyclist would be, “Never accept travel advice from a collection of old-timers who haven’t left the confines of their porches since Carter was in office.” It’s not that a group of old guys doesn’t know the terrain. With age comes wisdom and all that, but the world is a fluid place. Things change.

At a reservoir campground outside of Lodi, California, I enjoyed the serenity of an early-summer evening and some lively conversation with these old codgers. What I shouldn’t have done was let them have a peek at my map. Like a foolish youth, the next morning I followed their advice and launched out at first light along a “shortcut” that was to slice away hours from my ride to Yosemite National Park.

They’d sounded so sure of themselves when pointing out landmarks and spouting off towns I would come to along this breezy jaunt. Things began well enough. I rode into the morning with strong legs and a smile on my face. About forty miles into the pedal, I arrived at the first “town.” This place might have been a thriving little spot at one time—say, before the last world war—but on that morning it fit the traditional definition of a ghost town. I chuckled, checked my water supply, and moved on. The sun was beginning to beat down, but I barely noticed it. The cool pines and rushing rivers of Yosemite had my name written all over them.

Twenty miles up the road, I came to a fork of sorts. One ramshackle shed, several rusty pumps, and a corral that couldn’t hold in the lamest mule greeted me. This sight was troubling. I had been hitting my water bottles pretty regularly, and I was traveling through the high deserts of California in June.

I got down on my hands and knees, working the handle of the rusted water pump with all my strength. A tarlike substance oozed out, followed by brackish water feeling somewhere in the neighborhood of two hundred degrees. I pumped that handle for several minutes, but the water wouldn’t cool down. It didn’t matter. When I tried a drop or two, it had the flavor of battery acid.

The old guys had sworn the next town was only eighteen miles down the road. I could make that! I would conserve my water and go inward for an hour or so—a test of my inner spirit.

Not two miles into this next section of the ride, I noticed the terrain changing. Flat road was replaced by short, rolling hills. After I had crested the first few of these, a large highway sign jumped out at me. It read: ROUGH ROAD AHEAD: DO NOT EXCEED POSTED SPEED LIMIT.

The speed limit was 55 mph. I was doing a water-depleting 12 mph. Sometimes life can feel so cruel.

I toiled on. At some point, tumbleweeds crossed my path and a ridiculously large snake—it really did look like a diamondback—blocked the majority of the pavement in front of me. I eased past, trying to keep my balance in my dehydrated state.

The water bottles contained only a few tantalizing sips. Wide rings of dried sweat circled my shirt, and the growing realization that I could drop from heatstroke on a gorgeous day in June simply because I listened to some gentlemen who hadn't been off their porch in decades, caused me to laugh.

It was a sad, hopeless laugh, mind you, but at least I still had the energy to feel sorry for myself. There was no one in sight, not a building, car, or structure of any kind. I began breaking the ride down into distances I could see on the horizon, telling myself that if I could make it that far, I'd be fine.

Over one long, crippling hill, a building came into view. I wiped the sweat from my eyes to make sure it wasn't a mirage, and tried not to get too excited. With what I believed was my last burst of energy, I maneuvered down the hill.

In an ironic twist that should please all sadists reading this, the building—abandoned years earlier, by the looks of it—had been a Welch's Grape Juice factory and bottling plant. A sandblasted picture of a young boy pouring a refreshing glass of juice into his mouth could still be seen.

I hung my head.

That smoky blues tune "Summertime" rattled around in the dry honeycombs of my deteriorating brain.

I got back on the bike, but not before I gathered up a few pebbles and stuck them in my mouth. I'd read once that sucking on stones helps take your mind off thirst by allowing what spit you have left to circulate. With any luck I'd hit a bump and lodge one in my throat.

It didn't really matter. I was going to die and the birds would pick me clean, leaving only some expensive outdoor gear and a diary with the last entry in praise of old men, their wisdom, and their keen sense of direction. I made a mental note to change that paragraph if it looked like I was going to lose consciousness for the last time.

Somehow, I climbed away from the abandoned factory of juices and dreams, slowly gaining elevation while losing hope. Then, as easily as rounding a bend, my troubles, thirst, and fear were all behind me.

GARY AND WILBER'S FISH CAMP—IF YOU WANT BAIT FOR THE BIG ONES, WE'RE YOUR BEST BET!

"And the only bet," I remember thinking.

As I stumbled into a rather modern bathroom and drank deeply from the sink, I had an overwhelming urge to seek out Gary and Wilber, kiss them, and buy some bait—any bait, even though I didn't own a rod or reel.

An old guy sitting in a chair under some shade nodded in my direction. Cool water dripped from my head as I slumped against the wall beside him.

"Where you headed in such a hurry?"

"Yosemite," I whispered.

"Know the best way to get there?"

I watched him from the corner of my eye for a long moment. He was even older than the group I'd listened to in Lodi.

"Yes, sir! I own a very good map."

And I promised myself right then that I'd always stick to it in the future.

"Rough Road Ahead" by Joe Kurmaskie, from Metal Cowboy, copyright © 1999 Joe Kurmaskie.

Prompt

Write a response that explains how the features of the setting affect the cyclist. In your response, include examples from the essay that support your conclusion.

A4: Essay Set #4

Source

Winter Hibiscus by Minfong Ho

Saeng, a teenage girl, and her family have moved to the United States from Vietnam. As Saeng walks home after failing her driver's test, she sees a familiar plant. Later, she goes to a florist shop to see if the plant can be purchased.

It was like walking into another world. A hot, moist world exploding with greenery. Huge flat leaves, delicate wisps of tendrils, ferns and fronds and vines of all shades and shapes grew in seemingly random profusion.

"Over there, in the corner, the hibiscus. Is that what you mean?" The florist pointed at a leafy potted plant by the corner.

There, in a shaft of the wan afternoon sunlight, was a single blood-red blossom, its five petals splayed back to reveal a long stamen tipped with yellow pollen. Saeng felt a shock of recognition so intense, it was almost visceral.1

"Saebba," Saeng whispered.

A saebba hedge, tall and lush, had surrounded their garden, its lush green leaves dotted with vermilion flowers. And sometimes after a monsoon rain, a blossom or two would have blown into the well, so that when she drew the well water, she would find a red blossom floating in the bucket.

Slowly, Saeng walked down the narrow aisle toward the hibiscus. Orchids, lanna bushes, oleanders, elephant ear begonias, and bougainvillea vines surrounded her. Plants that she had not even realized she had known but had forgotten drew her back into her childhood world.

When she got to the hibiscus, she reached out and touched a petal gently. It felt smooth and cool, with a hint of velvet toward the center—just as she had known it would feel.

And beside it was yet another old friend, a small shrub with waxy leaves and dainty flowers with purplish petals and white centers. "Madagascar periwinkle," its tag announced. How strange to see it in a pot, Saeng thought. Back home it just grew wild, jutting out from the cracks in brick walls or between tiled roofs.

And that rich, sweet scent—that was familiar, too. Saeng scanned the greenery around her and found a tall, gangly plant with exquisite little white blossoms on it. "Dok Malik," she said, savoring the feel of the word on her tongue, even as she silently noted the English name on its tag, “jasmine.”

One of the blossoms had fallen off, and carefully Saeng picked it up and smelled it. She closed her eyes and breathed in, deeply. The familiar fragrance filled her lungs, and Saeng could almost feel the light strands of her grandmother’s long gray hair, freshly washed, as she combed it out with the fine-toothed buffalo-horn comb. And when the sun had dried it, Saeng would help the gnarled old fingers knot the hair into a bun, then slip a dok Malik bud into it.

Saeng looked at the white bud in her hand now, small and fragile. Gently, she closed her palm around it and held it tight. That, at least, she could hold on to. But where was the fine-toothed comb? The hibiscus hedge? The well? Her gentle grandmother?

A wave of loss so deep and strong that it stung Saeng’s eyes now swept over her. A blink, a channel switch, a boat ride into the night, and it was all gone. Irretrievably, irrevocably gone. And in the warm moist shelter of the greenhouse, Saeng broke down and wept.

It was already dusk when Saeng reached home. The wind was blowing harder, tearing off the last remnants of green in the chicory weeds that were growing out of the cracks in the sidewalk. As if oblivious to the cold, her mother was still out in the vegetable garden, digging up the last of the onions with a rusty trowel. She did not see Saeng until the girl had quietly knelt down next to her.

Her smile of welcome warmed Saeng. “Ghup ma laio le? You’re back?” she said cheerfully. “Goodness, it’s past five. What took you so long? How did it go? Did you—?” Then she noticed the potted plant that Saeng was holding, its leaves quivering in the wind.

Mrs. Panouvong uttered a small cry of surprise and delight. “Dok faeng-noi!” she said. “Where did you get it?”

“I bought it,” Saeng answered, dreading her mother’s next question.

“How much?”

For answer Saeng handed her mother some coins.

“That’s all?” Mrs. Panouvong said, appalled, “Oh, but I forgot! You and the Lambert boy ate Bee-Maags . . . .”

“No, we didn’t, Mother,” Saeng said.

“Then what else—?”

“Nothing else. I paid over nineteen dollars for it.”

“You what?” Her mother stared at her incredulously. “But how could you? All the seeds for this vegetable garden didn’t cost that much! You know how much we—” She paused, as she noticed the tearstains on her daughter’s cheeks and her puffy eyes.

“What happened?” she asked, more gently.

“I—I failed the test,” Saeng said.

For a long moment Mrs. Panouvong said nothing. Saeng did not dare look her mother in the eye. Instead, she stared at the hibiscus plant and nervously tore off a leaf, shredding it to bits.

Her mother reached out and brushed the fragments of green off Saeng’s hands. “It’s a beautiful plant, this dok faeng-noi,” she finally said. “I’m glad you got it.”

“It’s—it’s not a real one,” Saeng mumbled.

“I mean, not like the kind we had at—at—” She found that she was still too shaky to say the words at home, lest she burst into tears again. “Not like the kind we had before,” she said.

“I know,” her mother said quietly. “I’ve seen this kind blooming along the lake. Its flowers aren’t as pretty, but it’s strong enough to make it through the cold months here, this winter hibiscus. That’s what matters.”

She tipped the pot and deftly eased the ball of soil out, balancing the rest of the plant in her other hand. “Look how root-bound it is, poor thing,” she said. “Let’s plant it, right now.”

She went over to the corner of the vegetable patch and started to dig a hole in the ground. The soil was cold and hard, and she had trouble thrusting the shovel into it. Wisps of her gray hair trailed out in the breeze, and her slight frown deepened the wrinkles around her eyes. There was a frail, wiry beauty to her that touched Saeng deeply.

“Here, let me help, Mother,” she offered, getting up and taking the shovel away from her.

Mrs. Panouvong made no resistance. “I’ll bring in the hot peppers and bitter melons, then, and start dinner. How would you like an omelet with slices of the bitter melon?”

“I’d love it,” Saeng said.

Left alone in the garden, Saeng dug out a hole and carefully lowered the “winter hibiscus” into it. She could hear the sounds of cooking from the kitchen now, the beating of eggs against a bowl, the sizzle of hot oil in the pan. The pungent smell of bitter melon wafted out, and Saeng’s mouth watered. It was a cultivated taste, she had discovered—none of her classmates or friends, not even Mrs. Lambert, liked it—this sharp, bitter melon that left a golden aftertaste on the tongue. But she had grown up eating it and, she admitted to herself, much preferred it to a Big Mac.

The “winter hibiscus” was in the ground now, and Saeng tamped down the soil around it. Overhead, a flock of Canada geese flew by, their faint honks clear and—yes—familiar to Saeng now. Almost reluctantly, she realized that many of the things that she had thought of as strange before had become, through the quiet repetition of season upon season, almost familiar to her now. Like the geese. She lifted her head and watched as their distinctive V was etched against the evening sky, slowly fading into the distance.

When they come back, Saeng vowed silently to herself, in the spring, when the snows melt and the geese return and this hibiscus is budding, then I will take that test again.

“Winter Hibiscus” by Minfong Ho, copyright © 1993 by Minfong Ho, from Join In, Multiethnic Short Stories, by Donald R. Gallo, ed.

Prompt

Read the last paragraph of the story.

"When they come back, Saeng vowed silently to herself, in the spring, when the snows melt and the geese return and this hibiscus is budding, then I will take that test again."

Write a response that explains why the author concludes the story with this paragraph. In your response, include details and examples from the story that support your ideas.

A5: Essay Set #5

Source

Narciso Rodriguez

from *Home: The Blueprints of Our Lives*

My parents, originally from Cuba, arrived in the United States in 1956. After living for a year in a furnished one-room apartment, twenty-one-year-old Rawedia Maria and twenty-seven-year-old Narciso Rodriguez, Sr., could afford to move into a modest, three-room apartment I would soon call home.

In 1961, I was born into this simple house, situated in a two-family, blond-brick building in the Ironbound section of Newark, New Jersey. Within its walls, my young parents created our traditional Cuban home, the very heart of which was the kitchen. My parents both shared cooking duties and unwittingly passed on to me their rich culinary skills and a love of cooking that is still with me today (and for which I am eternally grateful). Passionate Cuban music (which I adore to this day) filled the air, mixing with the aromas of the kitchen. Here, the innocence of childhood, the congregation of family and friends, and endless celebrations that encompassed both, formed the backdrop to life in our warm home.

Growing up in this environment instilled in me a great sense that "family" had nothing to do with being a blood relative. Quite the contrary, our neighborhood was made up of mostly Spanish, Cuban, and Italian immigrants at a time when overt racism was the norm and segregation prevailed in the United States. In our neighborhood, despite customs elsewhere, all of these cultures came together in great solidarity and friendship. It was a close-knit community of honest, hardworking immigrants who extended a hand to people who, while not necessarily their own kind, were clearly in need.

Our landlord and his daughter, Alegria (my babysitter and first friend), lived above us, and Alegria graced our kitchen table for meals more often than not. Also at the table were Sergio and Edelmira, my surrogate grandparents who lived in the basement apartment. (I would not know my "real" grandparents, Narciso the Elder and Consuelo, until 1970 when they were allowed to leave Cuba.) My aunts Bertha and Juanita and my cousins Arnold, Maria, and Rosemary also all lived nearby and regularly joined us at our table. Countless extended family members came and went — and there was often someone staying with us temporarily until they were able to get back on their feet. My parents always kept their arms and their door open to the many people we considered family, knowing that they would do the same for us.

My mother and father had come to this country with such courage, without any knowledge of the language or the culture. They came selflessly, as many immigrants do, to give their children a better life, even though it meant leaving behind their families, friends, and careers in the country they loved. They struggled both personally and financially, braving the harsh northern winters while yearning for their native tropics and facing cultural hardships. The barriers to work were strong and high, and my parents both had to accept that they might not be able to find the kind of jobs they deserved. In Cuba, Narciso, Sr., had worked in a laboratory and Rawedia Maria had studied chemical engineering. In the United States, they had to start their lives over entirely, taking whatever work they could find. The faith that this struggle would lead them and their children to better times drove them to endure these hard times.

I will always be grateful to my parents for their love and sacrifice. I've often told them that what they did was a much more courageous thing than I could have ever done. I've often told them of my admiration for their strength and perseverance, and I've thanked them repeatedly. But, in reality, there is no way to express my gratitude for the spirit of generosity impressed upon me at such an early age and the demonstration of how important family and friends are. These are two lessons that my parents did not just tell me. They showed me with their lives, and these teachings have been the basis of my life.

It was in this simple house that my parents welcomed other refugees to celebrate their arrival to this country and where I celebrated my first birthdays. It was in the warmth of the kitchen in this humble house where a Cuban feast (albeit a frugal Cuban feast) always filled the air with not just scent and music but life and love. It was here where I learned the real definition of "family." And for this, I will never forget that house or its gracious neighborhood or the many things I learned there about how to love. I will never forget how my parents turned this simple house into a home.

— Narciso Rodriguez, Fashion designer

Hometown: Newark, New Jersey

"Narciso Rodriguez" by Narciso Rodriguez, from Home: The Blueprints of Our Lives.

Copyright © 2006 by John Edwards.

Prompt

Describe the mood created by the author in the memoir. Support your answer with relevant and specific information from the memoir.

A6: Essay Set #6

Source

The Mooring Mast

by Marcia Amidon Lüsted

When the Empire State Building was conceived, it was planned as the world's tallest building, taller even than the new Chrysler Building that was being constructed at Forty-second Street and Lexington Avenue in New York. At seventy-seven stories, it was the tallest building before the Empire State began construction, and Al Smith was determined to outstrip it in height.

The architect building the Chrysler Building, however, had a trick up his sleeve. He secretly constructed a 185-foot spire inside the building, and then shocked the public and the media by hoisting it up to the top of the Chrysler Building, bringing it to a height of 1,046 feet, 46 feet taller than the originally announced height of the Empire State Building.

Al Smith realized that he was close to losing the title of world's tallest building, and on December 11, 1929, he announced that the Empire State would now reach the height of 1,250 feet. He would add a top or a hat to the building that would be even more distinctive than any other building in the city. John Tauranac describes the plan:

> [The top of the Empire State Building] would be more than ornamental, more than a spire or dome or a pyramid put there to add a desired few feet to the height of the building or to mask something as mundane as a water tank. Their top, they said, would serve a higher calling. The Empire State Building would be equipped for an age of transportation that was then only the dream of aviation pioneers.

This dream of the aviation pioneers was travel by dirigible, or zeppelin, and the Empire State Building was going to have a mooring mast at its top for docking these new airships, which would accommodate passengers on already existing transatlantic routes and new routes that were yet to come.

The Age of Dirigibles

By the 1920s, dirigibles were being hailed as the transportation of the future. Also known today as blimps, dirigibles were actually enormous steel-framed balloons, with envelopes of cotton fabric filled with hydrogen and helium to make them lighter than air. Unlike a balloon, a dirigible could be maneuvered by the use of propellers and rudders, and passengers could ride in the gondola, or enclosed compartment, under the balloon.

Dirigibles had a top speed of eighty miles per hour, and they could cruise at seventy miles per hour for thousands of miles without needing refueling. Some were as long as one thousand feet, the same length as four blocks in New York City. The one obstacle to their expanded use in New York City was the lack of a suitable landing area. Al Smith saw an opportunity for his Empire State Building: A mooring mast added to the top of the building would allow dirigibles to anchor there for several hours for refueling or service, and to let passengers off and on. Dirigibles were docked by means of an electric winch, which hauled in a line from the front of the ship and then tied it to a mast. The body of the dirigible could swing in the breeze, and yet passengers could safely get on and off the dirigible by walking down a gangplank to an open observation platform.

The architects and engineers of the Empire State Building consulted with experts, taking tours of the equipment and mooring operations at the U.S. Naval Air Station in Lakehurst, New Jersey. The navy was the leader in the research and development of dirigibles in the United States. The navy even offered its dirigible, the Los Angeles, to be used in testing the mast. The architects also met with the president of a recently formed airship transport company that planned to offer dirigible service across the Pacific Ocean.

When asked about the mooring mast, Al Smith commented:

> [It's] on the level, all right. No kidding. We're working on the thing now. One set of engineers here in New York is trying to dope out a practical, workable arrangement and the Government people in Washington are figuring on some safe way of mooring airships to this mast.

Designing the Mast

The architects could not simply drop a mooring mast on top of the Empire State Building's flat roof. A thousand-foot dirigible moored at the top of the building, held by a single cable tether, would add stress to the building's frame. The stress of the dirigible's load and the wind pressure would have to be transmitted all the way to the building's foundation, which was nearly eleven hundred feet below. The steel frame of the Empire State Building would have to be modified and strengthened to accommodate this new situation. Over sixty thousand dollars' worth of modifications had to be made to the building's framework.

Rather than building a utilitarian mast without any ornamentation, the architects designed a shiny glass and chrome-nickel stainless steel tower that would be illuminated from inside, with a stepped-back design that imitated the overall shape of the building itself. The rocket-shaped mast would have four wings at its corners, of shiny aluminum, and would rise to a conical roof that would house the mooring arm. The winches and control machinery for the dirigible mooring would be housed in the base of the shaft itself, which also housed elevators and stairs to bring passengers down to the eighty-sixth floor, where baggage and ticket areas would be located.

The building would now be 102 floors, with a glassed-in observation area on the 101st floor and an open observation platform on the 102nd floor. This observation area was to double as the boarding area for dirigible passengers.

Once the architects had designed the mooring mast and made changes to the existing plans for the building's skeleton, construction proceeded as planned. When the building had been framed to the 85th floor, the roof had to be completed before the framing for the mooring mast could take place. The mast also had a skeleton of steel and was clad in stainless steel with glass windows. Two months after the workers celebrated framing the entire building, they were back to raise an American flag again—this time at the top of the frame for the mooring mast.

The Fate of the Mast

The mooring mast of the Empire State Building was destined to never fulfill its purpose, for reasons that should have been apparent before it was ever constructed. The greatest reason was one of safety: Most dirigibles from outside of the United States used hydrogen rather than helium, and hydrogen is highly flammable. When the German dirigible Hindenburg was destroyed by fire in Lakehurst, New Jersey, on May 6, 1937, the owners of the Empire State Building realized how much worse that accident could have been if it had taken place above a densely populated area such as downtown New York.

The greatest obstacle to the successful use of the mooring mast was nature itself. The winds on top of the building were constantly shifting due to violent air currents. Even if the dirigible were tethered to the mooring mast, the back of the ship would swivel around and around the mooring mast. Dirigibles moored in open landing fields could be weighted down in the back with lead weights, but using these at the Empire State Building, where they would be dangling high above pedestrians on the street, was neither practical nor safe.

The other practical reason why dirigibles could not moor at the Empire State Building was an existing law against airships flying too low over urban areas. This law would make it illegal for a ship to ever tie up to the building or even approach the area, although two dirigibles did attempt to reach the building before the entire idea was dropped. In December 1930, the U.S. Navy dirigible Los Angeles approached the mooring mast but could not get close enough to tie up because of forceful winds. Fearing that the wind would blow the dirigible onto the sharp spires of other buildings in the area, which would puncture the dirigible's shell, the captain could not even take his hands off the control levers.

Two weeks later, another dirigible, the Goodyear blimp Columbia, attempted a publicity stunt where it would tie up and deliver a bundle of newspapers to the Empire State Building. Because the complete dirigible mooring equipment had never been installed, a worker atop the mooring mast would have to catch the bundle of papers on a rope dangling from the blimp. The papers were delivered in this fashion, but after this stunt the idea of using the mooring mast was shelved. In February 1931, Irving Clavan of the building's architectural office said, "The as yet unsolved problems of mooring air ships to a fixed mast at such a height made it desirable to postpone to a later date the final installation of the landing gear."

By the late 1930s, the idea of using the mooring mast for dirigibles and their passengers had quietly disappeared. Dirigibles, instead of becoming the transportation of the future, had given way to airplanes. The rooms in the Empire State Building that had been set aside for the ticketing and baggage of dirigible passengers were made over into the world's highest soda fountain and tea garden for use by the sightseers who flocked to the observation decks. The highest open observation deck, intended for disembarking passengers, has never been open to the public.

"The Mooring Mast" by Marcia Amidon Lüsted, from The Empire State Building*. Copyright © 2004 by Gale, a part of Cengage Learning, Inc.*

Prompt

Based on the excerpt, describe the obstacles the builders of the Empire State Building faced in attempting to allow dirigibles to dock there. Support your answer with relevant and specific information from the excerpt.

A7: Essay Set #7

Prompt

Write about patience. Being patient means that you are understanding and tolerant. A patient person experience difficulties without complaining.

Do only one of the following: write a story about a time when you were patient OR write a story about a time when someone you know was patient OR write a story in your own way about patience.

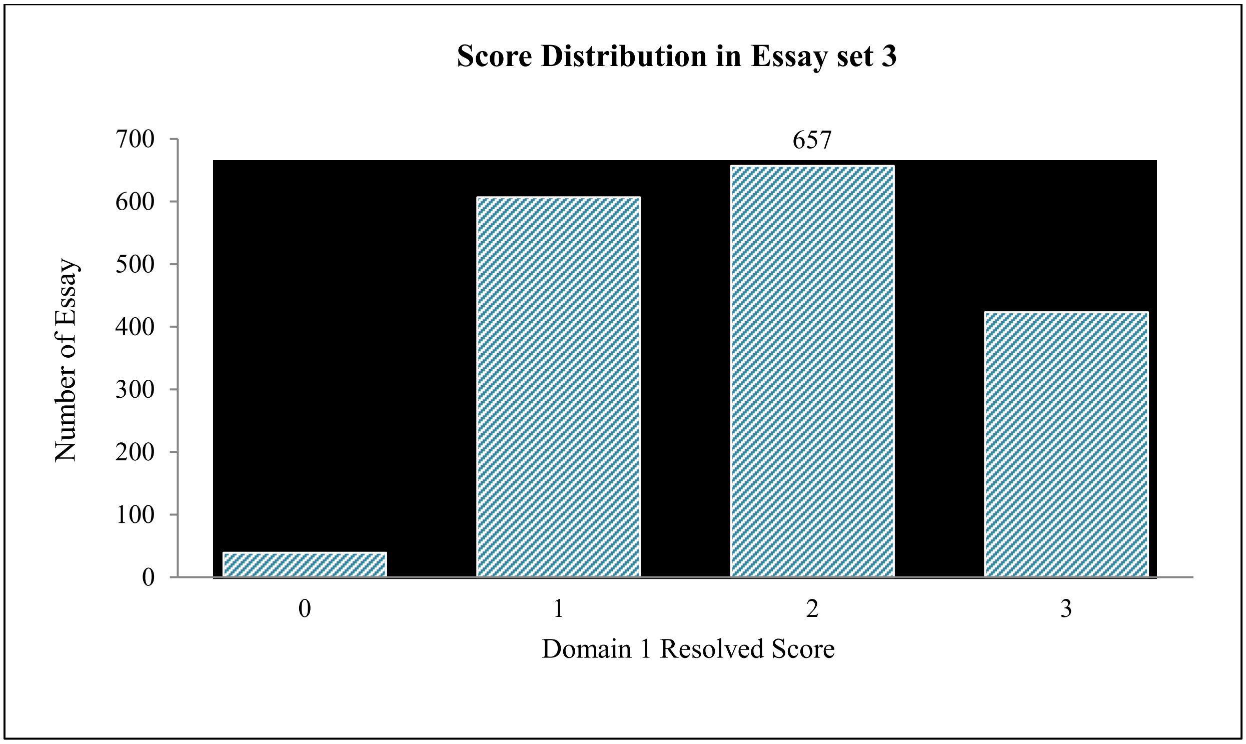
A8: Essay Set #8

Prompt

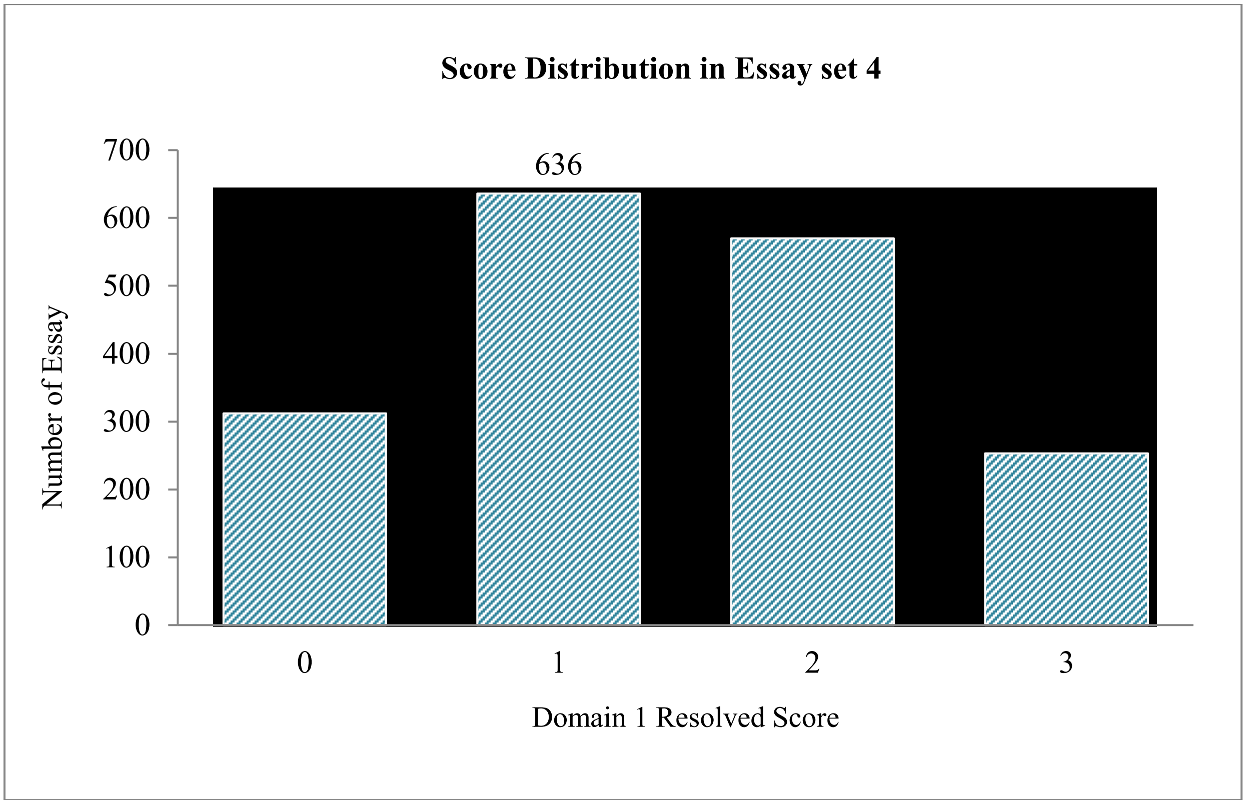
We all understand the benefits of laughter. For example, someone once said, “Laughter is the shortest distance between two people.” Many other people believe that laughter is an important part of any relationship. Tell a true story in which laughter was one element or part.

Appendix B: Essay Score Distribution

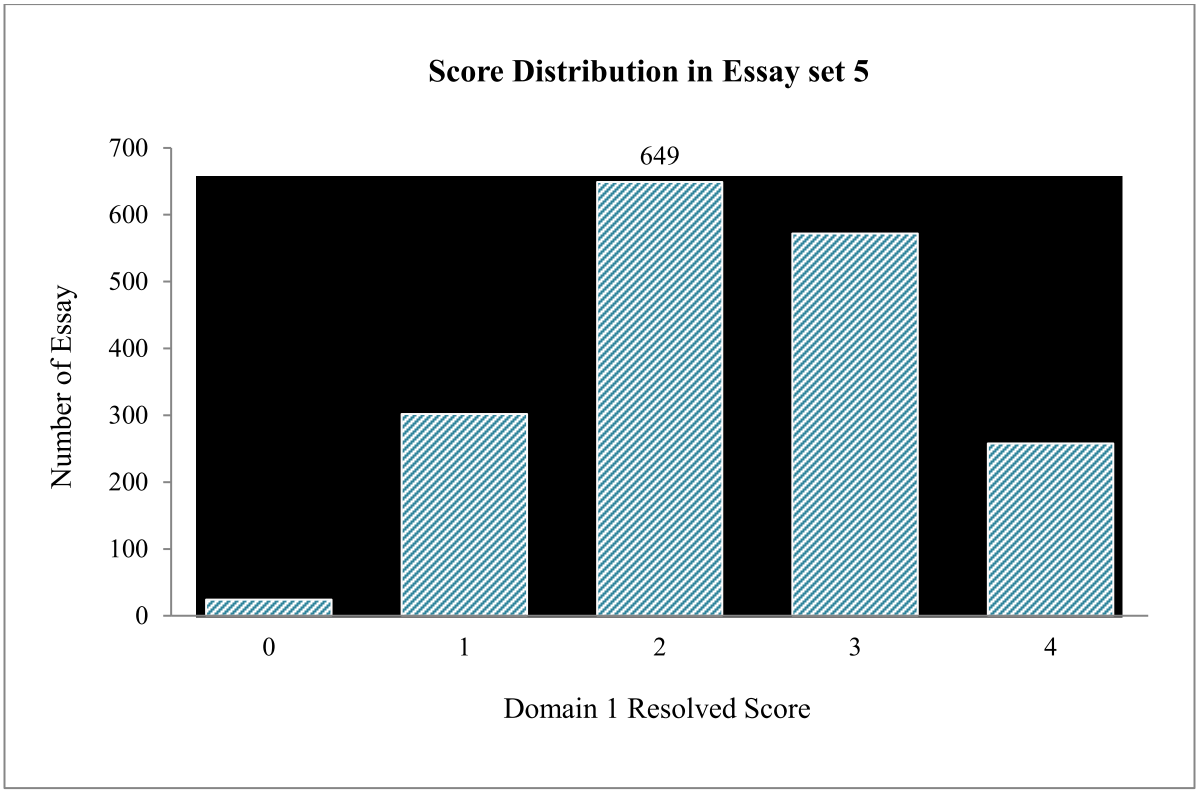
B1: Score Distribution in Essay set #3



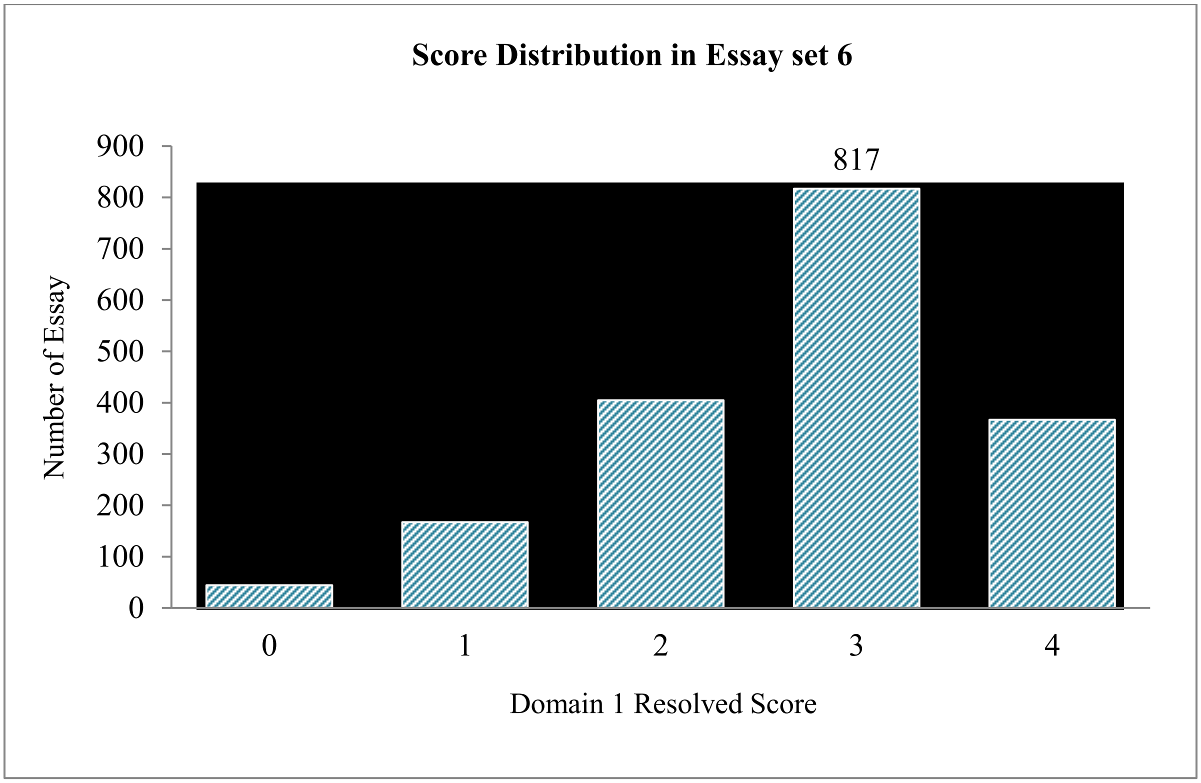
B2: Score Distribution in Essay set #4



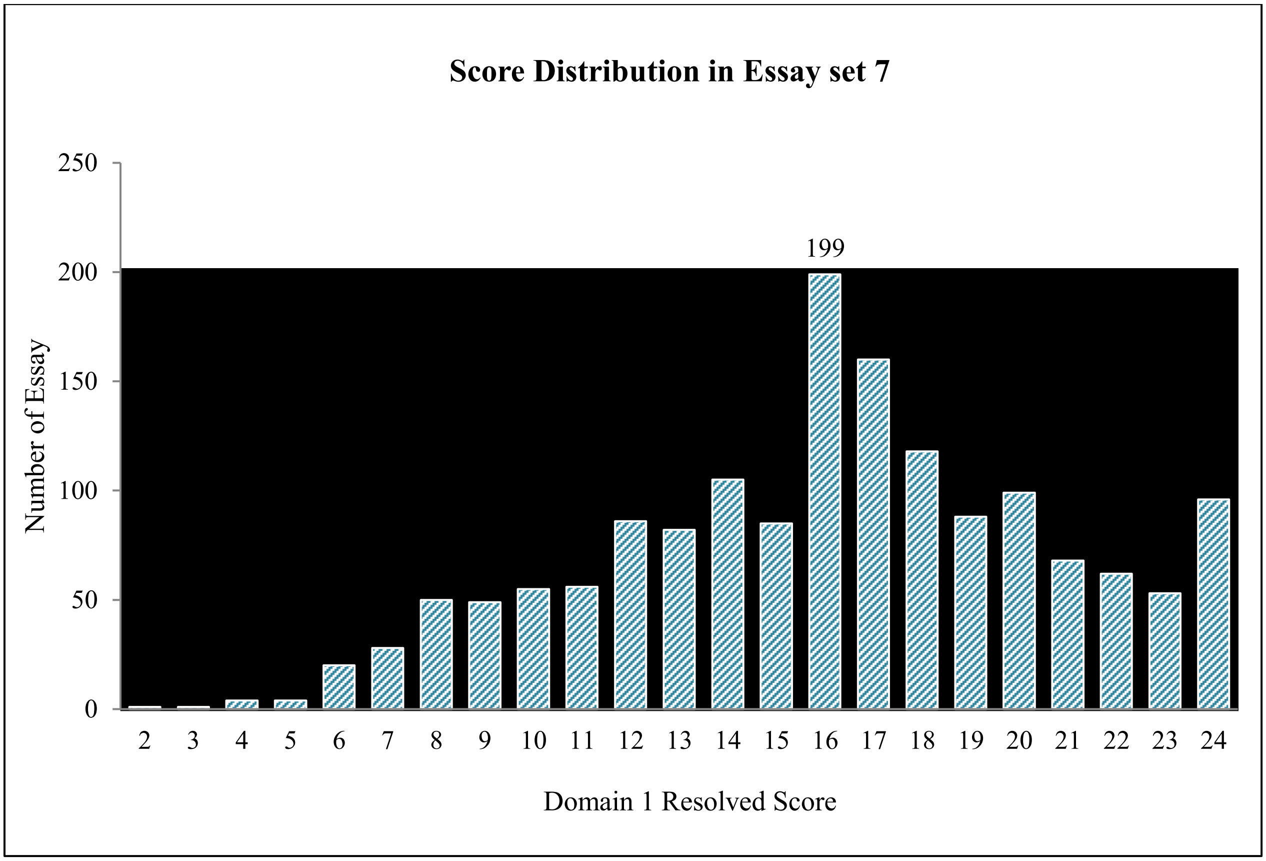
B3: Score Distribution in Essay set #5



B4: Score Distribution in Essay set #6



B5: Score Distribution in Essay set #7


Score Distribution in Essay set 7
Number of Essay
Domain 1 Resolved Score