AUTOMATED ESSAY SCORING SYSTEM USING MULTI-MODEL MACHINE LEARNING

Wilson Zhu[1] and Yu Sun[2]

[1]Diamond Bar High School, Diamond Bar, California, USA
[2]California State Polytechnic University, Pomona, California, USA

ABSTRACT

Standardized testing such as the SAT often requires students to write essays and hires a large number of graders to evaluate these essays which can be time and cost consuming. Using natural language processing tools such as Global Vectors for word representation (GloVe), and various types of neural networks designed for picture classification, we developed an automatic grading system that is more time- and cost-efficient compared to human graders. We applied our application to a set of manually graded essays provided by a previous competition on Kaggle in 2012 on automated essay grading and conducted a qualitative evaluation of the approach. The result shows that the program is able to correctly score most of the essay and give an evaluation close to that of a human grader on the rest. The system proves itself to be effective in evaluating various essay prompts and capable of real-life application such as assisting another grader or even used as a standalone grader.

KEYWORDS

Automated Essay Scoring System, Natural Language Processing, Multi-Model Machine Learning.

1. INTRODUCTION

Automated essay scoring originated with work of Ellis Batten Page. Page suggested the possibility of such a system in 1966 [1] and that such a system can match the performance of human judges.

Page created the Project Essay Grade (PEG) in 1968 [2-5] but technology at the time would not have allowed his system to be cost-effective [6] and he eventually sold his system, PEG, to Measurement Inc. Other systems have been developed such as the Intellimetrics by Vantage Learning which was first used in 1998[7] and the E-rater offered by Educational Testing Service that was first used in 1999 [8].

More recently, the 2012 Kaggle competition, Automated Student Assessment Prize, sponsored by Hewlett Foundation [9] saw numerous teams attempting to develop a program that is capable of scoring an essay to the same ability that a human grader could. The winning team achieved a kappa of 0.81407. There has been very few studies and breakthrough in this field after the Kaggle competition, and these recent studies are mostly based on this Kaggle competition.

Previous automated essay scoring systems such as the e-rater used feature extraction to obtain necessary information from an essay [10-12]. These systems are effective but lack the ability to accurately evaluate the content of the essay as it measures shared vocabulary between the prompt

David C. Wyld et al. (Eds): MLNLP, BDIoT, ITCCMA, CSITY, DTMN, AIFZ, SIGPRO - 2020
pp. 109-117, 2020. CS & IT - CSCP 2020 DOI: 10.5121/csit.2020.101211

and the essay and cannot reach the level of content-comprehension that a word vector model can achieve. However, e-rater does have error analysis and style analysis which word vector models cannot achieve. Despite its advantages, the feature extraction model ultimately falls short in the content evaluation aspect which is arguably the most crucial aspect of the essay.

In this paper, we attempt to combine the approaches of feature extraction model and word vector model. Using feature extractions to obtain word count, grammar mistakes, and part of speech count and implementing word vectors such as GloVe, this method can measure both the numerical features of the essay as well as the contextual feature and its relatedness to the topic. Compared to previous methods where only either feature extraction or word vector is present, our method combines both and takes advantage of both methods while minimizing the shortcomings of only using one. Therefore, we believe our automated essay grading has potential to be less vulnerable to 'study to test' essays.

To evaluate the result of the program, we test our data on the validation set consisting of 20 percent of the overall data and obtain a kappa score through the following function:

$$\kappa = \frac{e^{2z} - 1}{e^{2z} + 1}$$

Where the quadratic weighted kappa is calculated, and the mean is taken across all data. A kappa score of 1 demonstrates total agreement between the raters - machine and human - while 0 demonstrates random agreement and a kappa less than zero indicates agreement is less than that by chance.

The rest of the paper is organized as follows: Section 2 gives detail regarding the challenges faced during data transformation, data tokenization and network evaluation; Section 3 focuses on our methodology and solutions to the challenges offered in Section 2; Section 4 present the result of evaluation of the program on the validation set followed by presenting related work in Section 5; Finally, Section 6 gives conclusion remarks as well as possible future works of this project.

2. CHALLENGES

There are a multitude of challenges that exist in the project. They will be discussed in this section.

2.1. Challenge 1: Tokenizing the data

The data can be tokenized in a variety of ways. The essay can be tokenized through the use of a GloVe embedding layers and word tokenizer with ease. However, to add to the GloVe representation with numerical representation such as word count and grammar mistake count is a challenge. Since using the GloVe embedding layer required the input to be solely the word index of the individual word of the essay and convert the word indexes into a 2-dimensional array that represents the context of the essay. Thus, adding other information proves to be impossible and requires an overhaul of the tokenization method, either find a way to represent this additional numerical information within the 2-dimensional array or find another method to replace the 2-dimensional array.

2.2. Challenge 2: Choosing the neural network

There are numerous ways to create a neural network and choosing the optimal network proves to be a major challenge. There are multiple aspects to consider when choosing the network - the type of neural network, the activation function, and the number of neurons. The type of neural network depends heavily on the application. For example, Recurrent Neural Network is best suited for sequential data while Convolution Neural Network works best on image data. In addition, choosing the right activation can be crucial. Depending on the type of network, Rectified Linear Unit, Sigmoid, or SoftMax are utilized to maximize the accuracy. Furthermore, the number of neurons in each layer may influence the outcome as well. To optimize the neural network, choosing the correct elements can be difficult.

2.3. Challenge 3: Training and Evaluation

When training the model, there are multiple things to consider - epoch, batch size, optimizer, and metrics. It is imperative to find the optimal epoch and batch size combination to minimize the training time since the training can be extremely time-consuming. The training time and accuracy is also affected by the optimizer used. Therefore, the most efficient optimizer for this specific application needs to be selected. The most challenging aspect of training is choosing the correct metric to maximize the evaluation score. Since the model is being rated through quadratic weighted kappa, it is necessary to find or create the closest metrics available to the evaluation function and maximize the score.

3. SOLUTION

The Automated Essay consisted of two major components, the essay tokenizer and the neural network model. The tokenizer converts the essay, a string, into two vectors, one containing the numerical representation and another one containing the word vector representation of the essay. The vector is then passed into the neural network which evaluates the model using the trained model and returns a score.

In detail, the preprocessing process outputs a numerical representation for feature extraction as well as a sequence for the word vector model. The numerical representation consists of the following:

- Grammar Counts: Counts of various parts of speech using spaCy to reflect the writer's ability to utilize a wide aspect of the English language. After evaluating the essay's grammar and spelling mistakes, those mistakes are corrected for further processing.

- Numerical Counts: Total word count, character count, unique word count, average word length, sentence count, paragraph count, and comma count of the corrected essay; Stop words from the spaCy library are removed and total word count, character count, unique word count, and average word length are calculated and the difference between the pre-removal and post-removal are calculated as well.

The input sequence for word vector neural network consists of a sequence of fixed size from Keras text tokenizer. The overview of our solution is represented in the figure below:

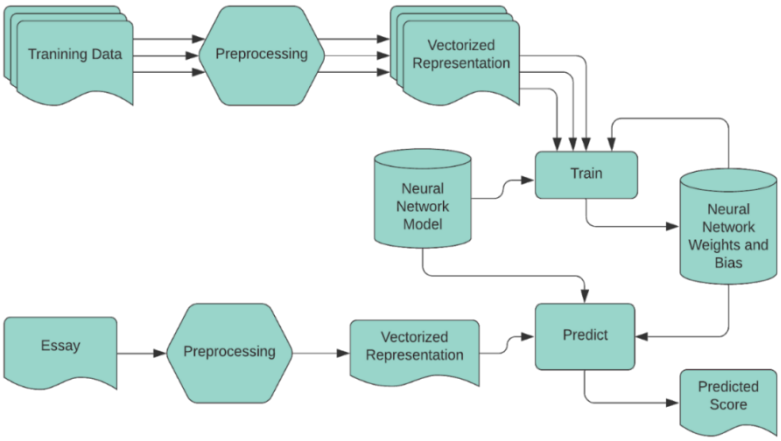

Figure 1: Overview of the Solution

Using Keras Functional API, we created a custom multi-model neural network consisting of a two-layer neural network to processes the numerical representation of the essay and a word vector neural work that processes the sequence from Keras tokenizer. Then the outputs of both neural networks are concatenated and outputted into another two layers neural network to produce the final score. Graphical representation of the multi-model neural network with Long Short-Term Memory (LSTM) as word vector neural network and a GloVe embedding dimension of 50:

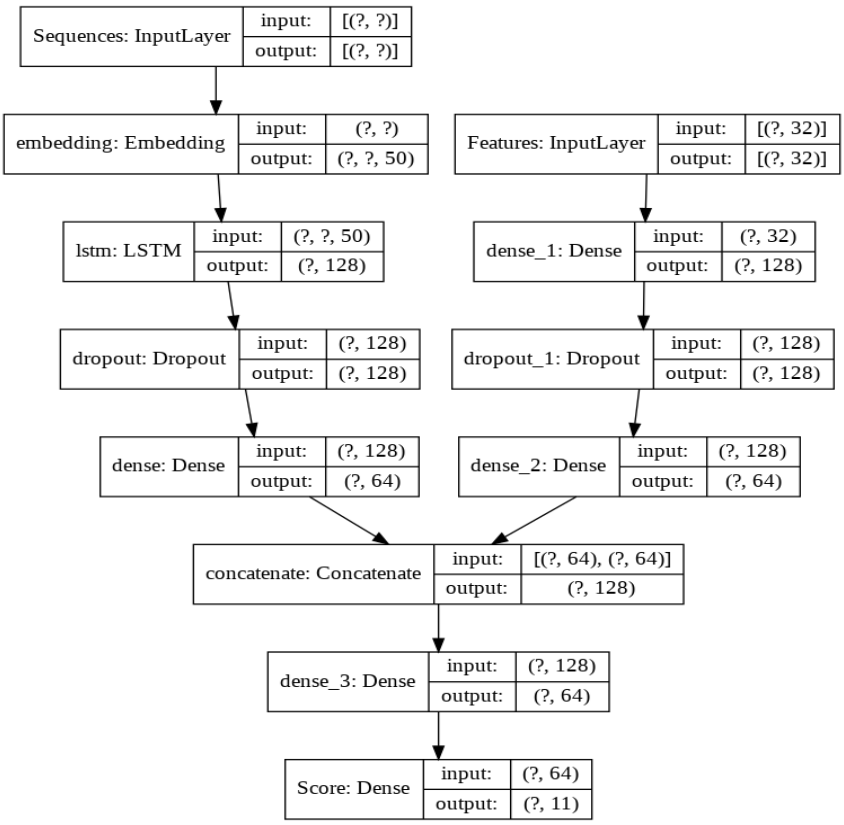

Figure 2: Graphical Representation of the Neural Network with LSTM

We used 'adaptive moment estimation' ('adam') optimizer and in order to minimize the training time we trained each neural network with a batch size of 50 essays, epoch of 40, and a varying learning rate as follows:

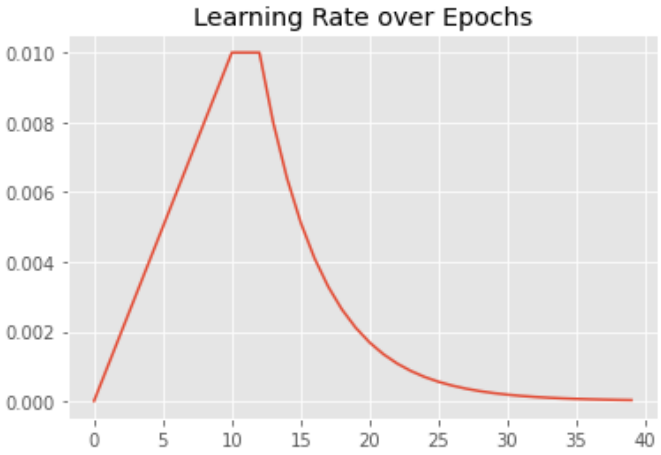

Figure 3: Custom Learning Rate over Epoch using 'adam' optimizer

4. EXPERIMENTS

To maximize performance, we decided to test two parameters – type of neural network used for GloVe embedding, and length of the GloVe embedding dimension. The types of neural network we tested are Long Short-Term Memory, Gated Recurrent Unit (GRU), and Bi-Directional LSTM (BiLSTM). The various length of GloVe embedding dimensions are 50, 100, 200 and 300.

Various neural network has their own advantages and disadvantages. For example, LSTM contains an input gate, an output gate, and a forget gate, whereas GRU contains a reset gate and an update gate. This difference leads to GRU having a shorter training time but LSTM having better performance on longer sequences. BiLSTM is LSTM with a bidirectional layer that reads the given text from beginning to end and end to beginning, which allows the neural network to capture more information since LSTM tend to 'forget' information in the beginning of the text. However, BiLSTM is more computationally expensive. Thus, we needed to test for the best neural network for our system.

Different sizes of GloVe word embedding can impact the neural network performance as a longer embedding dimension would allow the input essay to be expressed more thoroughly whereas a shorter embedding dimension allows the essay to be expressed more concisely. This difference could impact the performance since a larger embedding dimension would allow for too much unnecessary details, but a shorter embedding dimension may lead to crucial information being lost. Therefore, we decided to use the embedding dimension as one of the parameters for our experiment.

4.1. Neural Networks

To test for the most optimal neural network, we took the average kappa score across the various sizes of GloVe embedding dimension of each network. LSTM obtained an average of 0.69479, GRU obtained an average of 0.63776, and BiLSTM obtained an average of 0.67321. Furthermore, the best performing model for LSTM, GRU, and BiLSTM obtained a kappa of 0.70026, 0.68525, and 0.70024, respectively, with LSTM being the best performing neural

network overall. As the figures below demonstrate, LSTM has the highest training accuracy and lowest training categorical cross entropy in one of the eight essay set, with other training essay set having a similar trend.

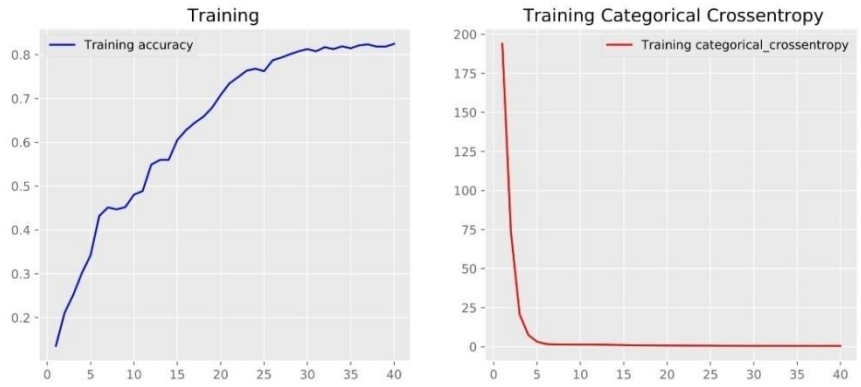

Figure 4: Training based on LSTM Model

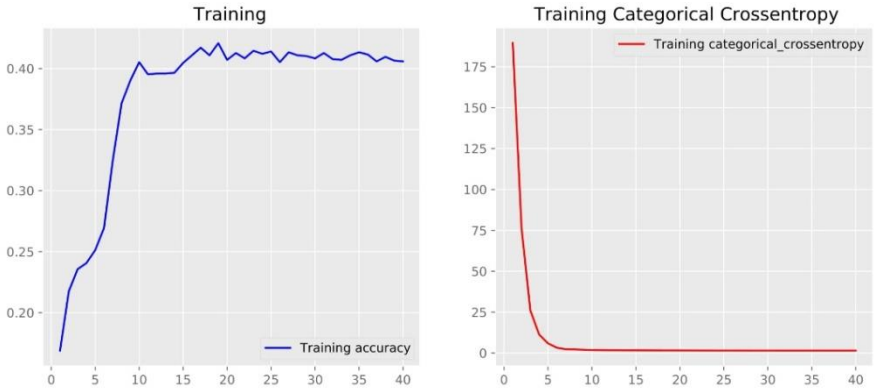

Figure 5: Training based on GRU Model

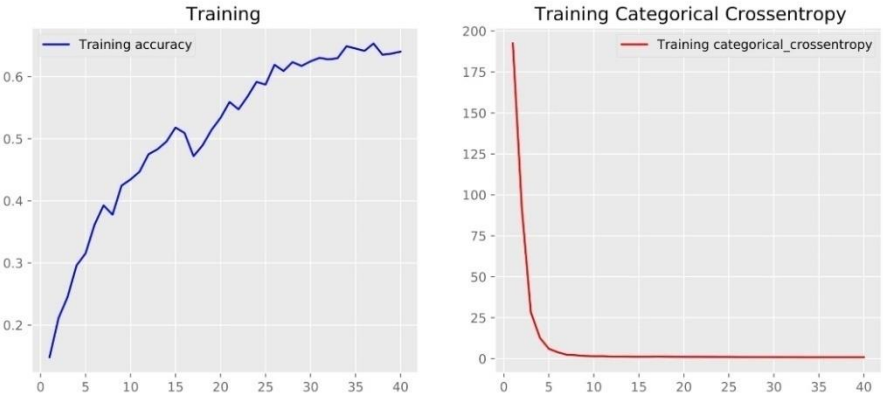

Figure 6: Training based on Bi-Directional LSTM Model

The result shows LSTM being the most optimal neural network. GRU performed worse than the other two because as stated above, it has a faster training time but is not better suited for longer text model than LSTM. BiLSTM on the other hand may have retained more information, but the kappa score evaluates agreements with a human reader. This poor result may be explained by human grader reading the essay from top to bottom. Although BiLSTM is able to retain most information, even the top of the essay since BiLSTM reads the essay in both directions, the human grader tends to remember information later in text as people only read from top to bottom, which is similar to LSTM. Thus, LSTM performed best on this application as it is most similar to how a human grader read and evaluate an essay.

4.2. GloVe Embedding Dimension

To find the most optimal GloVe embedding dimension, we compared the quadratic weighted kappa of each embedding dimension with different neural networks. The table below shows the scores:

Table 1: Quadratic Weighted Kappa of Tested Neural Networks

Dimension\Network	LSTM	GRU	BiLSTM	Average
50	0.70003	0.57328	0.70024	0.65785
100	0.68477	0.68525	0.67184	0.68062
200	0.70026	0.61821	0.68884	0.66910
300	0.69411	0.67478	0.63192	0.66694

From the table, we can see that the embedding dimension does not greatly affect the performance of LSTM model as much. However, the embedding dimension did affect GRU greatly and BiLSTM to a lesser extent. The result shows GRU preferring an embedding dimension of 100 and 300, which indicates that specific categories within the 100 and 300 embedding dimension allows for better understanding for GRU. The trend is clearer for BiLSTM with a strong preference for shorter embedding dimensions. This may be a result of BiLSTM also retaining much of the information and does not need for longer embedding dimensions to express the details.

In conclusion, we used LSTM neural network with 200 embedding dimensions as our final model since it achieved the highest quadratic weighted kappa at 0.70026.

5. RELATED WORK

Other methods have been written regarding the automated essay grading competition on Kaggle. For example, a team from Rice University obtained a kappa score of 0.63 through extraction of features such as word occurrence, word count, Kullback–Leibler divergence and using linear regression [13]. Similarly, another team from Stanford also attempted the problem using machine learning, with feature extraction and linear regression, scored a kappa of 0.72. The second used a different approach as they included features such as bag of words and part of speech count [14].

In contrast, a team from Stanford University used word vectors and a 2 layers neural network to achieve a score of 0.9447875. They utilized the GloVe word vector with various dimensions and types to tokenize the essay combined with various different types of neural networks and managed a high score [15]. However, this team does not evaluate the various features of the essay such as word count, unique word count, grammar mistakes, etc.

The first two teams were less effective in their method as they only accounted for the numerical features of the essay and cannot accurately account for the context due to the lack of word vectors. The third team can thoroughly evaluate the content and achieved a high kappa score as a result.

6. CONCLUSION AND FUTURE WORK

In conclusion, we developed an easy grading system using machine learning and natural language processing that achieve a quadratic weighted kappa of 0.70026. We used both feature extraction and word vectors to represent the essay and convert the essay into its vectorized representation and tokenized sequences. Then a LSTM neural network is utilized to evaluate the tokenized sequences and a 2-layer neural network to evaluate the vector representation. The results are concatenated, and another 2-layer neural network is used to predict the final score. The resulting kappa demonstrates that the current model is capable of real-world application but still has some shortcomings.

For example, the current model performs vastly better on certain prompts than others and performs significantly worse on longer essays compared to shorter ones since the word vector has a vaguer representation of longer essays and cannot accurately extract the context of the longer essays.

We hope to combat the longer essay problem by giving more weights to certain words using Term Frequency - Inverse Document Frequency. This approach allows less significant words, which appears more frequently in longer essays, to have less weight and thus the content of longer essays to become more expressed.

REFERENCES

[1] Page, E. B. (1966). "The imminence of... grading essays by computer". The Phi Delta Kappan. 47 (5): 238–243.
[2] Page, E.B. (1968). "The Use of the Computer in Analyzing Student Essays", International Review of Education, 14(3), 253-263.
[3] Hsieh, Kevin Li-Chun, Chung-Ming Lo, and Chih-Jou Hsiao. "Computer-aided grading of gliomas based on local and global MRI features." Computer methods and programs in biomedicine 139 (2017): 31-38.
[4] Slotnick, Henry B. "Toward a theory of computer essay grading." Journal of Educational Measurement 9, no. 4 (1972): 253-263.
[5] Czaplewski, Andrew J. "Computer-assisted grading rubrics: Automating the process of providing comments and student feedback." Marketing Education Review 19, no. 1 (2009): 29-36.
[6] Nguyen, Tien Dzung, Quyet Hoang Manh, Phuong Bui Minh, Long Nguyen Thanh, and Thang Manh Hoang. "Efficient and reliable camera based multiple-choice test grading system." In The 2011 International Conference on Advanced Technologies for Communications (ATC 2011), pp. 268-271. IEEE, 2011.
[7] "IntelliMetric®: How it Works", Vantage Learning. Retrieved 28 February 2012.
[8] Burstein, Jill (2003). "The E-rater(R) Scoring Engine: Automated Essay Scoring with Natural Language Processing", p. 113. In Shermis, Mark D., and Jill Burstein, eds., Automated Essay Scoring: A Cross-Disciplinary Perspective. Lawrence Erlbaum Associates, Mahwah, New Jersey, ISBN 0805839739
[9] Hewlett prize" Archived 30 March 2012 at the Wayback Machine. Retrieved 5 March 2012.
[10] Attali, Y. & Burstein, J. (2006). Automated Essay Scoring With e-rater® V.2. Journal of Technology, Learning, and Assessment, 4(3).
[11] Burstein, Jill, Karen Kukich, Susanne Wolff, Chi Lu, and Martin Chodorow. "Enriching automated essay scoring using discourse marking." (2001).

[12] Aluthman, Ebtisam S. "The effect of using automated essay evaluation on ESL undergraduate students' writing skill." International Journal of English Linguistics 6, no. 5 (2016): 54-67.
[13] Lukic, A., & Acuna, V. (n.d.). Automated Essay Scoring, Rice University.
[14] Manvi Mahana, Mishel Johns, and Ashwin Apte. (2012). Automated essay grading using machine learning. Mach. Learn. Session, Stanford University.
[15] Nguyen H. & Dery L. (2018). Neural Network for Automated Essay Grading. Stanford University.