Natural Language Processing For Guided English Language Learner In Grades 8 - 12

Uzair Tahamid Siam
University of Rochester
Rochester NY
usiam@u.rochester.edu

Runtao Zhou
University of Rochester
Rochester NY
rzhou15@u.rochester.edu

Nina (Yueyue) Long
University of Rochester
Rochester NY
ylong6@u.rochester.edu

Abstract

Writing is a foundational skill that few students are able to master, often because writing tasks are infrequently assigned in school. A rapidly growing student population, students learning English as a second language, known as English Language Learners (ELLs), are especially affected by the lack of practice. In this project we create an automated system using a Long short-term memory (LSTM) and a pretrained BERT model to score essays of ELLs from grades $8-12$ *on 6 measures. For the LSTM model we receive a MCRMSE of* 0.6314 *on the training data and* 0.6460 *on the validation dat and for the BERT model we receive a MCRMSE of* 0.0453 *on the training set and* 0.1280 *on the validation set.*

1. Introduction

Using the tools of Natural Language Processing (NLP), in this project we aim to evaluate the essay writing skills of English Language Learners (ELLs) from grades $8-12$ on six criteria - cohesion, syntax, vocabulary, phraseology, grammar, and conventions - where each criteria measures within the range $1.0-5.0$ in increments of 0.5. The idea is by sensitizing the automated systems to a certain level of proficiency we can make the model more fair towards these ELLs and in doing so can improve the usefulness of the feedbacks.

1.1. The Team

We are a team of three individuals each playing a different role. The role distribution for the project were as follows:

- Uzair Tahamid Siam
  - Exploratory Data Analysis
  - Building and experimenting with a LSTM model
  - Creating and hosting a demo that users can interact with on HuggingFace Spaces
- Runtao Zhou
  - Building and Experimenting BERT uncased model
  - Optimizing the training data for BERT uncased model
- Nina (Yueyue) Long
  - Collect related research on automatic essay rating

1.2. Motivation

While there are existing tools that can evaluate language knowledge, they were not created with English Language Learners in mind who have different needs as beginners. Leveraging modern NLP and Deep Learning methods we create an automated process for students to learn to write English, especially, students who want to learn it as a second language and might not have access to proper resources or supervision to do so. This work can help ELLs receive more accurate feedback on their language development and expedite the grading cycle for teachers. Additionally, these outcomes could enable ELLs to receive more appropriate learning tasks that will help them improve their English language proficiency.

1.3. The Data

The data used in this project is from a Kaggle competiton hosted by Vanderbilt University and the Learning Agency Lab. The training data provided has a total of 3911 unique essays each with a score for cohesion, syntax, vocabulary, phraseology, grammar, and conventions ranging from $1.0-5.0$ in increments of 0.5. The competition also provided a test set of only 3 unique essays with no labels. As a result we decided to not use the test set at all for our purposes. We further discuss the data in section 3.1.

2. Related Work

There have been many recent research focused on using LSTM and BERT models for automated essay scoring. For example, some studies have used LSTM to model the sequence of words in an essay and predict the overall score. [8] Other research has focused on using BERT, a transformer-based model, to capture the contextual information in an essay and generate a score.[7]

One of the key advantages of using NLP for essay scoring is that it can effectively capture the semantic and syntactic features of the text, which are important for evaluating the quality of an essay. This has led to high correlations with human scores on various datasets.[8]

However, there are also challenges associated with using LSTM and BERT models for essay scoring. One major challenge is the difficulty of training these models on large and diverse datasets, which is necessary to improve the generalizability of the model.[7] In addition, there is a need for careful evaluation and analysis of the model's performance to ensure that it is fair and unbiased.[8]

Overall, the use of NLP for automated essay scoring is an active area of research, with the potential to provide efficient and accurate evaluations of student writing. However, there are still many challenges to be addressed in order to fully realize the potential of these models in this domain.

Additionally, none of the research we found were specifically trained for ELLs who have specific needs as beginners. As a result, our research is aimed towards this specific demographic with the hopes of facilitating their learning process.

3. Methods

3.1. Exploratory Data Analysis

We explore the dataset to check for null values. As shown in Figure. 1 below, the data provided to us has no null values for any of the features. We also take a look at the distribution of the 6 measures to understand how the essays are labelled. As shown in Figure. 2, the measures are dominated 3.0 as one would expect given that 3.0 is the mean. Given this distribution we will have to be careful to ensure the models we build do not score all essays a 3.0 during inference. Additionally, we also look at a correlation heat-map of the scores to understand whether the measures are related to each other. We expected the 6 measures to be highly correlated as these were selected to provide a holistic evaluation of the essays. From Figure. 3 we see a high correlation between the 6 measures as we expected. Finally, an analysis of the word length distribution of the dataset essays is carried out. Figure. 4 shows that most of the essays range around $300 - 500$ words. This analysis will be useful when we pad sentences during the model training process.

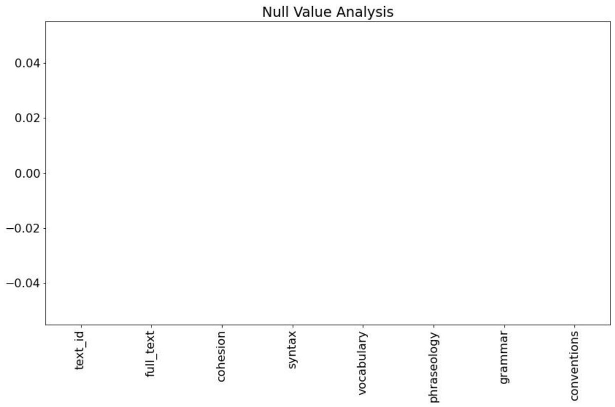

Figure 1. Null Value Density plot showing that there is no null-values in the dataset

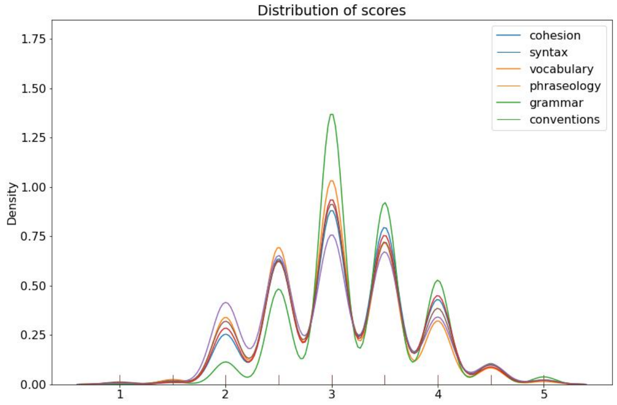

Figure 2. Distribution of the six measures. Mean for each measure: cohesion $= 3.1$, syntax $= 3.0$, vocabulary $= 3.2$, phraseology $= 3.1$, grammar $= 3.0$, conventions $= 3.1$

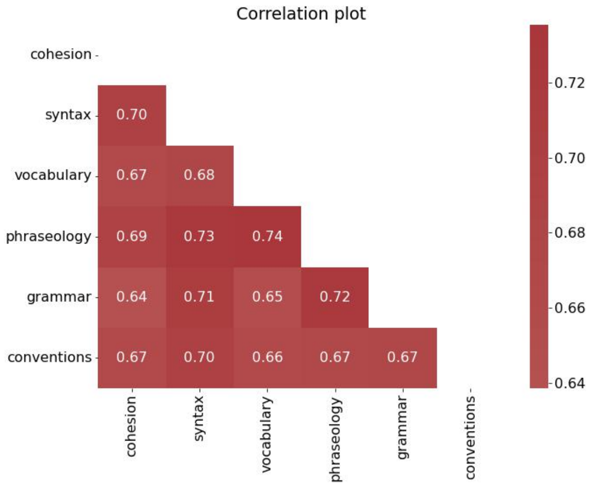

Figure 3. Correlation plot showing high correlation between the six measures

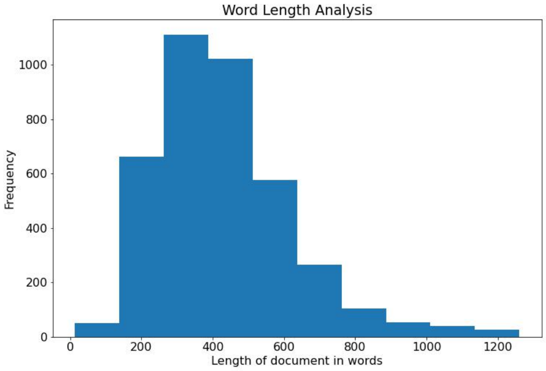

Figure 4. Distribution of length of essays in words

3.2. Long Short-Term Memory (LSTM)

As our baseline model we decided to create a LSTM [3] model. Since we were training on a specific corpus for a specific task we decided to train a tokenizer on the sample data to build a vocabulary instead of using a pretrained vocabulary like GloVe [6] or FastText [1]. The model parameters are given in Table 1. Additionally, using a learning rate finder provided by PyTorch Lightning library we found a suitable learning-rate of 0.0006 as shown in Figure. 5 Next, a suitable loss function for the task was chosen. Given that we are predicting six features, we decided to use a Mean-Columnwise-Root-Mean-Squared-Error (MCRMSE) given by the equation,

$$\text{MCRMSE} = \frac{1}{N_t}\sum_{j=1}^{N_t}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_{ij}-\hat{y}_{ij})^2}$$

where n is the number of essays and N_t is the number of predicted features (six). Finally, using an Adam [4] optimizer and a Cosine Learning Rate Annealer [5] the LSTM model was trained with a batch size of 64 for 25 epochs. The loss curve from the training process is shown below in Figure. 6 and clearly shows that the model is learning as both the training and the validation losses decrease over steps. The final MCRMSE score that we find using this method is shown in Table 2. With a baseline to compare against, we move on to train a more advanced transformer model that is known to perform significantly better than LSTMs.

Parameter	Value
Vocabulary Size	25089
Embedding Dimension	13
Hidden Dimension	8
Sequence Length	512
Num of Layers	4
Output Dimension	6

Table 1. Table of hyperparameters of the model

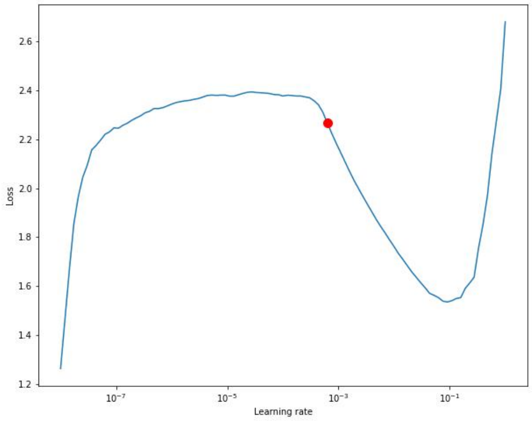

Figure 5. Plot to find the optimal initialization learning-rate

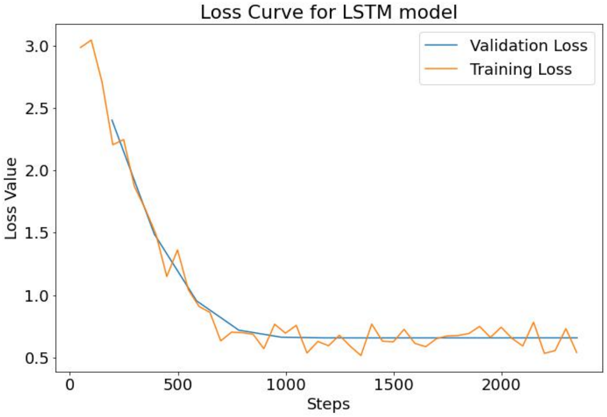

Figure 6. Loss curve for the training and validation data as a function of steps

Dataset	MCRMSE
Training	0.6314
Validation	0.6460

Table 2. MCRMSE for the training and validation data. This is a fairly large MCRMSE value as the scores range from only $1.0 - 5.0$ in 0.5 increments

3.3. BERT-Base-Uncased

For our final model, we decided to fine tune a pre-trained BERT[2] base uncased model. The reason we chose the BERT base model uncased as our final was because the model has been trained on large text corpurs and learned an inner representation of English language. Thus, it is very easy to extract features useful for downstream tasks such as evaluating essay writing skills of English Language Learners (ELLs). The pretrained BERT based uncased model parameters are given in the Table 3. Next, the loss function we used for pre-trained BERT base model was the Mean-Columnwise-Root-Mean-Square-Error (MCRMSE) as explained in LSTM model. The Adam optimizer[4] and Cosine Learning Rate Annealer[5] were used for training the BERT base uncased model with a batch size of 2 for 10 epochs. The loss curve for the training and validation has been shown in the Figure 7. The Bert uncased model's training and validation loss both decreased over the period of 10 epochs.

Parameter	Value
Vocabulary Size	40000
Hidden Size	768
Attention Heads	12
Num of Layers	12
Trainable Parameters	110M
Output Dimension	6

Table 3. Table of hyperparameters of the model

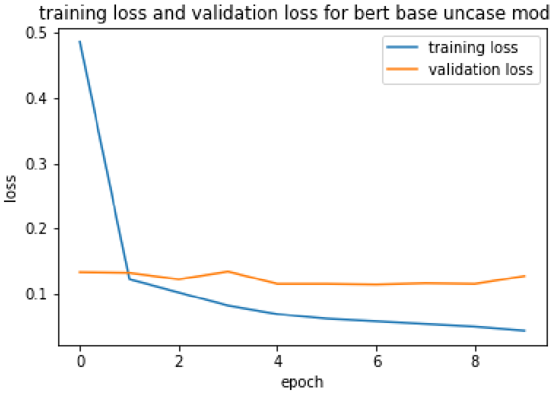

Figure 7. Loss curve for the training and validation data as a function of epochs

The final MCRMSE score for BERT base uncased model is shown in the Table 4. The table shows that BERT base uncased model's performance is much better than LSTMs.

Dataset	MCRMSE
Training	0.0453
Validation	0.1280

Table 4. MCRMSE for the training and validation data for BERT uncased model

4. Experiments

4.1. LSTM

Using our LSTM model we realized that all the predicted measures were in the range $3.0 - 3.5$ which is incorrect as shown in Table 5. Given the distribution of the scores as shown in Figure. 2 we had a hypothesis that the predictions were greatly influenced by the mean or the median of the measure distribution which the BERT model seemed to avoid. To test our hypothesis about this problem we applied a simple transformation to a subset of the essays thus creating a new distribution of the measures as shown in Figure. 8. This resulted in predictions for all the measures yet again close to the mean as shown in Table 6. We then increased the complexity of the LSTM model to increase the number of hidden units as well as making it bidirectional. However, this behavior still persisted. We assume that the model would behave better if the score distribution was uniform, however, due to time constraints we decided to not explore this.

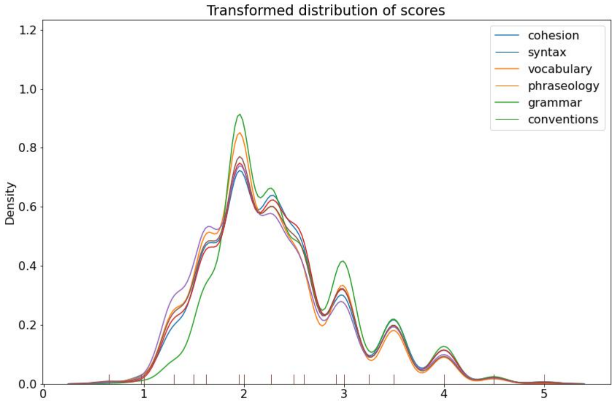

Figure 8. Transformed distribution of the measures after a subset of the data was multiplied by 0.65. Mean for each measure: cohesion $= 2.3$, syntax $= 2.3$, vocabulary $= 2.4$, phraseology $= 2.3$, grammar $= 2.3$, conventions $= 2.3$

coh.	**syn.**	**vocab.**	**phras.**	**gram.**	**con.**
3.5	3.0	3.5	3.5	3.0	3.5
3.5	3.5	3.5	3.5	3.5	3.5
3.5	3.5	3.5	3.5	3.5	3.5
3.5	3.5	3.5	3.5	3.5	3.5
3.5	3.5	3.5	3.5	3.5	3.5

Table 5. Table of predicted scores on the first 5 essays by the LSTM model on the original data. The predictions are close to the mean of the measures and this can be seen for the predictions on the entire dataset. The abbreviations were coh.:cohesion, syn.:syntax, vocab.:vocabulary, phras.:phraseology, gram.:grammar, con.:conventions

4.2. BERT-Base-Uncased

The result for BERT base uncased model's result was significantly better than the result from regular LSTM model. the predicted measures' range has increased and become much closer to the ground truth values. Although

coh.	syn.	vocab.	phras.	gram.	con.
2.5	2.5	2.5	2.5	2.5	2.5
2.5	2.5	2.5	2.5	2.5	2.5
2.5	2.5	2.5	2.5	2.5	2.5
2.5	2.5	2.5	2.5	2.5	2.5
2.5	2.5	2.5	2.5	2.5	2.5

Table 6. Table of predicted scores on the first 5 essays by the LSTM model on the transformed training data. The abbreviations were coh.: cohesion, syn.:syntax, vocab.:vocabulary, phras.:phraseology, gram.:grammar, con.:conventions

BERT base uncased model out performed the LSTM model, we still found that the model didn't evaluate the gibberish text e.g."xyz" very well. We argue that this is because the training sentences were all relatively long as shown in figure 4 and the training sentences didn't have evaluation scores close to 1 as shown in figure 2. Thus, the gibberish texts were evaluated to have higher scores than they supposed to be. some of the gibberish evaluation has been shown in Table 7.Theoretically, the short gibberish text should be evaluated to score 1 across all six criteria. In order to solve this problem, we appended more gibberish text into the training data with all six evaluation scores set to 1. We retrained the BERT base uncased model on the new data set with the same hyperparameters. After training the BERT base uncased model on the new data set, the evaluation result on the same gibberish text has become more accurate as shown in Table 8.

text	coh.	syn.	vocab.	phras.	gram.	con.
xyz	1.5	1.5	1.5	1.5	2.0	2.0
svem	1.5	2.0	1.5	1.5	2.0	2.0
vmapb	1.5	1.5	1.5	1.5	2.0	2.0
lserm	1.5	1.5	2.0	2.0	2.0	2.0
invew	2.0	1.5	1.5	1.5	2.0	2.0

Table 7. Table of predicted scores on 5 gibberish sentences by the BERT uncased model. The abbreviations were coh.:cohesion, syn.:syntax, vocab.:vocabulary, phras.:phraseology, gram.:grammar, con.:conventions

text	coh.	syn.	vocab.	phras.	gram.	con.
xyz	1.0	1.0	1.0	1.0	1.0	1.0
svem	1.0	1.0	1.0	1.0	1.0	1.0
vmapb	1.0	1.0	1.0	1.0	1.0	1.0
lserm	1.0	1.0	1.0	1.0	1.0	1.0
invew	1.0	1.0	1.0	1.0	1.0	1.0

Table 8. Table of predicted scores on 5 gibberish sentences by the BERT uncased model after training with new data. The abbreviations were coh.:cohesion, syn.:syntax, vocab.:vocabulary, phras.:phraseology, gram.:grammar, con.:conventions

5. Conclusion

In this project we intended to create an automated system for grading essays of English Language Learners using six measures. We used a LSTM model as our baseline and achieved a MCRMSE score of 0.6314 on the training data and 0.6460 on the validation data. We found that the LSTM was highly biased towards the mean of the scores as it predicted close to the mean every time. As a result we decided to train a transformed based BERT model achieving much better results of 0.0453 on the training set and 0.1280 on the validation set. We also acknowledged certain problems that were in the training data and decided to synthesize some data in order to make the BERT model more robust to gibberish inputs like "xyz." Additionally, due to restrictions in time and compute power we decided to leave hyperparameter tuning and other possible fine-tuning methods as a future work.

6. Appendix

You can find our demo of the BERT-based essay grader hosted on HuggingFace Spaces. All the training and data exploration code used for this project can be found on this github repository.

References

[1] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov. Enriching word vectors with subword information, 2016.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding, 2018.

[3] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, nov 1997.

[4] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization, 2014.

[5] I. Loshchilov and F. Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2016.

[6] J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[7] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *CoRR*, abs/1910.10683, 2019.

[8] K. Taghipour and H. Ng. A neural approach to automated essay scoring. 11 2016.