A Two-stage Deep Neural Network for Automated Essay Scoring

Sharon Wu, Candice Sener,
Vish Pillai
UC Berkeley, Berkeley, CA

ABSTRACT

Our project aims to develop AES by using BERT base and BERT-LSTM models in order to observe whether an RNN layer is needed for an effective two-stage learning framework. We found that our BERT base model is competitive to the original TSLF-1[6]. On another note, our implementation of clique-based coherence based on BERT's sentence pairing feature was ineffective in capturing coherence of longer passages. Overall, we found that TSLF-ALL demonstrated robustness with small sample sizes and adversarial samples.

INTRODUCTION

Currently, the assessment of essays is extremely time consuming and expensive as teachers spend hours on grading essays individually. Scoring automation can help to increase time efficiency and cost reduction and decrease potential grader biases. In addition, automated essay scoring (AES) can be applicable for the standardized tests environments such as the SAT or ACT.

The present research has relied heavily on fully designed features to evaluate and score essays.[1] In order to achieve a highly accurate AES system, we attempt to incorporate a two-stage learning framework, combining advantages of both feature-engineered and neural network models. Another consideration for a two stage model is to detect adversarial essays that are either off prompt or incoherent from sentence-to- sentence may bypass certain AES systems.

PROBLEM STATEMENT

First, we attempt to understand whether using solely a self-attention framework can improve the two-stage learning methodology. We will compare creating a BERT baseline fine-tuning model with a hidden layer, dropout, and sigmoid versus utilizing an LSTM as an additional layer and a preliminary step to map essays to low-dimensional embeddings prior to the hidden layer. In other words, we challenge the notion that additional RNN layers are necessary for our AES first stage model architecture.

Then, we also attempt to understand whether modeling local coherence using a clique-based methodology with BERT's sentence pair feature can effectively detect adversarial essays.

RELATED WORK

There have been several approaches to AES systems. One of the earliest projects was Project Essay Grade, in which they used linear regression over vectors of textual features to predict the scores of the essays. Also, there was another work, Bayesian Essay Test Scoring, where they used multinomial and Bernoulli Naive Bayes models in order to classify the texts based on its style features. In addition, there has been a project where they used both linear and logistic regression and features were based on the opinion expressions to develop models in order to assign a grade to the essays.[6] Automated Essay Scoring based on Two-Stage Learning" paper interesting where they have developed a Two Stage Learning Framework in order to utilize both feature-engineered and end-to-end AES methods. The first stage of the framework consisted of calculating the semantic, coherence and prompt-relevance scores, and during the second stage they focused on combining all three scores with handcrafted features to train the model further. [2]

Dataset

The dataset we used for the models is provided by The Hewlett Foundation: Automated

Essay Scoring for the Kaggle competition (https://www.kaggle.com/c/asapaes/data). There are 8 different sets of essays where some of them are dependent on a certain topic whereas others are general writing, and selected essays range from an average length of 150 to 550 words. The essays were written by students ranging from 7th grade to 10th grade, and scores for the essays range from 0 to 6.

Dataset Description

Before we dive deeper into the project, we have conducted some EDA to understand our dataset. For the essay_id, essay_set, domain1_score, we validated that there were no missing values.

In the figure below, we can see that there is a positive correlation between the essay domain1

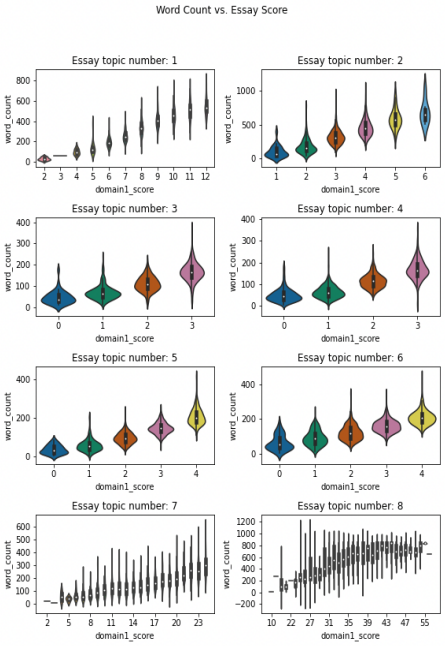


Figure 1: Length versus Essay Score partitioned by Prompt

score and the number of words in the essays. The positive correlation varies slightly for each prompt. Prompt 2 and 8 appear to have essays with longer average word counts compared to other prompts, which may affect the performance of our BERT models due to the 512 token limit.

DATA PROCESSING

Raw dataset is in tsv format where each row represents a single entry. The columns include essay_set, essay text, rater domain scores, and domain1 score.

In the original dataset, scores in different essay sets have different ranges. For consistency, the scores in the first stage are normalized to the range (0,1). Also, we removed anonymous entities such as "@person" and "@organization" along with stopwords, and single letter words for some specific feature engineering.

APPROACH

First stage

Semantic Score

We believe that semantic analysis would help us understand and interpret the meaning of the essay. To achieve this goal, we used the below methods (LSTM, BERT) for generating a semantic score for each essay.

LSTM

We created a semantic score based on Long Short-Term Memory (LSTM) neural network. We first generated the input tensors by tokenizing the text of each essay and truncating the longer sequence while padding the shorter sequences to the given maximum sequence length (200) using BERT tokenizer. Then we run those tensors through BERT, and collect the last hidden state to define the essay's representations. Those essay's representations then were fed into a dense layer in LSTM to transform the low-dimensional vector into a scalar value.

LSTM Training

To train the LSTM model, we used the parameters as seen in Table 2 with ReLU function to minimize the mean squared error. Since the test set from the original dataset is without any label, We used 5-fold cross validation and measured the loss for each fold.

Table 1: K-Fold Cross Validation for LSTM model

Fold No.	Mean Squared Error for Training	Mean Squared Error for Test
1	0.0215	0.0234
2	0.0218	0.0228
3	0.0217	0.0224
4	0.0241	0.0247
5	0.0212	0.0236

We didn't use any early stopping methods, instead, we trained the LSTM with a batch size of 32 and 20 epochs. We notice it shows a fast initial learning then tails off to a much slower reduction after 2nd epochs.

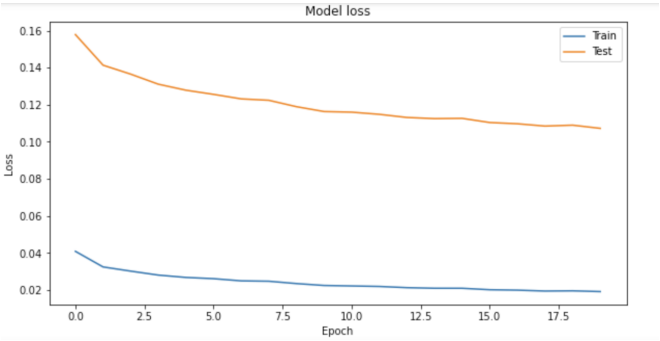


Figure 2: Loss vs Number of Epochs for First Fold of BERT-LSTM model

Bert Fine-Tuning

Another method for creating the semantic score for each essay is through a regression analysis using BERT finetuning. We first perform BERT embeddings, then model evaluation, and CLS token extraction. Then, we use a hidden layer, dropout layer, and then sigmoid activation layer to create a normalized score prediction. Mean squared error is used as our loss function. To verify the training algorithm avoids bias from our imbalanced dataset, we use the class balancing feature from the sklearn library to adjust class weights.

Prompt Relevant Score

In order to understand whether essays adhere to the prompt, we used Cosine Similarity and BERT model to create essay level prompt adherence scores.

Cosine Similarity

Cosine similarity is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. We used this metric to measure how the essay is relevant to the prompt. We calculated cosine similarity by using the en_core_web_lg model from Spacy library, which is trained using Word2ve. Based on this pre-trained model, we can create representative vectors of both essay and prompt by averaging the word-vectors of all the text's words. Later, the two vector representations are compared to produce the cosine similarity. After word vectors are established, we can observe that the essay which has higher Cosine Similarity with the prompt is also assigned a higher score.

Bert Fine-Tuning

Our methodology is structured similarly as the semantic score, though the primary difference is the tokenization step. We use BERT's sentence pairing feature. With the prompt-relevant score, we attempt to understand how each essay pairs with its respective prompt. To train our model to flag adversarial and prompt irrelevant essays, we match random essays with other random prompts. Also, we attempt to ensure the amount of samples in each essay set is consistent for the generation of each

adversarial essay. We assign the gold scores of all adversarial essays to zero.

Figure 3: Prompt Relevance Generation Using BERT Sentence Pairing Feature

Coherence Score

Local Coherence

We again utilize BERT's sentence pairing feature but rather to implement the clique strategy of local coherence (LC). This involves creating windows of sentence pairs for tokenization, referred to as cliques[5], for the embeddings of each essay. Then, we assign the gold score of the essay to each clique and utilize mean squared error as our loss function. For adversarial samples, we permute the order of sentences and assign a gold score of zero to each clique. The final local coherence score is the product of all cliques of an essay. Thus, any tangent within an essay will have a greater effect on the total local coherence metric for the essay.

Global Coherence

Our methodology is again structured identically to the semantic score for BERT finetuning. We consider the value obtained from this model to be a "global" coherence since we create an embedding for all sentences of a particular essay, which differs greatly from the local coherence model. In fact, the approach is similar to the coherence metric obtained in the original two-stage learning paper. [1] The global coherence (GC) metric can be extracted by including adversarial samples, using permuted sentences and assigning a gold score of zero.

Table 2: Model Hyperparameter Tuning

Model	Batch Size	Hidden Units	Dropout Rate	Learning Rate
Semantic - BERT	10	100	0.3	1.00E-05
Semantic - LSTM	32	300	0.5	2.00E-05
Prompt	10	300	0.5	3.00E-05
Local Coherence	10	500	0.3	3.00E-05
Global Coherence	10	500	0.3	1.00E-05

Handcrafted features

Along with those three scores, we also generated some handcrafted features, which are listed below.

- Number of Incorrect & Correct spelled word; Spellchecking is performed using Enchant API
- Lexical diversity; The ratio of total words to unique words
- Number of nouns, verbs, adverbs and adjectives

Second stage

During the second stage, we implement a multi-class classification model - XGBoost, in which the labels will use the original domain score ranges for each prompt. Equation (1) shows how the overall essay score is created. We reported the Quadratic Weighted Kappa scores (QWK) for each prompt, and averaged the scores to evaluate the overall performance. [6]

$$O_e = XGBoost\left(\left[H\,;\, S_e\,;\, P_e\,;\, GC_e\,;\, LC_e\right]\right) \quad (1)$$

DISCUSSION

We will analyze how our models perform directly on the ASAP dataset and synthetic dataset, containing adversarial samples. Consequently, our TSLF-ALL model only will contain coherence and

prompt scores, as referenced in Equation 1, in our adversarial section.

Non-Adversarial Essay Analysis TSLF-1 BERT baseline model performed better than TSLF-1 from Liu, Zhu, Xu[6] for six out of the eight prompts. This result appears to be consistent with the original paper on LSTM additional layers on BERT[8], in which an additional RNN layer may not be necessary for BERT fine-tuning. For GPU memory issues, we were forced to truncate the BERT embedding length to only 200 tokens for our LSTM model, which most certainly impacted its performance. A more robust hyperparameter search would also achieve a better performance.

Table 3: Non-Adversarial Essay Evaluation in QWK

Prompt	TSLF-1: BERT	TSLF-1: (Liu, Zhu, Xu)[6]	TSLF-1: LSTM	TSLF-2	TSLF-ALL: BERT	TSLF-ALL: LSTM	TSLF-ALL: LSTM + Cosine Sim.
1	0.783	0.757	0.595	0.802	**0.813**	0.801	0.804
2	0.674	0.698	0.564	0.634	**0.691**	0.667	0.673
3	**0.753**	0.725	0.752	0.642	0.728	0.729	0.722
4	0.832	0.796	0.832	0.679	0.828	0.821	0.828
5	**0.824**	0.810	0.793	0.787	0.816	0.808	0.805
6	**0.845**	0.783	0.791	0.627	0.834	0.792	0.783
7	0.778	0.727	0.745	0.656	**0.782**	0.754	0.743
8	0.443	0.544	0.264	0.523	**0.55**	0.504	0.486
Avg.	0.742	0.730	0.667	0.669	**0.755**	0.735	0.731

*Note: **LSTM** represents the model involving pretrained BERT embeddings processed through an LSTM layer. **BERT** represents the BERT baseline model. **TSLF-1** represents only the semantic score prediction. **TSLF-2** utilizes only handcrafted or engineered features. **TSLF-ALL** utilizes only the semantic score and handcrafted features.*

In addition, TSLF-ALL performs much better than TSLF-1 and TSLF-2, which demonstrates that the performance of AES can benefit from integrating both deep-decoded features and handcrafted features. The model with Cosine Similarity with Word2vec had lower accuracy. This is likely because the essay vector is computed as an average of all word vectors and tends to give too much weight to words that are irrelevant. Also, the approach of taking the average word vector across the sentence ignores the context in which words appear and perhaps generalises the sentences.

Adversarial Essay Analysis To test the quality of our coherence and prompt metrics, we created an adversarial dataset composed of a combination of prompt-irrelevant, permuted, and random standard essays. For every random standard essay in our ASAP dataset, we included a randomly chosen permuted essay and prompt-irrelevant essay. Rather than intentionally selecting corresponding essays to permute, we instead randomly chose an essay in our training data to permute for each standard essay, as opposed to the two-stage paper methodology[6]. Another condition in our synthetic dataset was that the total number of essays per prompt was consistent.

Table 4: Adversarial Essay Evaluation in QWK

Adversarial Dataset	TSLF-1: BERT	TSLF-2	TSLF-ALL: BERT
ADV1	0.029	0.033	**0.858**
ADV2	0.187	0.535	**0.951**
ADV1+ADV2	0.001	0.002	**0.885**

*Note: **ADV1** represents the dataset of permuted essays and **ADV2** represents the dataset of prompt irrelevant essays, while **ADV1+ADV2** is their combination. **TSLF-1 BERT** is our end-to-end semantic prediction baseline BERT model. **TSLF-1** represents only the semantic score prediction. **TSLF-2** utilizes only handcrafted or engineered features. TSLF-ALL utilizes all features: semantic, handcrafted, local coherence, global coherence, and prompt relevance.*

With over a 0.80 QWK baseline improvement on ADV1+ADV2, the results in Table 4 signify the robustness of our TSLF-ALL model, in which the local coherence, global coherence and prompt-relevant scores are all utilized. Our handcrafted features in TSLF-2 perform unexpectedly well on ADV2, which may ultimately be due to half the samples being ASAP standard. The prompt-relevance score outperforms the baseline by over 0.40 QWK, certainly due to detection of prompt-irrelevant essays being easier than that of permuted essays. As a sanity check, the 95th percentile of the prompt-relevance score on prompt-irrelevant essays is around 0.02 from Appendix 2A and 2C. For coherence, the generation of one random permuted sample per random ASAP essay during training rather than multiple permutations may be the primary

explanation for the high coherence performance of TSLF-ALL compared to the original clique-based LC paper.[5]

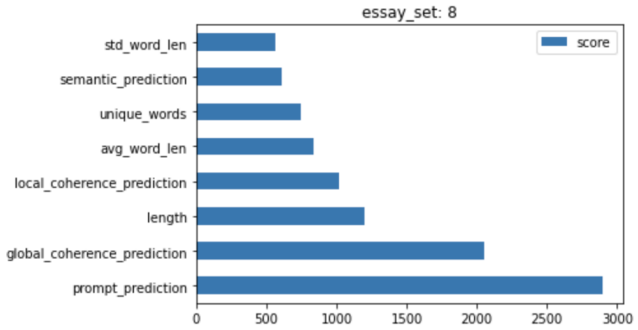


Figure 4: Feature Importance of Prompt 8 for Adversarial Samples

Overall, our LC score is a decent feature for our XGBoost model, but it appears to be only on the same tier as the handcrafted features for all prompts. Figure 4 displays prompt 8 as an example. Unfortunately, our expectation that the local coherence would perform better than global on Prompt 8 due to longer sequences was proven untrue. This may be rationalized by the fact that a 2-clique strategy is a fundamentally narrow approach to understanding overall essay coherence. Greater *n*-clique strategies should be utilized for future considerations of this project. Though, BERT was not pre-trained to process multiple *[SEP]* tokens. Other transformer architectures and strategies should be explored, even excluding *[SEP]* tokens between sentences altogether.

CONCLUSION

In general, we observed that TSLF-ALL achieves the highest average performance and is more robust when dealing with small sample sizes and adversarial samples. We observed the vast potential of neural network architectures in solving automated essay scoring tasks. In adding the adversarial samples, the TSLF-ALL remained stable which shows our coherence score and prompt relevant score are valid and effective for improving the AES systems.

REFERENCES

1. A text categorization approach to automated essay grading. (2003). *Automated Essay Scoring*,67–82. https://doi.org/10.4324/9781410606860-13
2. Alikaniotis, D., Yannakoudakis, H., & Rei, M. (2016). Automatic text scoring using neural networks. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. https://doi.org/10.18653/v1/p16-1068
3. Aroraaman. (2018, December 30). *Quadratic Kappa Metric explained in 5 simple steps*. Kaggle. https://www.kaggle.com/aroraaman/quadratic-kappa-metric-explained-in-5-simple-steps.
4. Bayes' theorem. (n.d.). *SpringerReference*. https://doi.org/10.1007/springerreference_81597
5. Farag, Y., Yannakoudakis, H., & Briscoe, T. (2018). Neural Automated Essay Scoring and Coherence Modeling for Adversarially Crafted Input. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. https://doi.org/10.18653/v1/n18-1024
6. Norming and scaling for automated essay scoring. (2019). *Automated Essay Scoring based on Two Stage Learning*, 177–187. https://doi.org/10.4324/9781410606860-20
7. Taghipour, K., & Ng, H. T. (2016). A neural approach to automated essay scoring. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. https://doi.org/10.18653/v1/d16-1193
8. Adam Thorne, Zac Farnsworth, Oscar Matus, Research of LSTM Additions on Top of SQuAD BERT Hidden Transform Layers, https://web.stanford.edu/class/archive/cs/cs224n/cs224n.1194/reports/default/15718571.pdf

APPENDIX

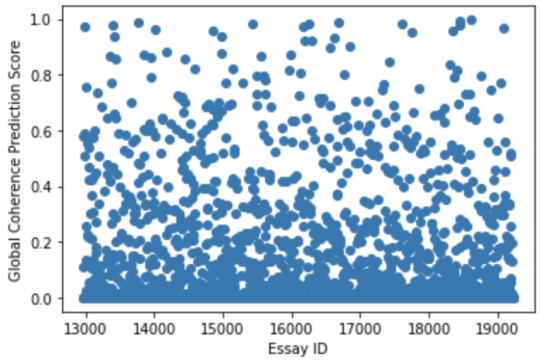
Appendix 1: Training/Validation Metrics for BERT Regression Models

Model	Training MSE	Training MAE	Validation MSE	Validation MAE
Semantic	0.0019	0.1071	0.0174	0.101
Prompt	0.001	0.0765	0.0159	0.0698
Local Coherence	0.0057	0.1739	0.0634	0.1682
Global Coherence	0.0017	0.1094	0.0396	0.1088

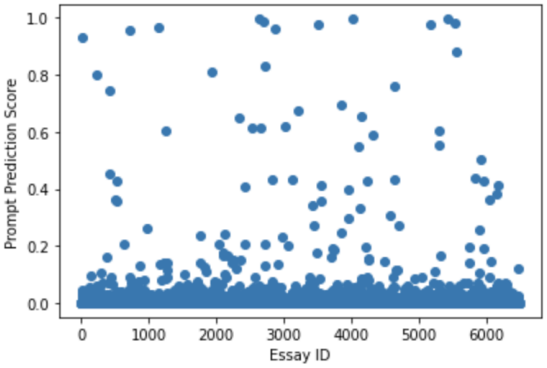
Appendix 2A: Statistics on Prediction Scores for Adversarial Samples

Percentile Statistics	Prompt Relevant	Local Coherence	Global Coherence
count	6487	6246	6246
mean	0.00978	0.01177	0.04811
std	0.06102	0.03619	0.14137
min	0.00000	0.00000	0.00001
25%	0.00001	0.00000	0.00004
50%	0.00015	0.00003	0.00016
75%	0.00167	0.00382	0.00402
90%	0.01275	0.03248	0.15163
95%	0.02897	0.06980	0.34423
99%	0.19584	0.18030	0.71718

Appendix 2B: Global Coherence Prediction Scores for Adversarial Samples



Appendix 2C: Prompt Relevance Prediction Scores for Adversarial Samples



Appendix 2D: Local Coherence Prediction Scores for Adversarial Samples