GAN-BERT for Automated Essay Scoring

Stanford CS224N Custom Project

Griffin Holt
Department of Electrical Engineering
Stanford University
gholt@stanford.edu

Theodore Kanell
Department of Computer Science
Stanford University
tkanell@stanford.edu

Abstract

Every year, millions of individuals take English language proficiency exams, such as TOEFL and IELTS, for professional and academic development. These exams are typically graded by human evaluators; automating the evaluation process can improve both efficiency and fairness of the examinations. Our approach to the Automated Essay Scoring (AES) task is to implement three variations of the GAN-BERT architecture: a feed-forward neural network generator; a BERT transformer generator; and a generator composed of a fine-tined GPT2 language model in tandem with a BERT transformer. We use a single pre-trained RoBERTa model, fine-tuned to our task and dataset, for a baseline comparison. All three GAN-BERT architectures outperformed the baseline model on the test set. The GAN-BERT models are also able to better differentiate between Low and Medium score essays, and Medium and High score essays. The GPT2-BERT generator demonstrated the most evidence of taking advantage of the competitive nature of the GAN structure to improve both generator and discriminator.

Key Information: Our TA Mentor is Abhinav Garg.

1 Introduction

Essay scoring for standardized examinations can be an arduous and subjective process, often involving multiple graders whose respective scores are averaged to produce a single final essay examination score. Grading exam essays can also place a difficult burden on educators, exacerbated especially by rapid increase of online education. The global English language learning market, in particular, is expected to reach $69.62 billion by 2029 with a CAGR of 9.5% until that year (Research and Markets, 2022); for comparison, the smartphone market expects a CAGR of only 7.3% during that same time (Fortune Business Insights, 2022). Applying recent improvements in NLP to the task of essay scoring can give English learners a rapid, consistent metric for their essays, especially for tests like the IELTS or TOEFL, while relieving a burden of overworked and underpaid educators.

Formally, the Automated Essay Scoring (AES) task is defined as follows: given an essay with m words $X = \{x_i\}_{i=1}^{m}$, we want to output a single score y that reflects the measure of the essay. For the ETS Corpus of Non-Native Written English dataset (Blanchard, Daniel et al., 2014) utilized in this paper, the score range is constrained to only $|S| = 3$ categories (Low, Medium, and High) and we therefore frame our specific problem as a classification task.

Previous attempts to create an effective and accurate AES system followed two basic designs: deep neural network models using either LSTM or CNN architectures using factors such as word length, spelling errors, or bag of words to featurize essays in a time consuming procedure (Rodriguez et al., 2019); and transformer-based models, such as BERT (Wang et al., 2022; Dong et al., 2017).

In this paper, we extend the GAN-BERT architecture–a unique adaptation of the Generative Adversarial Network (GAN) (Goodfellow et al., 2020) that incorporates a BERT transformer and was first introduced by Croce et al. (2020) for various NLP tasks–to the Automated Essay Scoring task. We anticipate that the GAN-BERT architecture will help our model be more robust in scoring across a variety of different prompts. We therefore do not create separate models for each prompt, but we instead utilize a general model for all prompts in our data set.

2 Related Work

2.1 AES Research

Deep Neural Networks using LSTM or CNN architectures have produced excellent models for AES and are able to automatically learn many intricate features of essays, and therefore require less pre-computation to generate and design features for the essays (Taghipour and Ng, 2016). However, best results are obtained by incorporating work intensive handcrafted features (Uto et al., 2020).

Pre-trained language models such as BERT are able to reach state of the art results, with three papers out performing other deep learning models. All three papers employed additional training optimization. Cao et al. (2020) utilized domain adversarial training, Yang et al. (2020) combined regression and ranking for training, and Wang et al. (2022) employed three different levels of granularity to encapsulate the essay for the model.

2.2 GAN-BERT

Very few researchers have applied GAN networks to NLP tasks and none have applied it to the AES task. Croce et al. (2017) employed a kernel-based GAN which combined expressive kernels and deep neural networks to model structured information and learn non-linear decision functions. Croce et al. (2017) was able to achieve state-of-the-art results in Question Classification, Community Question-Answering, and Argument Boundary detection. Croce et al. (2020) demonstrated that applying a semi-supervised GAN on a NLP task can enable the model to achieve high results with far fewer labeled data points on Sentiment Classification.

3 Approach

We proceed to describe our approach to this task for our novel models–several variations on the GAN-BERT architecture first proposed by Croce et al. (2020)–as well as our approach for the baseline model–a single pre-trained RoBERTa (Liu et al., 2019) model fine-tuned to our classification task.

3.1 GAN-BERT Architecture

The GAN-BERT architecture is an adaptation of the Generative Adversarial Network structure (Goodfellow et al., 2020) that incorporates a BERT transformer and is thus more optimized for NLP tasks. In the Generative Adversarial Network architecture for a classification task, a discriminator is trained over $(K + 1)$ classes: its goal is for real examples to be classified into one of the target categories $\{1, \ldots, K\}$ and for fake or generated examples to be classified as class $K + 1$. A generator is then trained to generate fake examples that deceive the discriminator.

We will now describe the GAN-BERT architecture (see Figures 1) more formally, illustrating how it combines the traditional GAN architecture and loss functions with a BERT model to address NLP tasks. Let G denote the generator network and D denote the discriminator network. Let $X_k = \{x_i\}_{i=1}^{m}$ represent a real essay from our dataset (see Section 4.1 for details regarding the

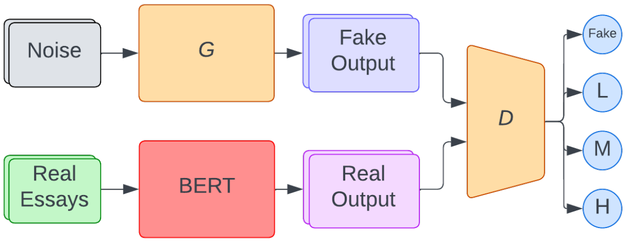


Figure 1: The GAN-BERT architecture, as described by Croce et al. (2020), but changed to fit our use case: the size-constrained essays are passed into the BERT transformer module, and the discriminator D outputs the essay score $\hat{y} \in \{\text{Fake}, \text{Low}, \text{Medium}, \text{High}\}$, where $\hat{y} = \text{Fake}$ signifies D identified the input as generated the generator G.

content and distribution of the essays). Each essay X_k is labeled with a human-evaluated score $y_k \in \{\text{Low}, \text{Medium}, \text{High}\} = \mathcal{S}$.

An essay X_k is first tokenized using WordPiece (Devlin et al., 2018) with a maximum length of L tokens. The token sequence is truncated if its length exceeds L and padded if its length is less than L. The tokenized sequence $T = [t_1, t_2, \ldots, t_L]$ is then passed into a pre-trained BERT module which we also fine-tuned in advance on the AES task. As suggested by Devlin et al. (2018), we utilize the CLS hidden state h_{CLS} as our single vector output $v_B \in \mathbb{R}^{768}$ from the BERT module.

The BERT output v_B is then passed into the discriminator network D. For our specific implementation of the GAN-BERT architecture, the discriminator D (see Figure 2d) is a feed-forward neural network composed of (in order) a dropout layer with dropout probability p; a hidden linear layer with an output dimension of 768; a LeakyRELU activation function; an additional dropout layer with dropout probability p; an output linear layer with an output dimension of $|S| + 1 = 4$; and a softmax layer, which outputs the final probabilities for each of the $|S| + 1$ classes: $\{\text{Fake}, \text{Low}, \text{Medium}, \text{High}\}$. The final output of the discriminator D is the predicted class $\hat{y}_k$, the class having maximum probability from the softmax layer.

Separately, “noisy input" is passed into the generator G. For the AES task, we experiment with three different generator structures, each of which are described in detail in Section 3.1.2. The exact definition of “noisy input" depends on the structure of the generator itself. Regardless of its internal structure, the generator G produces an output $v_G \in \mathbb{R}^{768}$: a “fake" sample that, ideally, mimics the output v_B of the BERT module when fed a real essay X. This generator output v_G is then passed into the discriminator D and assigned a class probability score and final prediction $\hat{y}_{\tilde{k}}$.

3.1.1 Loss Functions

Let y denote the true class label for a real essay X. Let v be a generic input to the discriminator D (i.e., v may come from a real essay X processed by BERT or from the generator G). Let p_G be the distribution of inputs v generated from the generator G. Let p_B be the distribution of inputs v produced by the BERT module processing a real essay. Let $p_D(\hat{y} = y|v, y = 0)$ be the probability that an input v associated with the fake class is classified by the discriminator D as fake. Let $p_D(\hat{y} = y|v, y \in \{1, 2, 3\})$ be the probability that an input v associated with one of the real essay scores is classified with the correct essay score.

The discriminator loss function ℓ_D is designed to motivate the discriminator to both differentiate between real inputs v_B and fake inputs v_G and assign a correct essay score $\hat{y}_k \in \{\text{Low}, \text{Medium}, \text{High}\}$ to real inputs v_B. The discriminator loss function ℓ_D is given as $\ell_D = \ell_{D_{\text{Score}}} + \ell_{D_{\text{RF}}}$, where

$$\ell_{D_{\text{Score}}} = -\mathbb{E}_{v,y\sim p_B}\left[\log p_D(\hat{y} = y|v, y \in \{1, 2, 3\})\right] \tag{1}$$

measures the discriminator's error in score classification for a real essay X; and

$$\ell_{D_{\text{RF}}} = -\mathbb{E}_{v,y\sim p_B}\left[\log\left(1 - p_D(\hat{y} = y|v, y = 0)\right)\right] - \mathbb{E}_{v,y\sim p_G}\left[\log p_D(\hat{y} = y|v, y = 0)\right] \tag{2}$$

measures the discriminator's error in misclassifying real examples as fake and fake examples as real.

In contrast, the generator loss function ℓ_G is designed to motivate the generator G to generate discriminator inputs v_G that are similar to the inputs v_B from the distribution of real examples p_B. Let $f(v)$ be the activation of the hidden layer in the discriminator D for a given input v. Then, to encourage the generator G to produce outputs v_G statistically similar to the BERT's outputs v_B, we define the *feature matching* generator loss to be

$$\ell_{G_{\text{feature matching}}} = \left\|\mathbb{E}_{v\sim p_B}\left[f(v)\right] - \mathbb{E}_{v\sim p_G}\left[f(v)\right]\right\|_2^2. \tag{3}$$

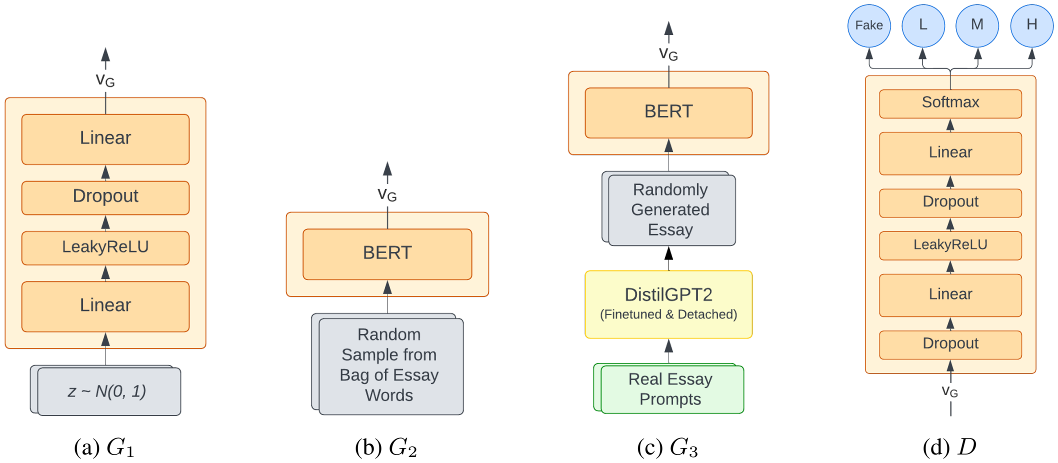
This feature matching loss technique was suggested by Salimans et al. (2016) for traditional GANs and implemented by Croce et al. (2020) in their original GAN-BERT architecture. The complete generator loss function ℓ_G is then given by $\ell_G = \ell_{G_{\text{feature matching}}} + \ell_{G_{\text{caught}}}$, where

$$\ell_{G_{\text{caught}}} = -\mathbb{E}_{v,y\sim p_G}\left[\log\left(1 - p_D(\hat{y} = y|v, y = 0)\right)\right] \tag{4}$$

directly penalizes the generator G for producing fake examples that were “caught" (identified as fake) by the discriminator D. In a code implementation, all expectations (for ℓ_D and ℓ_G) are taken empirically.

3.1.2 Generators

We will now describe the structures of the three variants of generators G_1, G_2, G_3 that we implemented to address the AES task. **We note that the G_1 generator, described below, was used by Croce et al. (2020) in the original GAN-BERT paper, but G_2 and G_3 were our original contributions. We wrote all code for this project–including the implementation of G_1–from scratch, referring only to the code from Croce et al. (2020) when mathematical details were missing from their paper.** We will use the terms G_1, G_2, and G_3 to refer to both the individual generators and the entire GAN-BERT model (the generator combined with its discriminator).



(a) G_1 (b) G_2 (c) G_3 (d) D

Figure 2: The three generator architectures–the Neural Network Generator G_1; the BERT Generator G_2; and the GPT2-BERT Generator G_3–and the Discriminator D architecture

G_1: A Feed-Forward Neural Network

The first generator variant G_1 (see Figure 2a) is a feed-forward neural network (FFNN) composed of (in order) a linear layer with an input dimension of $d_i = 100$ and output dimension of $d_h = 768$; a LeakyRELU activation function; a dropout layer with dropout probability p; and an output linear layer with an output dimension of $d_o = 768$. The FFNN generator G_1 takes as input a noisy vector $z \in \mathbb{R}^{100}$, $z_i \sim \mathcal{N}(0, 1)$ whose inputs are generated from the standard Gaussian distribution. G_1 then outputs a vector $v_G \in \mathbb{R}^{768}$ to be passed to the discriminator D.

G_2: A BERT Generator

The second generator variant G_2 (see Figure 2b) is a single pre-trained BERT transformer module (Devlin et al., 2018). Note that this BERT transformer module is separate from the BERT transformer module which processes real essays (pictured in red in Figure 1). Whereas the parameters of the BERT transformer module processing real essays are frozen and detached from the gradient, the parameters of BERT transformer module composing G_2 are updated according to the loss function ℓ_G.

To create a "noisy input" to feed into the G_2 BERT transformer module, we first created a "bag of words" from all words present in essays in the training set. A random essay $\tilde{X}$ was then generated by selecting $L = 510$ words according to the frequency with which they are present in the training set essays. Note that each word in $\tilde{X}$ was selected independently from the same distribution: no attempt was made at this point to force the words in $\tilde{X}$ to form a cohesive sentence. This random essay $\tilde{X}$ was then tokenized by the WordPiece BERT tokenizer and the resulting token sequence $\tilde{T}$ was fed into G_2. Similar to the other BERT module, the output $v_G \in \mathbb{R}^{768}$ from G_2 to be fed into the discriminator is the CLS hidden state h_{CLS}.

G_3: A DistilGPT2-BERT Generator

The third generator variant G_3 (see Figure 2c) is another single pre-trained BERT transformer module. However, this time, we fine-tuned a pre-trained DistilGPT2 language model (Sanh et al., 2019) to generate a fake essay $\tilde{X}$ when given one of the eight real essay prompts P1, . . . , P8. The DistilGPT2 module was fine-tuned on the real essays X_k in the training set and their respective prompts.

The DistilGPT2-generated essay $\tilde{X}$ was then tokenized and fed into the BERT transformer module. The output $v_G = h_{CLS} \in \mathbb{R}^{768}$ from G_3 is again the CLS hidden state of the BERT module.

Note that only the BERT transformer is connected to the gradient; after fine-tuning the DistilGPT2 language model to generate fake essays from real essay prompts, its parameters are frozen and detached from the GAN-BERT loss functions.

3.2 Baseline Models: RoBERTa

Our baseline model is a single pre-trained RoBERTa (Liu et al., 2019) model fine-tuned to our classification task. We utilize the RoBERTa tokenizer–a byte-level variant of the Byte-Pair Encoding tokenizer (Sennrich et al., 2016)–to split the essay into a token sequence $T = [t_1, t_2, \dots, t_L]$, truncated or padded to a sequence length of $L = 510$. The final input representation is then the sum of the token embeddings, segmentation embeddings, and position embeddings. The RoBERTa model then outputs a logit $l \in \mathbb{R}^N$ from which we can generate an output prediction $\hat{y} = \arg\max_{i=1,\dots,3} l_i$. Our loss function for fine-tuning the pre-trained model to this task is Cross Entropy Loss.

The pre-trained RoBERTa parameters were downloaded from HuggingFace, but all other parts of the approach described above were implemented by us from scratch with pertinent libraries.

4 Experiments

4.1 Data

Our models (baseline and GAN-BERT) were trained on the ETS Corpus of Non-Native Written English (Blanchard, Daniel et al., 2014), a compilation of 12,100 English essays written by speakers of 11 non-English native languages (1,100 essays for each language) across 8 different essay prompts as part of the international academic English language proficiency exam, TOEFL. The dataset was developed specifically for native language identification, but, as acknowledged by its authors, can be used for other tasks (such as AES).

As stated earlier, each essay X_k is labeled with a human-evaluated score $y_k \in \{\text{Low}, \text{Medium}, \text{High}\} = \mathcal{S}$. The training set is composed of $n = 9900$ essays, and the development and test sets are each composed of $\tilde{n} = 1100$ essays. The distribution of essays prompts P1, . . . , P8 and score categories Low, Medium, Low for the training, development, and test sets are presented in Table 1 and Table 2, respectively.

Prompt	**Frequencies**		
	Train	*Dev*	*Test*
P1	0.1383	0.1382	0.1227
P2	0.1312	0.1091	0.1300
P3	0.1168	0.0845	0.1336
P4	0.1222	0.1282	0.1436
P5	0.1382	0.1527	0.1018
P6	0.0783	0.0645	0.1036
P7	0.1383	0.1645	0.1236
P8	0.1368	0.1582	0.1409

Table 1: Distribution of essay prompts for the training, development, and test sets

Score	**Frequencies**		
	Train	*Dev*	*Test*
Low	0.1080	0.1200	0.1173
Medium	0.5420	0.5436	0.5491
High	0.3500	0.3364	0.3336

Table 2: Distribution of essay scores for the training, development, and test sets

Because of the imbalance of the three score classes in the dataset, we also experimented with incorporating class weights $w_i, i = 1, \dots, |S|$ into our loss functions for both the GAN-BERT architecture and our RoBERTa baseline modules. The weight w_i for class $i \in S$ is given by

$$w_i = \frac{n}{|S| \sum_{k=1}^{n} \mathbf{1}\{y_k = i\}}. \tag{5}$$

Incorporating the class weights into the GAN-BERT loss functions simply changed the computation of $\ell_{D_{\text{Score}}}$ to a weighted empirical mean. To incorporate the class weights into the baseline RoBERTa model, we simply used Weighted Cross Entropy Loss.

4.2 Evaluation Method

Our primary evaluation metric for the performance of our models on the AES task is the Quadratic Weighted Kappa (QWK) score (Cohen, 1968). This score is frequently used to compare the performance of an automated grading system against human graders and is the standard for AES performance comparison (Wang et al., 2022). The details of computing QWK are outlined in Section 4.2.1 further below.

In addition to measuring the QWK performance of our models on the AES task, we also measured the performance of each GAN-BERT model's discriminator D in identifying “real" versus “fake" (i.e., generated by a generator G) inputs. We measured each model's Real-Fake Classification Accuracy, Real-Fake Precision, and Real-Fake Recall to understand how well each generator G was able to mimic real examples (and thereby, hopefully, improve the discriminator's ability to distinguish between each score class).

4.2.1 Quadratic Weighted Kappa (QWK)

The details of computing QWK are outlined below:

Let $O \in \mathbb{R}^{|S|\times|S|}$ be the confusion matrix associated with the model's score classifications $\hat{y}_k \in \mathcal{S}, k = 1, \ldots, n$ and the actual scores $y_k \in \mathcal{S}, k = 1, \ldots, n$. By convention (Pedregosa et al., 2011), O_{ij} is equal to the number of essays known to have score $y_k = i$ and predicted to have score $\hat{y}_k = j$ by the model. Then, normalize O to get $\tilde{O} = \frac{1}{\sum_{i,j} O_{ij}} O_{ij} \in \mathbb{R}^{|S|\times|S|}$.

Let $W \in \mathbb{R}^{|S|\times|S|}$ be a weight matrix defined entrywise as $W_{ij} = \frac{(i-j)^2}{(|S|-1)^2}$. The weight matrix gives partial credit in the final QWK score to the model for proximity to the correct label (e.g., if the model guessed Medium when the essay was actually labeled High, it is penalized less than if it had classified the essay as Low).

Let $a, b \in \mathbb{R}^{|S|}$ be count vectors defined entrywise such that a_i is the number of essays with an actual score $y_k = i$, and b_i is the number of essays predicted to have score $\hat{y}_k = i$. Then, let $E = ab^T \in \mathbb{R}^{|S|\times|S|}$, and normalize it to get $\tilde{E} = \frac{1}{\sum_{i,j} E_{ij}} E \in \mathbb{R}^{|S|\times|S|}$.

Finally, the Quadratic Weighed Kappa score κ for model performance is given by

$$\kappa = 1 - \frac{\sum_{i,j} W_{ij} \tilde{O}_{ij}}{\sum_{i,j} W_{ij} \tilde{E}_{ij}}. \tag{6}$$

4.3 Experimental details

For each of the GAN-BERT models, we utilized the Adam optimizer with a learning rate of $\lambda = 0.0002$. We trained G_1 for 5000 epochs with a batch size of $B_1 = 100$, G_2 for 20 epochs with a batch size of $B_2 = 24$, and G_3 for 10 epochs with a batch size of $B_3 = 9$. The training process for each GAN-BERT model took between 45 minutes (for G_1) and 2-3 hours (for G_2 and G_3).

For the baseline RoBERTa model, we used the Adam optimizer with a learning rate of $\lambda = 2 \times 10^{-5}$, a batch size of $B = 16$, and a weight decay of $\nu = 0.01$. We were only able to train for 10 epochs as the checkpoints exhausted the memory. Given the significant number of parameters in RoBERTa models, training took about two hours.

For the fine-tuned DistilGPT2 module that fed into G_3, we used the AdamW optimizer with a learning rate of $\lambda = 2 \times 10^{-5}$, a batch size of $B = 9$, and trained for 10 epochs.

For the G_1 GAN-BERT model, we experimented with two different dropout rates $p = 0.1, 0.5$ for the dropout layers. We also experimented with the maximum sequence length $L = 64, 128, 510$ for the tokenized input to the BERT module. For G_2 and G_3, we exclusively used a dropout rate of $p = 0.5$ and a maximum sequence length of $L = 510$. For all three GAN-BERT models, we also experimented with the inclusion and exclusion of class weights w_i in the loss function to account for the class imbalance in our dataset.

Model	**Dropout**	**Class Weights?**	**Maximum Seq. Len.**	**L-M-H QWK**			**R-F Accuracy**			**R-F Precision**			**R-F Recall**		
				Train	*Dev*	*Test*	*Train*	*Dev*	*Test*	*Train*	*Dev*	*Test*	*Train*	*Dev*	*Test*
Worst Case	–	–	–	-0.808	-0.802	-0.800	–	–	–	–	–	–	–	–	–
Random	–	–	–	0.009	-0.014	-0.044	–	–	–	–	–	–	–	–	–
RoBERTa	–	No	–	0.712	0.707	0.675	–	–	–	–	–	–	–	–	–
RoBERTa	–	Yes	–	0.761	0.755	0.709	–	–	–	–	–	–	–	–	–
GAN-BERT G_1	0.1	No	510	0.9488	0.7420	0.6989	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
GAN-BERT G_1	0.1	No	128	0.3780	0.3964	0.3610	0.9219	0.9326	0.9228	0.9999	1.0000	1.0000	0.8439	0.8652	0.8455
GAN-BERT G_1	0.1	No	64	0.0883	0.1050	0.0746	0.8777	0.8813	0.8777	1.000	1.0000	0.9989	0.7553	0.7625	0.7563
GAN-BERT G_1	0.5	No	510	0.8098	0.7752	0.7311	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
GAN-BERT G_1	0.5	Yes	510	0.8008	0.7583	0.7280	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
GAN-BERT G_2	0.5	No	510	0.8003	0.7459	0.7344	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
GAN-BERT G_2	0.5	Yes	510	0.7890	0.7408	0.7169	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000	1.0000
GAN-BERT G_3	0.5	No	510	0.7718	0.7351	0.7192	0.8849	0.8966	0.8869	0.8679	0.8766	0.8614	0.9223	0.9327	0.9345
GAN-BERT G_3	0.5	Yes	510	0.7685	0.7326	0.7179	0.8807	0.8807	0.8848	0.8612	0.8543	0.8624	0.9224	0.9330	0.9342

Table 3: Training, development, and test set evaluation metrics for each of our AES models, including QWK scores on real examples; Real-Fake Classification Accuracy; Real-Fake Precision; and Real-Fake Recall

4.4 Results

The evaluation metrics for each of our models–baseline and GAN-BERT–are presented in Table 3. For comparison, we also present the worst-case QWK scores for our dataset (all Low essays scored as High, all High essays scored as Low, and all Medium essays scored as either Low or High) as well as the QWK scores from assigning a random score to each essay (over 100 trials). Confusion matrices for the non-class-weight baseline RoBERTa model and the best GAN-BERT G_1, G_2, and G_3 models are also presented in Figure 3 (see the Appendix: Section 7).

The RoBERTa baseline model experienced a significant increase in QWK performance with the inclusion of class weights. With the correct set of hyperparameters, each of the GAN-BERT models (G_1, G_2, G_3) offered an improvement in $0.1 - 0.3$ QWK points over the class-weighted baseline RoBERTa model. Notably, class weights did nothing to improve any of the GAN-BERT models.

Overall, the non-class-weighted G_1 and G_2 models acheived the highest QWK scores. However, the GAN-BERT G_3 model had the lowest R-F Classification Accuracies, signifying that the G_3 DistilGPT2-BERT generator produced the most fake examples that deceived the discriminator D. This result was particularly interesting: we knew that G_3 had the most potential for fake essay generation due to the capabilities of GPT2 language models, but we did not expect it to be able to trick the discriminator one out of ten times.

5 Analysis

When we compare our baseline on the ETS Corpus to baselines for different AES datasets (Phandi et al., 2015), we see that our baseline performed similarly: for example, the QWK for our baseline models ($\kappa = 0.675, 0.709$ for the non-class-weighted baseline and class-weighted baseline, respectively) was similar to the EASE model (Phandi et al., 2015) on the ASAP data ($\kappa = 0.675$). However, it is important to note that achieving a higher QWK value is more difficult when the score range is wider: our score range only consists of three possible scores and is therefore not as difficult as the score range for some of the models on other AES datasets (Phandi et al., 2015).

The baseline model performed particularly well at identifying essays in the High category, but it struggled in identifying Low category essays as evidenced in Figures 3a, 3b, 3c. We believe this could be due to the under-representation of Low-scored essays in the ETS Corpus dataset. Including class weights in the loss function for the baseline RoBERTa boosted QWK performance significantly. The baseline model also rarely (and, on the Development dataset, *never*) confused Low and High scored essays. If it confused score categories, it erred on the upward side, mistaking Low for Medium and Medium for High. Both of these trends are desired qualities in an Automated Essay Scorer, which made us optimistic for the performance of our more complex GAN-BERT models.

Varying the maximum sequence lengths L for G_1 offered quantitative insight into the conflict between the G_1 generator and its respective discriminator. A sequence length of $L = 512$ caused the output v_B of the BERT module to be too complex for the feed-forward neural network structure of G_1 to mimic (as evidenced by the 100% R-F Accuracy). On the other hand, a decrease in sequence length to $L = 64$ improved the $G1$ generator's performance (as evidenced by a decrease in R-F Accuracy), but denied the discriminator enough information in such a short token sequence to differentiate between the quality of essay categories (as evidenced by the significant decrease in QWK).

In utilizing a GAN architecture, the hope is that the competition between generator and discriminator will cause both modules to increase in performance. The generator, in mimicking BERT outputs for real essays with labeled scores, might encourage the discriminator to better understand the differences between the score categories. However, both the G_1 and G_2 generators were never able to capture the complexity of BERT's output and definitively failed to produce believable examples. Although the discriminators for G_1 and G_2 produced the highest QWK scores above the baseline, we don't believe that either of these GAN-BERT models were able to take full advantage of their generative adversarial structure.

Fortunately, the G_3 GAN-BERT architecture was able to better capitalize on the competition between generator and discriminator to elevate both modules. The G_3 generator caused a decrease in Real-Fake Accuracy, Real-Fake Precision, and Real-Fake Recall without compromising significant losses in QWK scores on real essays. The confusion matrices in Figure 3 also show us that the G_3 generator produced a discriminator D that was able to differentiate between Low and Medium essays much better than the G_1 or G_2 models. Thus, the essays and fake outputs v_G generated by G_3 enabled the discriminator to better understand what exactly defined a Low-scoring essay.

To understand better how the G_3 generator successfully deceived its discriminator, we examined three examples of essays produced by the DistilGPT2 fine-tuned language model that that were misclassified by the discriminator as real essays with High, Medium, and Low scores, respectively (see the Appendix: Section 7.2.1 for the text and discriminator classification probabilities of these three generated essays). Even from these three examples, it seems that *essay length*, *spelling* (i.e., presence or absence of mispelled words), *the use of smooth transition words and phrases* (e.g., “furthermore", “moreover", “to the contrary", etc.), *sentence variety*, and *vocabulary mastery*–all of which are factors used by human evaluators–may be factors utilized by the G_3 GAN-BERT discriminator in its automated evaluation.

Finally, when we look at the confusion matrices for the baseline model and the three generator architectures (see Figure 3), we see an interesting pattern emerge: the baseline RoBERTA model was the best at identifying High-scoring essays (94% accuracy); the G_1 and G_2 model discriminators were the best at identifying Medium-scoring essays (81-92% accuracy); and the G_3 model was the best at identifying Low-scoring essays (86-94% accuracy). This would suggest that a stacked ensemble or soft-voting ensemble of all four models has potential to achieve higher QWK scores.

6 Conclusions & Future Work

In summary, all three GAN-BERT architectures–the feed-forward neural network generator G_1, the BERT generator G_2, and the DistilGPT2-BERT generator G_3–had better Quadratic Weighted Kappa (QWK) performance for essay scoring than the baseline RoBERTa model. More specifically, all three GAN-BERT architectures improved upon the baseline RoBERTa model in being able to better differentiate between Low and Medium level essays, and between Medium and High level essays. The feed-forward generator G_1 and G_2 GAN-BERT models had the highest QWK scores ($\kappa = 0.78, 0.75$ on the development set and $\kappa = 0.73, 0.73$ on the test set, respectively). The G_3 model demonstrated the most evidence of taking advantage of the competitive nature of the GAN structure: the generator confused the discriminator more often (one times out of ten) without compromising too much QWK performance ($\kappa = 0.73, 0.71$ on the development and test sets, respectively).

One of the largest limiting factors on the development of our models was the cost of training time. The success of the G_3 GAN-BERT in producing a competitive generator suggests to us that, with more training time, it would be useful to attach the DistilGPT2 module in the G_3 generator to the gradient (as opposed to fine-tuning it in advance and freezing its parameters during the GAN-BERT training).

With additional training time, we also suggest another GAN-BERT structure altogether G_4: the discriminator D itself is a BERT transformer; essays are tokenized and fed directly into the discriminator; and the generator G is a DistilGPT2 language model attached to the gradient. Such a structure would actually be simpler than the G_3 network (as it occludes the pre-preprocessing BERT transformer for real essays), but would require more time to train.

Finally, because of the unique expertise of each of our four types of models as mentioned in Section 5, we believe that an ensemble of the four models (RoBERTA, G_1, G_2, and G_3) could achieve exceptionally high QWK scores and would be worth investigation.

References

Blanchard, Daniel, Tetreault, Joel, Higgins, Derrick, Cahill, Aoife, and Chodorow, Martin. 2014. Ets corpus of non-native written english ldc2014t06.

Yue Cao, Hanqi Jin, Xiaojun Wan, and Zhiwei Yu. 2020. Domain-adaptive neural automated essay scoring. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 1011–1020, New York, NY, USA. Association for Computing Machinery.

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

Danilo Croce, Giuseppe Castellucci, and Roberto Basili. 2020. GAN-BERT: Generative adversarial learning for robust text classification with a bunch of labeled examples. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2114–2119, Online. Association for Computational Linguistics.

Danilo Croce, Simone Filice, Giuseppe Castellucci, and Roberto Basili. 2017. Deep learning in semantic kernel spaces. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 345–354, Vancouver, Canada. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Fei Dong, Yue Zhang, and Jie Yang. 2017. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 153–162, Vancouver, Canada. Association for Computational Linguistics.

Fortune Business Insights. 2022. Smartphone market size, share: Growth analysis report [2029].

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2020. Generative adversarial networks. *Commun. ACM*, 63(11):139–144.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized BERT pretraining approach. *CoRR*, abs/1907.11692.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peter Phandi, Kian Ming A. Chai, and Hwee Tou Ng. 2015. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, Lisbon, Portugal. Association for Computational Linguistics.

Research and Markets. 2022. Global english language learning market report to 2029 - featuring babbel, linguistica 360, mondly and elsa among others.

Pedro Uria Rodriguez, Amir Jafari, and Christopher M. Ormerod. 2019. Language models and automated essay scoring.

Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, Xi Chen, and Xi Chen. 2016. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS EMC*$^2 Workshop$.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas. Association for Computational Linguistics.

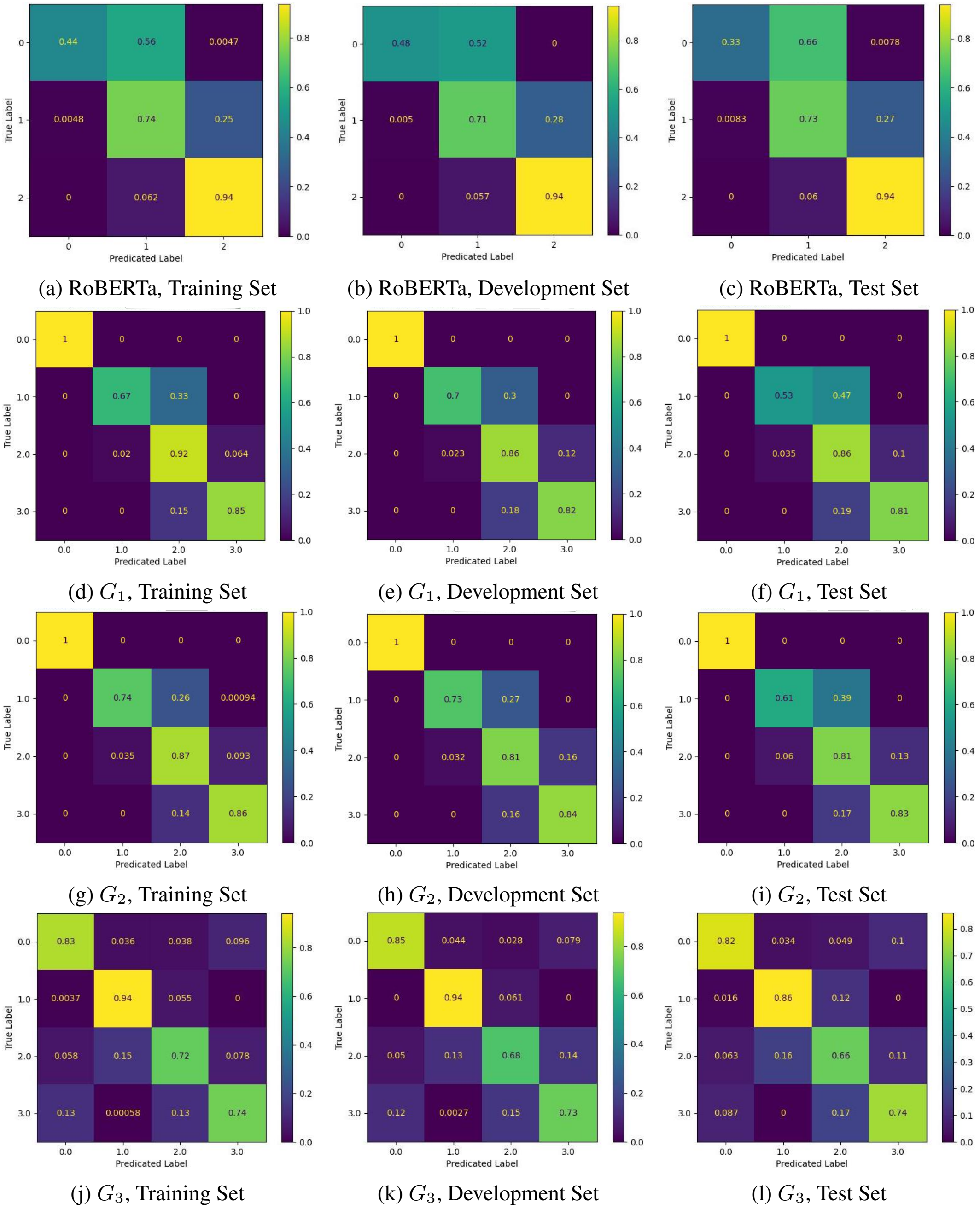
Masaki Uto, Yikuan Xie, and Maomi Ueno. 2020. Neural automated essay scoring incorporating handcrafted features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Yongjie Wang, Chuang Wang, Ruobing Li, and Hui Lin. 2022. On the use of bert for automated essay scoring: Joint learning of multi-scale essay representation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3416–3425, Seattle, United States. Association for Computational Linguistics.

Ruosong Yang, Jiannong Cao, Zhiyuan Wen, Youzheng Wu, and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569, Online. Association for Computational Linguistics.

7 Appendix

7.1 Additional Figures



(a) RoBERTa, Training Set (b) RoBERTa, Development Set (c) RoBERTa, Test Set

(d) G_1, Training Set (e) G_1, Development Set (f) G_1, Test Set

(g) G_2, Training Set (h) G_2, Development Set (i) G_2, Test Set

(j) G_3, Training Set (k) G_3, Development Set (l) G_3, Test Set

Figure 3: Confusion Matrices for RoBERTa Baseline and GAN-BERT Model Performance on the Training, Development, and Test Sets, normalized over the true counts (i.e., by row)

7.2 Fake Essays generated by Fine-tuned DistilGPT2 for G_3, but classified as Real by D

Prompt: Do you agree or disagree with the following statement? Young people nowadays do not give enough time to helping their communities.

7.2.1 DistilGPT2-Generated Essay Example #1: Scored as “High"

Discriminator D’s Classification Probabilities:

$$p(y = [\text{Fake}, \text{Low}, \text{Medium}, \text{High}]) = [0.3574 \quad 0.0007 \quad 0.0086 \quad 0.6333]$$

Essay Text:

The statement that young peoples nowadays do not give enough time to helping their communities is more of a general concequence than a reflection of the reality of the world. Some scholars contend that it is a fact that young people nowadays don’t give enough time to helping their communities, while others assert that the young people’s attitude towards helping their communities is only a manifestation of their true belief.

It is certainly true that young people, especially young people, are busy with their job and other expenses related their working life like the rest. However, today’s working conditions are still more severe, causing most of the young employees to go for holidays, or to spend the weekend doing something for the holiday.

These days, the most conspicuous example is the recent global war in which all countries joined to a "social justice" campaign. Since the United Nations launched a massive civil war in South East Asia, almost all people in the region could not afford their own cars. The consequences led to the mass protests in all countries of the world, in particular the US.

A lot of people still think that young people nowadays are just spoiled, even children. However, the fact is that nowadays they do care more about their future. They prefer to work hard and go out everyday. They seem to be working in the offices instead of just doing their job. Of course, there is a need to spend time for themselves, but there is also a need to focus their efforts with their relatives.

Moreover, people are less intrested in their future. People live in situations like conflicts and difficulties like poverty, which does not allow young people to contribute themselves in their life. They are also not able to organize their own work, as they do not have enough money. Additionally, most of young people enjoy studying outside the university and it helps them earn money in the future.

As far as I am concerned, the reality seems to be to the contrary that young people do devote enough time to help their communities. I have no doubt that nowadays young people are more careless than the previous. They are not doing their work for fun, but because of their selfishness, their attitude towards the work, the lack of experience and the lack of willingness to have responsibility. For example, we can think that an art college student spends her vacation with her friends in a small town, because she is too busy with her schoolwork. They are not able to dedicate their time to helping the communities.

Furthermore, if I lived in the U.S. I do not know how much time I want to spend for my community, it will not become an issue. I think that young people nowadays can choose to spend the most of their time instead of spending free time for other interests.

7.2.2 DistilGPT2-Generated Essay Example #2: Scored as “Medium"

Discriminator D’s Classification Probabilities:

$$p(y = [\text{Fake}, \text{Low}, \text{Medium}, \text{High}]) = [0.4171 \quad 0.1063 \quad 0.4634 \quad 0.0133]$$

Essay Text:

Yes, and I think there are various advantages of young people nowadays helping their communities. There are two advantages for me: For example, they tend the most important things during the college years, and generally help other people to have a home for themselves. They would be able to do the same job at the same time. In addition, the younger people would be able to get more responsabilities for the parents, and more, the older people would be able to think responsable to the children. It is hard to say why the younger people is helping their communities because the education system is a complex one. They also could not be aware of certain benefits such as their educatnce and the society.

For example, a high school student is able to study law, but also to have friends. He would not be able to go to a public university but if he goes to a college, and if he needs to pay his tuition, the college will take that decision.

As I mentioned before, we are very important people, and are not aware of the ways they can help their communities. If they want to give a part on the society, young people would have it more and more, and they would not get any more time.

7.2.3 DistilGPT2-Generated Essay Example #3: Scored as “Low"

Discriminator D’s Classification Probabilities:

$$p(y = [\text{Fake}, \text{Low}, \text{Medium}, \text{High}]) = [0.2521 \quad 0.4527 \quad 0.2905 \quad 0.0047]$$

Essay Text:

Well,I do agree with young people nowadays do not give enough time to helping their communities. So young people nowadays don’t have any time to spend their time to learn and learn a lot, but we don’t want to.So it is very different and I don’t know if it could happen for our live then it can happen in our world.There are many reasons that young people don’t want to do this. So I do agree some ways.

First,in my study I came to college for my studies and I found alot of bad things in the world. But I study hard and I had problem for my study some exam year that night that I felt very bad. And in my exam I felt very bad and because I was going to test the exam in the lab which is the same exam. So I said to me that I can’t teach about studying by heart, But I study many people and I tried to work and I feel bad.

Second, I think we are very busy in world because we are not a good people. So the old people don’t want to do my study or I think in the future they have some important situations we have not. So what it could happen in our life is that we live at a good or bad place if we do nothing. So in my opinion it is really better