Automating English Language Proficiency Assessments

Stanford CS224N Custom Project

Ethan Allavarpu
Department of Statistics
Stanford University
eallavar@stanford.edu

Duncan Ross
Department of Statistics
Stanford University
dunross@stanford.edu

Spencer Siegel
Department of Statistics
Stanford University
siegels@stanford.edu

Abstract

We have seen the proliferation in English proficiency assessments for non-native speakers–especially in writing and speaking–for academic and professional opportunities. While human graders have been the gold standard for evaluation, automated tools provide a scalable, objective alternative. We propose transformer models for written and spoken proficiency (based on DistilBERT and Wav2Vec2, respectively) finetuned on datasets from ETS, ICNALE, ELL, and Speechocean. We evaluated various models for both essay and speech assessments and found that simple models built on top of the transformers perform best. Our best models achieve 0.4254 and 0.5718 test R^2 for the written and spoken assessment total scores, respectively. Our predicted written and spoken assessment scores have correlations of 0.66 and 0.77 with the corresponding human-graded scores. The results of our study suggest that transformer models have the potential to be a reliable and efficient alternative to human graders for English proficiency assessments.

1 Key Information to include

- Mentor: Abhinav Garg
- External Collaborators (if you have any): N/A
- Sharing project: N/A

2 Introduction

Our primary goal is to create a neural network to score written and spoken proficiency exams (e.g., TOEFL, FCE) taken by ESL (English as a Second Language) learners. Proficiency exams are widely used by universities in the United States as an admission criterion for international students. By creating a more robust, accurate autograding system that can outperform simpler models, we have the potential to (1) improve efficiency in grading these exams (autograders score essays more quickly than their human counterparts) and (2) create an analytical approach to scoring these exams. This work could serve as a major asset and cost-saver for testing companies if the autograder can mirror the scores of human graders well. Additionally, this tool could be a valuable diagnostic for students preparing to take these proficiency exams. For evaluating essay responses, we built a model on top of the DistilBERT transformer. We then trained single and multi-output networks to predict overall score as well as categorical sub-scores to understand the grading criteria. Our approach to modeling spoken English profiency exams consisted of building on top of Facebook's Wav2Vec2 transformer. We trained a multi-output network to predict a variety of sentence-level sub-scores as well as an overall score. We also utilized human-graded word and phoneme-level sub-scores as additional targets in a more complex model, but did not find significant improvements with the more granular targets. We hope to show that neural networks can learn human-graded proficiency scoring and predict similar scores to humans on unseen test examples.

3 Related Work

Prior research in assessing proficiency exams has centered around (1) simpler models for evaluation and (2) error detection. In particular, Yannakoudakis et al. (2011) focused on using support vector machines (SVMs) to predict scores for First Certificate in English (FCE) exams. The model focuses on the FCE essays, so it does not guarantee generalizability across tests. In contrast, we aim to create a model that works for different examinations–such as the FCE exam as well as the TOEFL and other proficiency tests–rather than confining ourselves to a single exam. We believe that a deep-learning implementation for scoring could prove more useful for such generalizability than simple SVMs.

In a related paper, Rei and Yannakoudakis (2016) utilized the same FCE data but focused instead on implementing neural networks for error detection. Prior error detection research focused on context pattern matching or learning weights for context n-grams, but such methods struggled due to data sparsity and limitations from fixed context sizes. Rei and Yannakoudakis (2016) extended the prior research in error detection by generalizing to create an all-encompassing, deep-learning error detection model. They incrementally incorporated training data from different data sources to improve their model's performance, demonstrating the effectiveness of adding more training data (from different sources). This incremental incorporation of different data sources inspired us to use different datasets in a hierarchical or iterative model to learn more about the structure of language, though we plan to do so for scoring systems, not error detection.

Rei and Yannakoudakis (2016) attempted to perform error detection in English essays by non-native speakers, which correlates with our goal of creating an effective autograder for English proficiency tests. They suggest methodologies like the bidirectional LSTM and iterative training on increasingly more data, providing us with an example to consider when approaching written exam evaluation. That said, they did not provide suggestions about generating scores: they only focused on error detection. Ultimately, our work aims to combine the results from Yannakoudakis et al. (2011) and Rei and Yannakoudakis (2016) to create an autograder system that mimics human graders for overall scores (rather than error detection), but with a deep-learning focus (rather than SVMs). We do not aim to simply combine the results of these two papers, but plan to expand on them by considering different modalities: we will apply our approach for proficiency assessments to both text and audio mediums.

Additionally, machine learning scientists have used convolutions in deep learning and natural language processing for tasks like summarization or information extraction (Lopez and Kalita, 2017). Proficiency scoring is similar to these tasks: we aim to consider the document (essay or audio) in total to assess overall quality, so surrounding context may prove important–similar to how context is important when extracting information. As such, we experimented with using convolutions after the transformer to see if nearby context would improve overall quality of predicting proficiency scores.

One paper related to speech assessment models involved a model-based approach to spontaneous (not designed) spoken English (Wang et al., 2018). Wang et al. (2018) argued that such speech data from non-native speakers provides signal into mastery of language. In their models, they extracted features from transcribed text rather than modeling the audio data directly with a transformer. While Wang et al. (2018) obtained satisfactory results, they mentioned the difficulties with transcribing speech; we take a different approach by modeling with a speech model transformer to predict a variety human-graded sub scores. We show that we can capture signal without feature engineering from transcribed text, which highlights the power of Facebook's Wav2Vec2 transformer.

4 Approach

When modeling essay scores, we built on the DistilBERT transformer because it is a fast, cheap, and light transformer from Hugging Face that utilizes the BERT architecture (Sanh et al., 2019). We then added a single dropout and a linear layer to build our baseline model. Our baseline was trained exclusively on one dataset and evaluated on our holdout test set from a separate exam. We trained all other basic models on a single dataset and incorporated additional dropout or nonlinearities (ReLU or 1-dimensional convolution), alternating between linear layers, dropouts, and nonlinearities. Nonlinearities could improve expressivity and performance and, in particular, convolutions could take into account nearby context when assessing the essay's quality (Lopez and Kalita, 2017). Additionally, incorporating dropout before each linear layer further decorrelates the weight matrices.

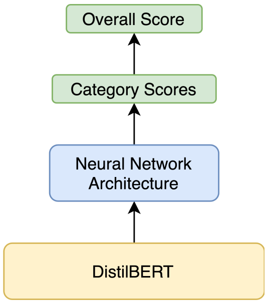


Figure 1: Hierarchical Model

We aimed to leverage multiple datasets, so we considered architectures that used multiple sources. Our iterative models were trained sequentially on different data: we finetuned on one dataset and loaded these adjusted parameters to subsequently finetune on a separate dataset. Doing this allowed us to train on different sources as suggested by Rei and Yannakoudakis (2016). Our hierarchical model trains simultaneously on two different datasets. One dataset has 6 category scores while another has an overall score: the hierarchical model connects a length-6 vector to a single output, linking the two outputs (Figure 1). We simultaneously train on the loss for the length-6 vector and the final output, interleaving data from both sources into the same batch to reduce variance in the loss and gradient between batches. If the hierarchical model performed well, we then could establish a causal link between the 6 categories and the overall score.

For the multi-task model, we still trained on two different datasets simultaneously, but rather than having the length-6 vector pass through to a single output, we had our final output as a length-7 vector. The first 6 elements correspond to the ELL categories and the last element corresponds to the overall score for the ICNALE data (Figure 2). For prediction, we take the last element as our prediction of the overall essay score. We considered this as an alternative to the hierarchical in case the 6-1 hierarchical structure provided too strong of a constraint on the model that hindered performance. A multi-task response allowed us to simultaneously optimize for granular and high-level overall assessment without that relational constraint.

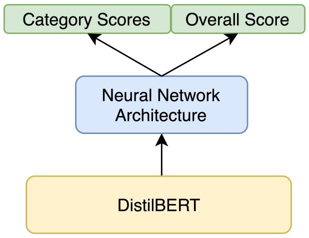


Figure 2: Multi-task Model

Our final model type was an ensemble model, which simply combines the predictions of multiple models and computes a weighted average as the final prediction. For all implementations (written and speech), we added neural network architecture with PyTorch to the existing transformer for finetuning. We also note the transformers used (i.e., DistilBERT) are not novel: we utilize Hugging Face (Sanh et al., 2019). Additionally, we incorporated boilerplate code from CS224N Assignment 5 for training.

Our first speech modeling approach is a sentence-level multi-output model to predict human-graded speech scores for English language learners. We built this model on top of Wav2Vec2 with audio files from English learners and corresponding scores (Baevski et al., 2020). Wav2Vec2 processes audio files into useful representations for downstream tasks by training on speech audio and finetuning on transcribed speech. Wav2Vec2 is trained using connectionist temporal classification (CTC) which is a useful technique for handling sequence tasks like speech in which the timing varies. Wav2Vec2 also can handle larger audio input when compared to similar models such as Whisper (Radford et al., 2022). We passed tokenized audio input to Wav2Vec2 and then utilized the last hidden state to pass to the rest of the network. We added a dropout layer ($p = 0.3$) followed by a linear layer to predict human-graded sentence scores. This final linear layer outputs 4 sentence-level scores: accuracy, fluency, prosodic, and total score. This multi-output technique is powerful since these sentence-level subscores are related and can help the model learn more specific traits of English proficiency. We simply used the average mean-squared error across the 4 scores as our loss function for this network.

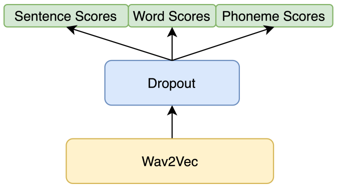


Figure 3: Granular-Output Speech Model

The second model that we implemented for speech involved the utilization of more granular human-graded scores. Along with sentence-level scores, our dataset contained word-level scores (accuracy, stress, total score) and a phoneme-level score (accuracy). We tried many techniques to incorporate these granular sub-scores and arrived at adding extra regression heads (Figure 3). The sentences in our dataset were all very short in length: each contained at most 10 words and 30 phonemes. Hence, we output 3 sub-scores for each of the potential 10 words in the sentence (total output of length 30 for word-level output). Similarly, we output a vector of length 30 which correspond to phoneme accuracy for the possible 30 phonemes in the sequence. As such, this model in total had 64 outputs: 4 on sentence-level, 30 on word-level, 30 on phoneme-level. For the loss function, we calculated the average mean-squared error for each of the 3 output levels separately (sentence, word, phoneme) then summed these losses for our loss criterion. We attempted to utilize weights to allow for a more optimal weighting of losses over the 3 levels, but did not find any significant improvements. When a sentence contained fewer than 10 words or 30 phonemes, we masked the loss for these outputs.

5 Experiments

5.1 Data

Our test essay data from Cambridge includes 1,223 hand-graded exams, on a 0-40 scale, from the First Certificate in English (FCE) (Yannakoudakis et al., 2011). For the training datasets, we have data from ICNALE, with 5,600 writing samples by college students in ten different Asian countries, with scores on a 0-100 scale for approximately 600 essays. We also have data from ELL that provides a more granular score of 6 constituent categories, each on a scale of 1-6 with half point increments, for 3,911 essays. Our third training dataset from ETS provides 12,100 scored essays (Blanchard et al., 2014). We primarily used this dataset in initial stages of training/finetuning, as the scores were categorical: "low," "medium," or "high." This data allowed us to help the model gain a general idea of a "bad," "okay," or "good" essay before proceeding to continue training with ELL, ICNALE, or both to get specific scores for these essays. We scaled the numeric essay scores to 0-100 to make the deviations comparable across datasets.

For speech, we used a dataset from Speechocean with 5000 English sentences by 250 different non-native speakers who speak Mandarin as a first language (Junbo Zhang, 2021). The sentences were all very short as previously mentioned and we processed the WAV audio files through Hugging Face's audio dataset framework. The Speechocean data includes human-graded sentence-level scores (averaged from 5 experts) for accuracy, fluency, prosodic, and total (all graded on a scale of 10). The dataset also included human-graded scores for each phoneme (out of 2) as well as word-level scores for accuracy, stress, and total (out of 10). The sentence and word-level scores were all integers, whereas the phoneme scores had one decimal place. We scaled all targets to be out of 100 for consistency and to align with the written targets. Unfortunately, this was the only speech dataset with human scores that we acquired, so we trained on this dataset and evaluated on a hold-out test set of 500 samples (10%) from subjects that did not appear in the training or validation sets.

5.2 Evaluation method

The evaluation metrics we used to assess performance were RMSE (root mean squared error) and test $R^2 = 1 - (SSE/SST) = 1 - (N_{\text{test}}(RMSE)^2)/(\sum_{i=1}^{N_{\text{test}}}(y_i - \bar{y})^2)$ to compare predictions to true scores. We created predictions for the test set and then calculate our metrics with respect to the predicted and observed values (i.e., the scores by the humans who graded the essay). We used test R^2 because it has a maximum of 1, providing an easy metric to understand model performance (higher values are better). For each dataset, we made sure that the ranges of scores were adapted to be the same (i.e., out of 100 with the same mean). For the essay models, we mean-adjusted because different exams might have different expected means, but the results should be centered around a known average. In practice, exam graders know what the overall average performance for a given exam would be. For the ELL dataset, because the output is 6 category scores, we averaged these to get a single essay score (to align with the FCE test set and our end goal).

5.3 Experimental details

We built on top of the DistilBERT transformer for all of our essay models. When finetuning, we made an 80-20 train-validation split and used the AdamW optimizer. Our initial models used a learning rate of 2×10^{-5}, and weight decay of 0.1 for all parameters other than bias and layer norm parameters. We made our learning rate small because we did not want to overfit to the small-sized training data. We also attempted to combat overfitting with weight decay and dropout. However, we did not decay the learning rate over the 20 epochs used in the finetuning process.

For subsequent essay models, we utilized the Ray Tune framework for hyperparameter tuning (Liaw et al., 2018). Specifically, we experimented with various combinations of hyperparameters such as learning rate, learning rate decay, weight decay, batch size, and number of epochs to achieve the best results. We used a grid search with predefined ranges for each hyperparameter and use the validation set to evaluate the model's performance. Through the use of Raytune, we found that a learning rate of 6×10^{-6} and no learning decay was optimal, as well as smaller batch sizes of 8 or 16. We saved the model which achieved the lowest validation loss during training. For these essay models, we utilized the AdamW optimizer with mean-squared error as the loss criterion in all cases except when training on the ETS dataset; since the ETS dataset had a categorical output, we used cross-entropy as the loss function for that dataset.

The speech model experimentation framework was similar to that of the essays. Since we only had the Speechocean dataset, we used a 80-10-10 train-validation-test split for the 5000 sentences in this dataset. We split on subject-level to ensure that all sentences for a subject appeared in the same split to prevent leakage. Additionally, we performed the experiment 5 separate times with varying seeds to obtain significant, trust-worthy results. We then averaged across the 5 experiments for evaluation purposes. As with the essay models, we once again utilized Ray Tune with a grid search for hyperparameter tuning. The speech models performed optimally with a learning rate of 2×10^{-5}, no custom learning decay, and a batch size of 16. We trained the sentence-level model for 10 epochs and the granular-output model for 15 epochs, saving the model which had the lowest sentence-level validation loss. For both speech models, we utilized the AdamW optimizer (weight decay of 0.1 on all parameters besides bias and layer norm) and mean-squared error for the loss.

5.4 Results

Model	Additional Dropout?	Nonlinearity	Data Used	RMSE	Test R^2
Baseline	No	None	ICNALE	11.1732	0.3255
Basic	No	None	ELL	10.8779	0.3607
	Yes	ReLU	ICNALE	10.3576	0.4204
	Yes	ReLU	ELL	10.9572	0.3513
	Yes	Convolution	ICNALE	12.0160	0.2199
Iterative	No	None	ELL, ICNALE	10.7399	0.3768
	Yes	ReLU	ETS, ELL	10.8381	0.3654
Hierarchical	Yes	None	ELL, ICNALE	13.1572	0.0647
Multi-task	Yes	None	ELL, ICNALE	10.6521	0.3870
Ensemble				**10.3125**	**0.4254**

Table 1: Essay Models

Our essay model performance (Table 1) varied a bit by model. We expected the models to struggle generalizing because we evaluated on FCE, which was a hold-out test set from a different exam (so we did not know if the distribution of scores and the human-grading criteria would align). We see this phenomenon, as test R^2 scores for our best model (the ensemble model[1]) was 0.4254. We had a correlation of 0.66 between our predictions and the true scores, which is a strong showing for generalizability. Additionally, we note that the ensemble model performing best aligned with our initial expectations–combining the results from multiple models produced a slightly better model (not drastically better, but better nonetheless).

The two models that performed worse than expected were the model with a convolution layer and the hierarchical model. While we expected the convolution to work well (and take in surrounding context in assessing essays), its test R^2 was a mere 0.2199. One reason for this might be that the transformer already accounts for context, and by potentially double-counting we may have focused on noisy aspects of the grading process. Additionally, this demonstrated to us that simpler models may be better: a ReLU layer seemed to work better than a convolution and the best model (outside the ensemble), was the basic model trained on a single dataset. Moreover, we see that incorporating a hierarchical model resulted in extremely poor performance: the test R^2 was 0.0647, which was marginally better than predicting the mean for every essay. Perhaps the strong constraint of a direct link between the 6 category scores and the one overall score caused the model to struggle to learn from either loss function–so it ended up staying around the mean. We contrast this to the multi-task model, which jointly predicted both category and overall scores at the same level–that approach seemed to perform similarly to our other models. Overall, this demonstrated that a causal link, though nice in theory, was not necessarily optimal in this case and that placing too many constraints (like the 6-to-1 linear layer) in the model could lead to poor performance.

While the essay models performed about as well as we expected, the speech models overall performed very well and even better than anticipated. As expected and shown in Table 2, both speech models performed significantly better on the training set than the test set. The results in the table include R^2 and RMSE values for each sentence-level output, averaged across the 5 random seeds to generate train-val-test datasets. Obviously it is difficult to generalize to unseen examples, but the training samples also included 20 sentences from 200 subjects. Since we have multiple sentences from the same subject, this may have caused the model to pick up subject-specific tendencies that contribute to certain scores. However, we also note that the granular-output model had better training RMSE and training R^2 than the sentence-level model. The granular-output model was able to train for more epochs than the sentence-level model without severe overfitting to the training set–we see the performance for both models is similar on the test set.

[1] The produced ensemble model combined the predictions from (1) the basic model trained on ICNALE with additional dropout and ReLU and (2) the iterative model trained on ELL and ICNALE

	Training			
	Accuracy	Fluency	Prosodic	Total Score
Sentence-Level Model	0.7842 (7.3588)	0.8181 (6.6130)	0.8147 (6.2748)	0.8286 (6.4178)
Granular-Output Model	0.8351 (6.5726)	0.8635 (5.5180)	0.8558 (5.5714)	0.8691 (5.7884)
	Testing			
	Accuracy	Fluency	Prosodic	Total Score
Sentence-Level Model	0.5371 (11.6225)	0.6380 (9.0692)	0.6369 (8.9554)	0.5718 (11.1194)
Granular-Output Model	0.5152 (11.8528)	0.6574 (8.7936)	0.6511 (8.7406)	0.5610 (11.2090)

Table 2: Average speech model test R^2 (RMSE in parentheses) across 5 random train-val-test splits

From Table 2, we see that the speech models generalized well to unseen test sentences from different non-native speakers. The sentence-level model achieved a test R^2 score of 0.5718, while the granular-output model had a test R^2 of 0.5610. Additionally, the correlation between the test set predictions for total score and the human-graded total score was 0.772 for the sentence-level model and 0.764 for the granular-output model. This was an encouraging result: our models could learn from human-scores of English proficiency and grade similarly on new data. Another interesting result was that both models perform better on the fluency and prosodic sub-scores than on accuracy and total score. This pattern was not apparent on the training set but was pretty clear on the test set; both models achieved test R^2 above 0.63 for fluency and prosodic sub-scores and 0.5-0.6 for the others. One potential explanation for this is the variance of the scores themselves: the variance for fluency and prosodic scores are 2.16 and 2.23, respectively, while the variance for accuracy and total scores are 2.59 and 2.62, respectively. This higher variability in the scores could lead to worse performance on a test set.

There appeared to be no significant difference on test-set performance between the two speech models. Overall, the sentence-level speech model performed very well despite its simplicity. The granular-output model slightly outperformed the sentence-level model in terms of test R^2 on fluency and prosodic sub-scores. Overall, the results were pretty similar despite the additional word and phoneme-level outputs in the granular-output model. This can likely be attributed to the fact that the model could not learn how these granular outputs were derived. We passed these additional outputs to the model, but the model does not know the exact occurrence of each phoneme or word in the sequence and its corresponding score. The Wav2Vec2 model is trained using connectionist temporal classification (CTC) but we have not implemented an automatic speech recognition (ASR) model on top of this to learn the phonemes and words in the sequence in which the granular outputs reference. While these granular outputs could have offered some added benefit to the model by simply having extra responses, we believe that we could add more complexity to the model to take complete advantage of these outputs.

6 Analysis

One takeaway from constructing our essay models was that the best models were typically trained with a smaller number of epochs. In particular, we saw that while the training loss continued to decrease as the number of epochs increased, the validation loss stagnated after around 5 epochs (Figure 4). As such, this bolstered our understanding that "simpler is better." If we trained for more epochs, we would end up overfitting and thus, worsening the generalizability of our model. Hence, we trained for a limited number of epochs in our best models.

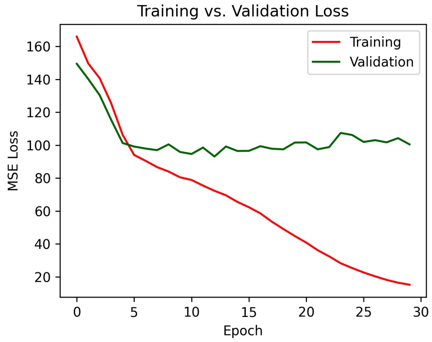
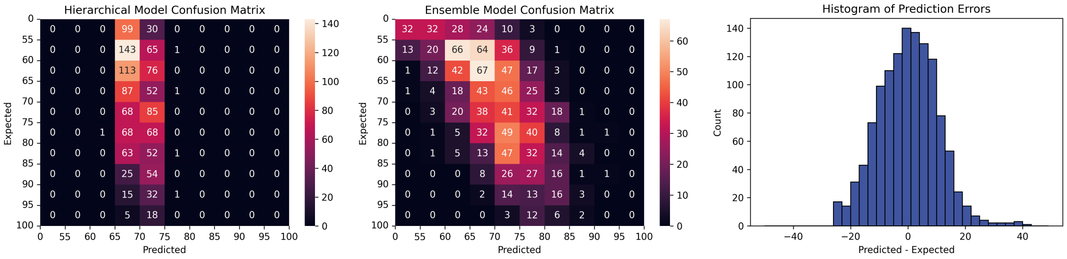


Figure 4: Train vs. Validation Loss

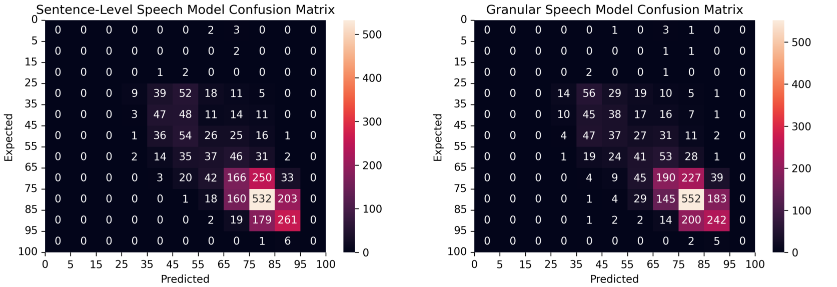
To understand the strengths and weakness of specific models, we constructed confusion matrices. We bucketed the essay scores in increments of 5 starting at 55. Figure 5a shows the hierarchical model's predictions vs expected results. The hierarchical model tried to optimize loss on the two datasets, but failed to do so. It placed a strict constraint by modeling a causal link between the 6 category scores and the overall score: we believe that this constraint limited the effectiveness in learning both components, so the model compromised by predicting near the mean score for all essays. We noticed that, when we removed this sequential constraint (as we saw for the multi-task model), performance was comparable to other model types. That said, one interesting aspect of our ensemble model (which was true of many essay models we tried) was that we rarely predicted scores above 85 even though we see a decent number of essays with true scores above 85. This phenomenon is something we aim to correct in future model iterations to more accurately predict on the higher end.



(a) Confusion Matrix (Hierarchical) (b) Confusion Matrix (Ensemble) (c) Prediction Error (Ensemble)

Figure 5: Analysis of Essay Models

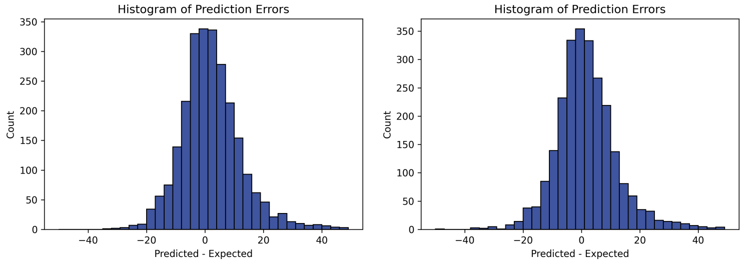
While the hierarchical model failed, we saw that the ensemble model performed strongly: Figure 5b shows its predictions vs. the expected results, and the ensemble model performs much better than the hierarchical as the relative trend fits closer to the diagonal. We observe plenty of instances around the diagonal, indicating we often predicted ± 5 points of the true score. The noise about the diagonal (rather than values exactly on the diagonal) could be attributed to the limited data available for finetuning. We suspect that if we increased the size of training corpus (currently, we only have around 4,500 essays), we would observe more points on the diagonal and more accurate predictions because of the data increase. We further see this through a histogram of the prediction errors as found in Figure 5c. With a bin size of 3, we notice that our predictions errors are approximately normally distributed around 0. If we increased training size n, the distribution would smooth and tighten around the mean as desired, resulting in stronger model performance.



(a) Confusion Matrix (Sentence-Level) (b) Confusion Matrix (Granular-Output)

Figure 6: Speech Model Confusion Matrices

Unlike the essay models, we evaluated the speech models on the same dataset, so we did not have to consider generalization to different human-grading scoring criteria. Furthermore, our audio files were recordings of examiners speaking short sentences with an average length of 6.36 words, so we could not test the model on longer audio input. Figure 6a shows the confusion matrix for our sentence-level model for the total sentence-level score. Since all actual scores are a multiple of 10, we created buckets that indicate whether our prediction is within 5 points of the actual score. The confusion matrix seems to be diagonal for the sentence-level model and this trend is nearly identical for the granular-output speech model in Figure 6b. The sentence-level model had 44% of the predictions in the correct bucket (within 5 points) and 87% of predictions within one bucket of the actual (within 15 points). The granular-output output had 45% in correct bucket and 87% within one bucket. We saw this as a positive indicator, but we examined the confusion matrix more closely to see that most of the actual scores fall within 60-90 (84.6% of actual total scores). The models struggled to predict well on extremely low scores as the predictions never fell below 25 despite a handful of human-graded scores below this number. We can also see this with the histograms in Figures 7a and 7b. The errors appear to be centered at 0, but there is a right tail which shows a bias towards higher predictions. Additionally, the model never predicted greater than 95 despite having some actual perfect test scores. It also indicates that the model learned the distribution of the human-graded scores from the Speechocean dataset.



(a) Sentence-Level Prediction Error (b) Granular-Output Prediction Error

Figure 7: Speech Model Prediction Errors

7 Conclusion

In our project, we found moderate performance in predicting essay scores. We achieved a test R^2 above 0.4, with the correlation between our predictions and the actual essay scores around 0.66, indicating moderate correlation. We claim that this is a strong performance because we assess the generalizability of our models to different English proficiency exams (i.e., we trained on ELL, ETS, or ICNALE data but evaluated on FCE). That said, we note that one limitation is that these scores are from different tests, which might have different minutia in their grading schema. Thus, we captured a general trend in the scores (our models accurately gave essays with better scores higher predictions). One reason we may not perform much better is that the scoring criteria may differ across exams. Another limitation is that essay scores are inherently noisy, as each human grader has inherent biases and the specific score depends on the specific grader. We note that the different graders might give the same essay for the same test different scores, showing that the grading process itself is noisy and prone to high variability. The correlations of 0.66 of our predictions with human-graded FCE scores show strong performance and generalizability in spite of these limitations.

We have shown significant results with our speech modeling for non-native English Speakers. We obtained test R^2 above 0.56 on total score on the Speechocean dataset. Both the sentence-level and granular-output model were able to predict within 5 points of the actual total score in over 40% of test samples and within 15 points on 87% of test samples. We demonstrated an ability to grade speech similar to the Speechocean human graders, but we are curious to test how this model would generalize. Additionally, we did not see a significant improvement with utilizing the word and phoneme-level outputs in the granular-output model. We believe that we could further improve our models to include an automatic speech recognition (ASR) model which could learn the phonemes and words in the sentence while simultaneously predicting the sub-scores which correspond to the words and phonemes. This could allow the model to learn the location in the sequence from which a given phoneme or word score is derived. While we have shown an ability to differentiate between various levels of spoken English proficiency, we believe that we can do more work to take advantage of the phoneme and word-specific scores available in the Speechocean dataset.

The next steps for this project are to merge our models into a single autograder system that encapsulates written and spoken proficiency together. We currently have strong separate autograders, but have not been able to combine the two for the creation of an autograder for a complete proficiency exam because of the data restrictions–we have not found a dataset that has both written and speech components publicly available with scores for both sections. Ideally, if we could find such a dataset, we would be able to combine our models to predict a single overall proficiency score akin to an overall TOEFL score. Currently, we have models that can produce useful results by scoring written and oral proficiency separately, but to combine these two for an overall score in the future, we will need a dataset that contains information on both modalities for the same individual.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in neural information processing systems*, 33:12449–12460.

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

Yongqing Wang Zhiyong Yan Qiong Song Yukai Huang Ke Li Daniel Povey Yujun Wang Junbo Zhang, Zhiwen Zhang. 2021. speechocean762: An open-source non-native english speech corpus for pronunciation assessment. In *Proc. Interspeech 2021*.

Richard Liaw, Eric Liang, Robert Nishihara, Philipp Moritz, Joseph E Gonzalez, and Ion Stoica. 2018. Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*.

Marc Moreno Lopez and Jugal Kalita. 2017. Deep learning applied to NLP. *CoRR*, abs/1703.03091.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust speech recognition via large-scale weak supervision.

Marek Rei and Helen Yannakoudakis. 2016. Compositional sequence labeling models for error detection in learner writing. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1181–1191, Berlin, Germany. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter.

Y. Wang, M.J.F. Gales, K.M. Knill, K. Kyriakopoulos, A. Malinin, R.C. van Dalen, and M. Rashid. 2018. Towards automatic assessment of spontaneous spoken english. *Speech Communication*, 104:47–56.

Helen Yannakoudakis, Ted Briscoe, and Ben Medlock. 2011. A new dataset and method for automatically grading ESOL texts. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 180–189, Portland, Oregon, USA. Association for Computational Linguistics.