DATS 6312

Natural Language Processing

Prof. Amir Jafari

Evaluating Language Knowledge of ELL Students

TEAM 8

Individual Report

by

Sanchit Vijay

Table of Contents

1. Introduction

This project uses advanced NLP models, including BERT, Electra, RoBERTa, and DeBERTa, to assess English proficiency in essays by 8th-12th grade English Language Learners (ELLs). Utilizing the ELLIPSE corpus, the goal was to predict linguistic proficiency across dimensions like cohesion, syntax, and grammar. Various pooling methods, such as LSTM and Mean pooling, were explored to aggregate contextual data effectively. This approach is crucial for providing accurate feedback to ELLs and aiding educators. The project showcases the potential of NLP in education, particularly in personalized language learning. Its unique strategy of multilabel regression, differential learning rates, and additional fine-tuning of saved models underscores its innovative approach to understanding language proficiency in ELL writings.

2. Description of Dataset

The ELLIPSE corpus, provided by Vanderbilt University, comprises argumentative essays written by 8th-12th grade English Language Learners (ELLs). These essays, which assess students' proficiency in English, are a key part of this project. Each essay is evaluated across six dimensions: cohesion, syntax, vocabulary, phraseology, grammar, and conventions, with scores ranging from 1.0 to 5.0 in half-point increments. This detailed scoring system offers a multi-faceted view of each student's English skills. The focus on argumentative essays provides rich insights into the students' ability to articulate thoughts and reason in English, making the ELLIPSE corpus a valuable resource for assessing and predicting language proficiency.

3. Description of the NLP Model and Algorithm

3.1 Transformer-Based Models for Language Proficiency Evaluation

Our project employed a range of advanced transformer-based models to analyze English proficiency in student essays. These models, known for their exceptional language processing capabilities, were selected for their unique strengths:

BERT-base-uncased: Known for deep bidirectional processing, this model handles text without case sensitivity, making it ideal for generalized text analysis and contextual understanding.

Electra Base Discriminator: Its innovative approach in differentiating real from artificial words allows for fine linguistic detail detection, crucial in assessing syntax and phraseology.

RoBERTa Large: As an optimized BERT variant with more parameters, RoBERTa provides a deeper contextual analysis, essential for evaluating complex sentence structures.

DeBERTa v3 Base and Large: These models enhance BERT's capabilities with a disentangled attention mechanism, allowing for a more nuanced understanding of word

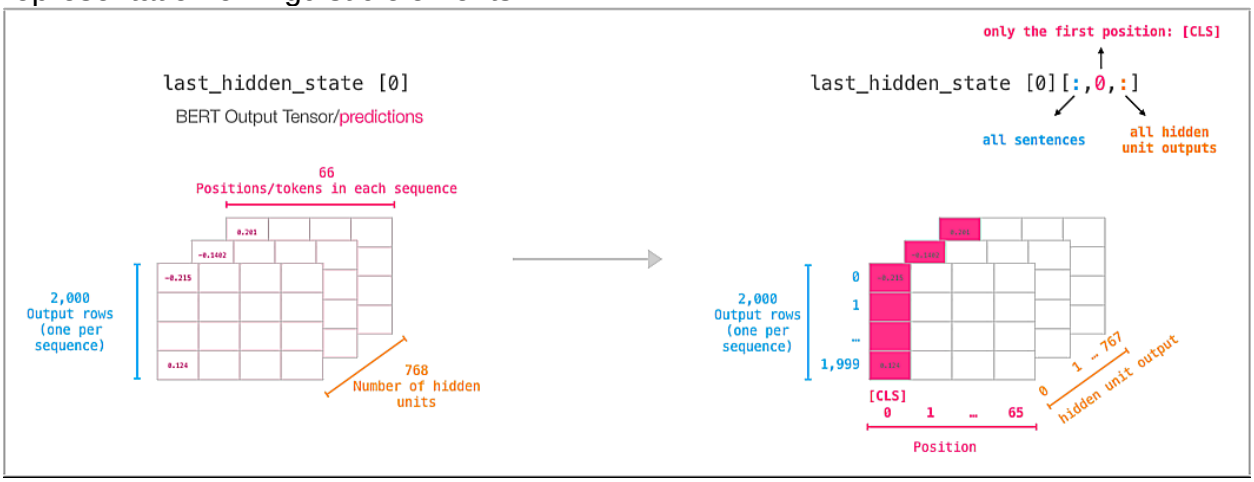
relationships. The large variant, in particular, excels in analyzing complex linguistic features.

Each model underwent fine-tuning on our dataset of English Language Learner (ELL) essays, adapting to the specific linguistic characteristics of language learners. This diversity in models facilitated an effective comparison of their abilities to assess language skills in an educational context.

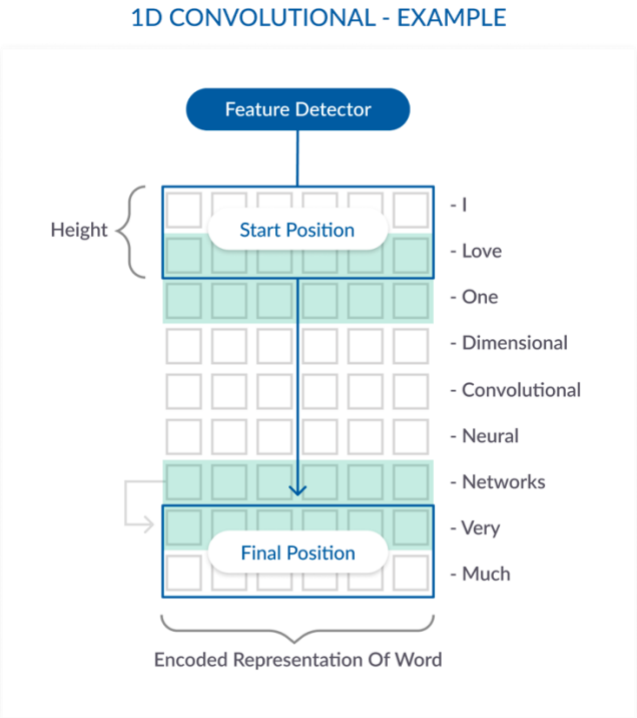
3.2 Data Preprocessing and Tokenization
Data preprocessing was critical in our project, involving tokenization to convert raw text into a format suitable for model processing. We employed tokenizers with specific configurations, including the addition of special tokens, setting maximum sequence lengths tailored to each model, and ensuring proper truncation of texts. The dataset was strategically divided for training (80%), validation (20%), and unseen testing (1%), ensuring comprehensive model training and real-world applicability.

3.3 Pooling Techniques in Language Processing
We explored various pooling methods to extract essential features from transformer model outputs:

Mean Pooling: Averages token embeddings, effectively capturing the overall semantic meaning.
LSTM Pooling: Processes embedding sequences, ideal for understanding syntax and cohesion.
Concat Pooling: Combines last hidden states from multiple layers, offering a rich representation of linguistic elements.



Conv1D Pooling: Applies a 1D convolutional network, focusing on local contextual features.



Each pooling method brought a unique perspective in processing transformer model outputs, enhancing our models' capabilities in assessing different aspects of language proficiency.

3.4 Training Setup and Optimization
Our training approach revolved around a regression setup, using Mean Squared Error (MSE) for training and validation. We employed Adam optimizer with differential learning rates for various model components, ensuring each part learned at an appropriate pace. The learning rate scheduler followed a cosine annealing schedule with a warm-up phase, promoting faster convergence initially and refined learning later.

3.5 Utilizing Weights and Biases for Experiment Tracking
Weights and Biases (WandB) played a vital role in our project for systematic experiment tracking and real-time logging. This tool significantly enhanced the efficiency of our project, providing a robust platform for tracking machine learning experiments and outcomes.

3.6 Hyperparameter Settings
The project's hyperparameters were meticulously chosen to balance learning efficiency and computational constraints. Key parameters included the seed for reproducibility, batch sizes optimized for different models, learning rates for various model components, and a scheduler to adjust learning rates effectively. In the fine-tuning phase, we decreased the learning rates and increased weight decay to prevent overfitting and enhance model reliability.

Our project stands out in its approach to multilabel regression, a challenging task in NLP. By combining various backbones with multiple pooling techniques, we achieved a

nuanced understanding of language proficiency. Differential learning rates and fine-tuning strategies further enhanced our models' performance, showcasing their potential in providing accurate language proficiency assessments in educational settings.

Backbones	Training batch size	Validation batch size	Maximum Lengths	Pooling	Trainable parameters
Bert-base-uncased	32	64	512	Mean	109,486,854
				LSTM	116,836,614
				Concat	109,500,678
				Conv 1D	109,778,054
Electra-base-discriminator	32	64	512	Mean	108,896,262
				LSTM	116,246,022
				Concat	108,910,086
				Conv 1D	109,187,462
Roberta-large	12	32	512	Mean	355,365,894
				LSTM	363,762,694
				Concat	355,384,326
				Conv 1D	355,753,862
Deberta-v3-base	12	32	768	Mean	183,760,134
				LSTM	191,109,894
				Concat	183,773,958
				Conv 1D	184,051,334
Deberta-v3-large	2	8	1024	Mean	433,916,934
				LSTM	442,313,734
				Concat	433,935,366
				Conv 1D	434,304,902

4. Results

4.1 Results of pre-training

Backbone \ Pooling	Mean	LSTM	Concat	Conv 1D
Bert-base-uncased	0.5116	0.5357	0.4818	0.5272
Electra-base-discriminator	0.5419	0.5024	0.5619	0.5031
Roberta-large	0.5050	0.4727	0.4616	0.4660
Deberta-v3-base	0.4684	0.4807	0.4712	0.4758
Deberta-v3-large	0.4545	0.4717	0.4751	0.4606

4.2 Results of fine-tuning

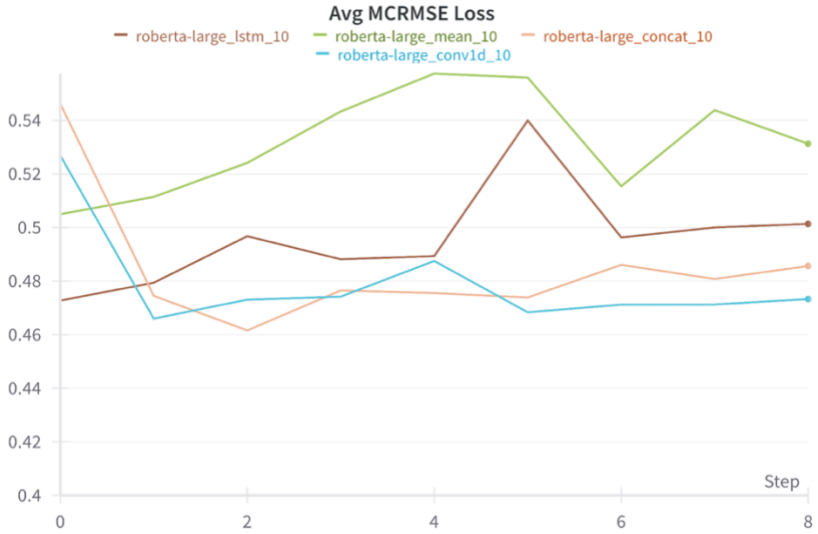
Backbone \ Pooling	Mean	LSTM	Concat	Conv 1D
Roberta-large	0.4815	0.4928	0.4130	0.4394
Deberta-v3-base	0.4492	0.5125	0.4644	0.4340
Deberta-v3-large	0.3986	0.4221	0.4358	0.4131

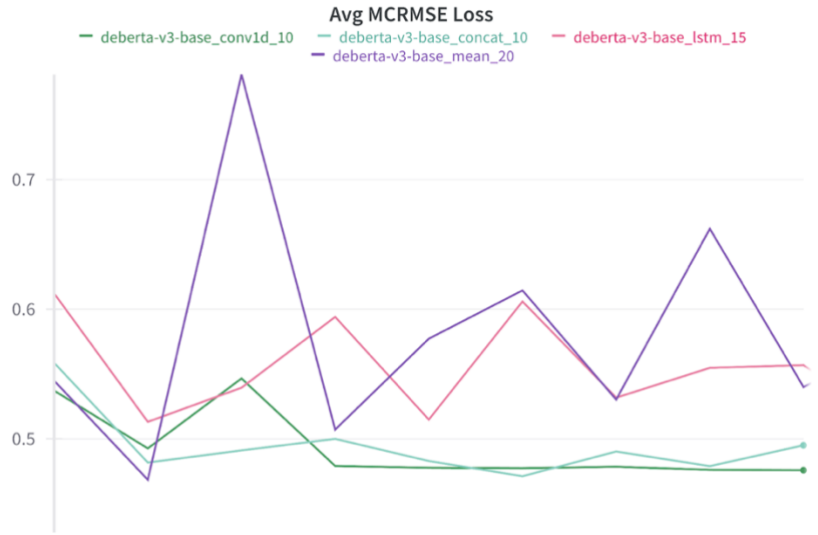
In our project, the performance of various backbone models combined with different pooling techniques was evaluated using the average MCRMSE loss metric, revealing key insights into their compatibility. Concat Pooling excelled with BERT-base-uncased, highlighting its ability to integrate information from different layers effectively. For the Electra-base-discriminator, LSTM and Conv1D Pooling proved more effective, aligning well with Electra's unique representations, while Mean and Concat Pooling lagged behind. RoBERTa-large favored Concat Pooling, leveraging its rich, layered output, with Conv1D also showing strong performance. DeBERTa-v3 models responded best to Mean and Conv1D Pooling, efficiently capturing their complex representations.

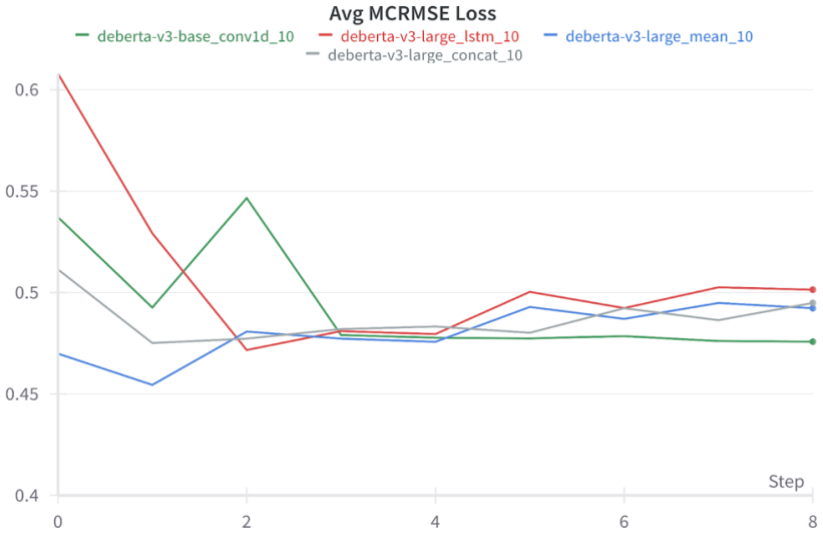
The fine-tuning phase, particularly for larger models like RoBERTa-large and DeBERTa-v3 variants, showed notable differences from the pretraining results. Concat Pooling significantly improved in RoBERTa-large, indicating effective synergy with fine-tuning, while Mean Pooling's performance dropped. DeBERTa-v3-base saw a balanced performance across pooling methods, with Mean Pooling leading slightly. In DeBERTa-v3-large, Mean Pooling demonstrated high effectiveness but suggested potential overfitting, while LSTM Pooling maintained a solid performance.

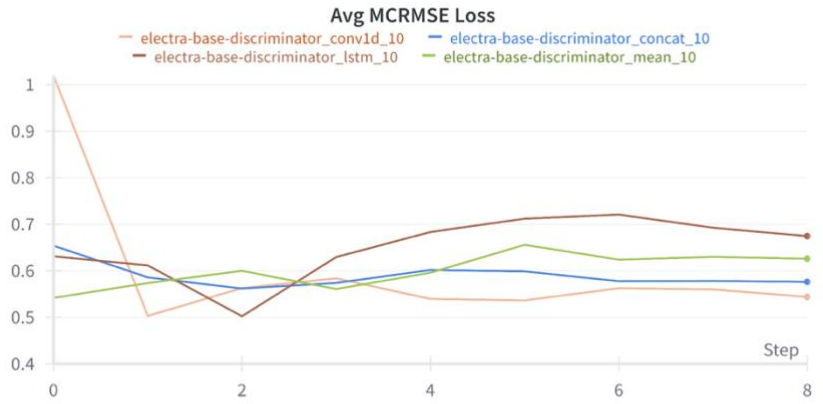
These findings highlight the importance of selecting appropriate pooling strategies in alignment with model architectures, particularly under fine-tuning conditions. The variance in results underscores the nuanced relationship between different model types and pooling techniques, emphasizing the need for tailored approaches in optimizing model performance for language proficiency assessment.

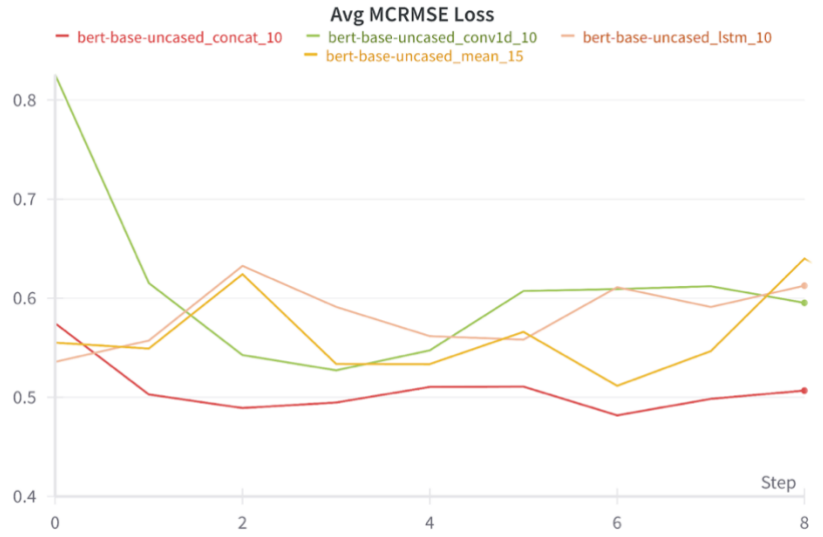
4.3 Comparing a backbone with different poolings



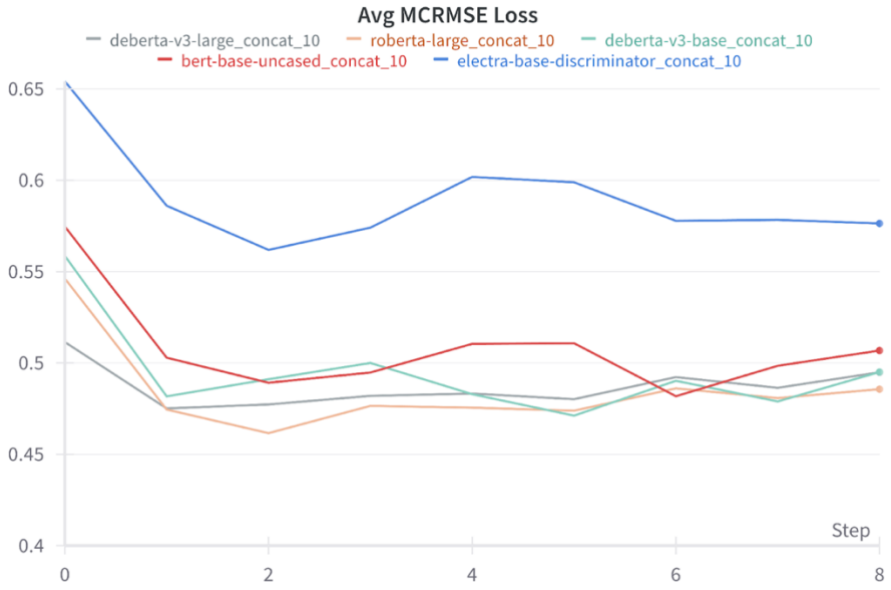


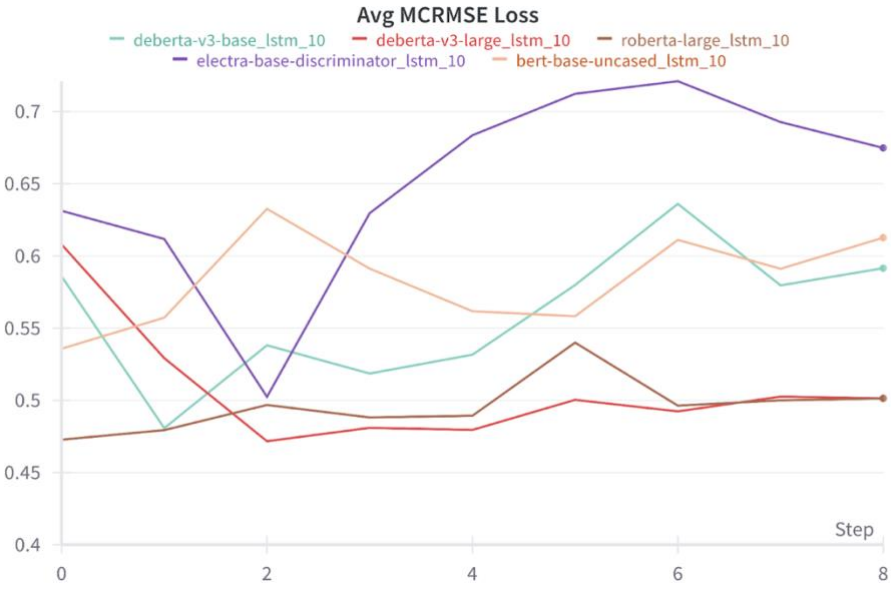

Avg MCRMSE Loss
deberta-v3-base_conv1d_10
deberta-v3-large_lstm_10
deberta-v3-large_mean_10
deberta-v3-large_concat_10
0.6
0.55
0.5
0.45
0.4
Step
0
2
4
6
8

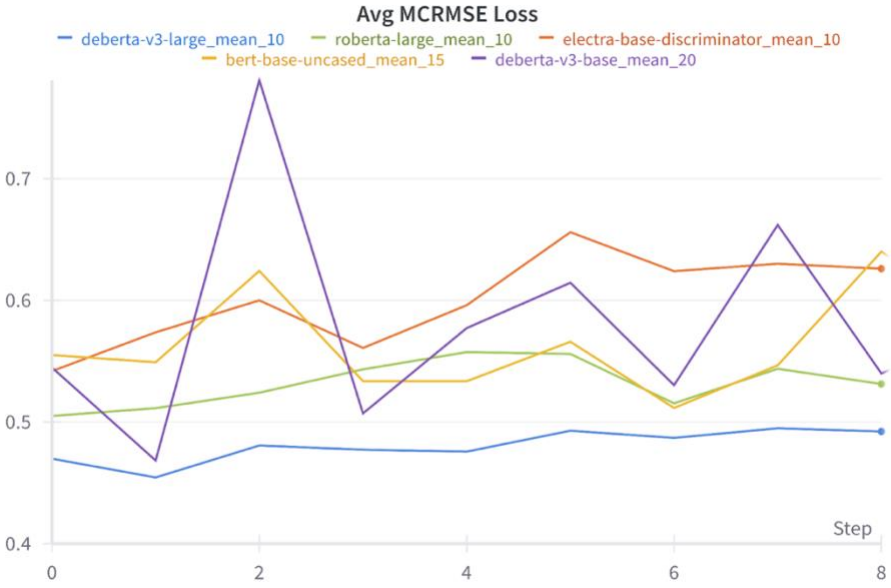

Avg MCRMSE Loss
electra-base-discriminator_conv1d_10
electra-base-discriminator_concat_10
electra-base-discriminator_lstm_10
electra-base-discriminator_mean_10
1
0.9
0.8
0.7
0.6
0.5
0.4
Step
0
2
4
6
8

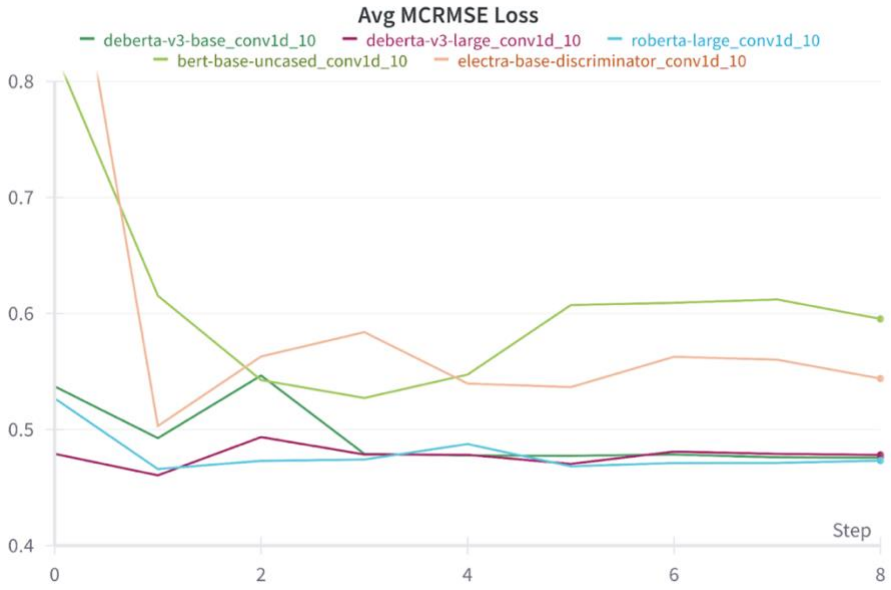

Avg MCRMSE Loss
bert-base-uncased_concat_10
bert-base-uncased_conv1d_10
bert-base-uncased_lstm_10
bert-base-uncased_mean_15
0.8
0.7
0.6
0.5
0.4
Step
0
2
4
6
8

4.4 Comparing a pooling with different backbones






Avg MCRMSE Loss
deberta-v3-large_mean_10
roberta-large_mean_10
electra-base-discriminator_mean_10
bert-base-uncased_mean_15
deberta-v3-base_mean_20
0.7
0.6
0.5
0.4
Step
0
2
4
6
8


Avg MCRMSE Loss
deberta-v3-base_conv1d_10
deberta-v3-large_conv1d_10
roberta-large_conv1d_10
bert-base-uncased_conv1d_10
electra-base-discriminator_conv1d_10
0.8
0.7
0.6
0.5
0.4
Step
0
2
4
6
8

5. Summary and Conclusions

Our project utilized cutting-edge NLP models (BERT, RoBERTa, DeBERTa, and ELECTRA) and diverse pooling methods (Mean, LSTM, Concat, Conv1D) to analyze English proficiency in essays written by 8th-12th grade ELLs. The ELLIPSE corpus served as our dataset, challenging us to predict proficiency scores across multiple linguistic dimensions. We approached this as a multi-label regression problem, with meticulous data preprocessing and model training, applying differential learning rates for optimization. The training involved Mean Squared Error (MSE) loss, while validation used both MSE and a custom metric average Mean Column wise Root Mean Square Error (MCRMSE).

The results highlighted the efficacy of advanced NLP models in educational contexts, particularly in assessing language proficiency, aiding both learners and educators. This project emphasized NLP's potential in enhancing educational tools and methodologies. We showcased NLP's versatility through different learning rates and pooling techniques, contributing to automated essay scoring and language proficiency assessments. Overall, this initiative not only provided insights into language assessment but also opened new avenues for NLP applications in education, marking a significant step in the evolution of language learning and evaluation tools.

6. Future Improvements

In future improvements, a multi-pronged approach can enhance model performance and robustness. First, incorporating pseudo-labeling during pretraining, utilizing it alongside half of the original data, can enrich the model's learning experience. This approach, followed by fine-tuning on the remaining original data, could potentially refine the model's understanding and adaptability to real-world scenarios. Additionally, experimenting with a broader range of pooling methods could uncover more effective strategies for data representation, particularly in complex models. Finally, exploring a wider variety of backbone architectures would offer insights into their respective strengths and weaknesses, enabling more tailored and effective model designs for specific NLP tasks.

7. Code Percentage

$$(500 - 100) / (500 + 555) * 100 = 39.8$$

8. References

- https://www.kaggle.com/code/shreydan/lstm-embeddings
- https://www.kaggle.com/code/javigallego/deberta-from-the-ground-up-2-approaches#Model-Inputs-Explained
- https://www.kaggle.com/code/yasufuminakama/fb3-deberta-v3-base-baseline-train#Model
- https://www.kaggle.com/code/shreydan/using-transformers-for-the-first-time-pytorch#Tokenizer,-Dataset-and-DataLoaders
- https://www.kaggle.com/code/rhtsingh/utilizing-transformer-representations-efficiently
- https://huggingface.co/docs/transformers/model_doc/deberta-v2#transformers.DebertaV2ForTokenClassification
- https://www.kaggle.com/code/nischaydnk/fb3-pytorch-lightning-training-baseline
- https://github.com/amedprof/Feedback-Prize--English-Language-Learning
- https://www.kaggle.com/competitions/feedback-prize-english-language-learning/discussion/369457
- https://github.com/rohitsingh02/kaggle-feedback-english-language-learning-1st-place-solution/tree/main