DATS 6312

Natural Language Processing

Prof. Amir Jafari

Evaluating Language Knowledge of ELL Students

Individual Final Report

Kyuri Kim

Table of Contents

1. Introduction

The proficiency in English language writing is a pivotal skill for academic success, especially for English Language Learners (ELLs) in grades 8 to 12. This project presents an innovative approach to leveraging advanced natural language processing (NLP) techniques to analyze and classify argumentative essays written by English Language Learners (ELLs) from grades 8 to 12. Utilizing the ELLIPSE corpus provided by Vanderbilt University, our objective was to develop a model that could accurately predict proficiency levels across various linguistic dimensions. These dimensions include cohesion, syntax, vocabulary, phraseology, grammar, and conventions, each scored on a scale from 1.0 to 5.0.

To achieve this, we employed transformer-based models, notably BERT, Electra, RoBERTa and DeBERTa, renowned for their efficacy in understanding and processing complex linguistic patterns. Given the nuanced nature of the task, we explored various pooling strategies, including LSTM, Concatenating, Mean pooling and Conv1D, to effectively aggregate contextual information from these models. Our methodology involved fine-tuning these advanced models on the corpus, ensuring that they adapt well to the specific linguistic features present in ELL writings.

The project's significance lies in its potential educational impact. By accurately assessing language proficiency levels, our model aims to provide valuable feedback to ELL students, aiding in their language development journey. Furthermore, it offers educators a tool to expedite the grading process, ensuring that students' language abilities are assessed fairly and accurately.

This project not only demonstrates the applicability of cutting-edge NLP technologies in educational settings but also paves the way for more personalized and effective language learning tools. The insights gained from this project are expected to contribute significantly to the development of automated feedback tools that are more attuned to the unique needs of English Language Learners.

Our work stands out due to its unique approach to multilabel regression, a challenging task in NLP. We combined various backbones with multiple pooling techniques to capture the nuanced features of language proficiency in essays. A distinctive aspect of our training involved the use of differential learning rates, allowing us to fine-tune different parts of the models with varying intensity. This approach was complemented by further fine-tuning the saved models for three additional epochs using slower learning rates. This meticulous and layered training strategy significantly enhanced the model's ability to predict the multifaceted language skills of English Language Learners accurately.

2. Description of my individual work

2.1. EDA Analysis

Exploratory Data Analysis (EDA) is an important first step in any data analysis project. It enables a thorough understanding of the dataset, unveiling patterns, and extracting valuable insights about the underlying data structure. EDA involves visually and statistically analyzing the data, identifying relationships, distributions, and potential outliers. It serves as a foundation for subsequent data preprocessing and model building steps. For this project, our EDA on the ELLIPSE corpus revealed insightful patterns and characteristics inherent in the students' essays.We observed a diverse range of vocabulary and syntactic structures, reflecting the varying proficiency levels of English Language Learners. Analysis of word frequencies and sentence structures helped in understanding common linguistic trends among the students. We also noted correlations between different analytical measures such as cohesion and syntax, which provided deeper insights into how different aspects of language proficiency are interrelated.

2.2. Logistic Regression

Logistic Regression is a powerful and widely used classification algorithm that can be used for both binary and multi-class classification problems. It is a parametric algorithm that works by finding the best fit of a sigmoid function to the training data. The sigmoid function transforms a linear equation into a probability between 0 and 1, which can be interpreted as the likelihood of a particular class given a set of input features.

The logistic regression model is trained using the maximum likelihood estimation method, which involves finding the parameter values that maximize the likelihood of observing the training data. The parameter values are updated iteratively until convergence is achieved. The most commonly used optimization algorithm for logistic regression is the gradient descent algorithm.

The decision boundary for logistic regression is a linear hyperplane that separates the different classes in the feature space. The logistic regression algorithm is sensitive to the scaling of the input features, and it is important to normalize or standardize the input features before training the model.

Logistic regression is a simple yet powerful algorithm that can achieve high accuracy on a wide range of classification problems. It is widely used in various fields such as finance, healthcare, and social sciences. However, it may not perform well in cases where the decision boundary is non-linear or where there are complex interactions between the input features.

2.3. BERT

BERT (Bidirectional Encoder Representations from Transformers) is a pioneering language model developed by Google in 2018. The model is characterized by its bidirectional pre-training, enabling it to consider both left and right context in all layers, making it highly effective for various natural language processing tasks.BERT undergoes pre-training on a large corpus through two unsupervised tasks: Masked Language Model(MLM) and Next Sentence Prediction (NSP).

In the MLM task, random words within a sequence are masked, and BERT is trained to predict the masked words based on the context of surrounding words. The equation governing this task is $\mathrm{Pr}(\mathrm{wordi} \mid \mathrm{context})\mathrm{Pr}(\mathrm{word}_i \mid \mathrm{context})$.

In the NSP task pairs of sentences are sampled, and the model is trained to predict whether the second sentence follows the first in the original text. The corresponding equation is $\mathrm{Pr}(\mathrm{IsNext} \mid \mathrm{sentence1}, \mathrm{sentence2})\mathrm{Pr}(\mathrm{IsNext} \mid \mathrm{sentence}_1, \mathrm{sentence}_2)$.

Following pre-training, BERT can be fine-tuned for specific downstream tasks such as text classification or named entity recognition. At the heart of BERT lies the self-attention mechanism, represented by the equation:

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

BERT stands out for its deep bidirectional understanding of language. The 'base' configuration, a leaner version compared to the 'large' variant, contains fewer parameters while maintaining remarkable performance. The 'uncased' version disregards letter case, enhancing its adaptability across diverse text inputs.

By inherently grasping bidirectional context, BERT excels in comprehending the intricate details within ELL essays. Its proficiency in understanding language nuances greatly contributes to a comprehensive evaluation of language skills. This bidirectional capability, formulated through self-attention mechanisms, allows BERT to discern contextual relationships between words bidirectionally within a sequence.

2.4. Data Preprocessing

In our project, data preprocessing plays a pivotal role in preparing the ELL essays analysis. The corpus was preprocessed for optimal model training.

2.4.1. Data cleaning

For the cleaning, lowercase whitespace, numbers removal was done, along with POS tagging. Attention was paid to ensure the tokenization process retained the linguistic nuances of the ELL essays.

2.4.2. Tokenization

A significant part of this preprocessing involves tokenization, where we convert raw text into a format that is understandable and processable by our models. We implemented the ‘add_special_tokens=True’ setting, ensuring the inclusion of special tokens such as ‘[CLS]’—applied at the beginning of each text for classification tasks—and ‘[SEP]’, used to delineate different texts or segments within a given text.

To manage the tokenized sequences effectively, we employed varying maximum lengths for different model backbones. For Bert-base-uncased, electra-base-discriminator, and roberta large, a maximum length of 512 was set. Deberta v3 base utilized a maximum length of 768, while Deberta v3 large opted for a maximum length of 1024. This tailored approach allowed us to optimize the performance of each backbone model based on its unique characteristics and requirements.

Additionally, we implemented the ‘truncation=True’ parameter, a crucial consideration to ensure that if a text surpasses the specified maximum length (max_length), it undergoes truncation to fit within the designated constraints. This proactive measure ensures that our models effectively handle input texts of varying lengths while maintaining consistency in processing.

3. Describe the portion of the work

During the initial phase of our project, our team collaboratively focused on comprehensive data understanding. I took a lead role in this aspect, conducting exploratory data analysis (EDA) over the entire dataset. This involved generating distribution plots for the scoring dimensions and conducting a detailed EDA to gain a deeper understanding of the data. Subsequently, as each team member completed their individual analyses, I assumed the responsibility of consolidating all code, ensuring consistent formatting, and generating formatted plots that would be instrumental in our presentation and final project report.

In terms of modeling, I spearheaded the development of the basic logistic regression model, which served as the foundational step in our project. This initial model provided a crucial overview before delving into more complex modeling techniques. Additionally, I took charge of building the LSTM and Bert models. Despite facing challenges with the LSTM model, rendering it impractical for our project, the Bert model yielded successful outcomes. However, its performance was comparatively lower when benchmarked against Transformer-based networks.

4. Results

4.1. Baseline Logistic Regression

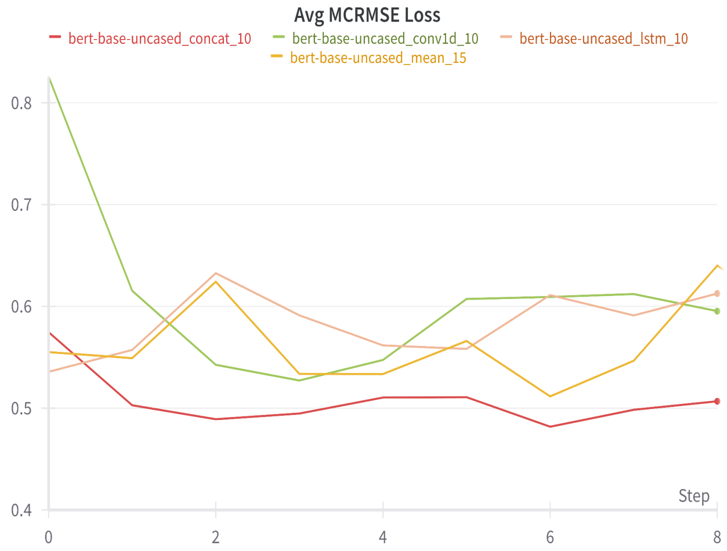
The result of logistic regression is shown as below:

Target	Accuracy
cohesion	0.63
syntax	0.62
vocabulary	0.67
phraseology	0.62
grammer	0.60
conventions	0.62

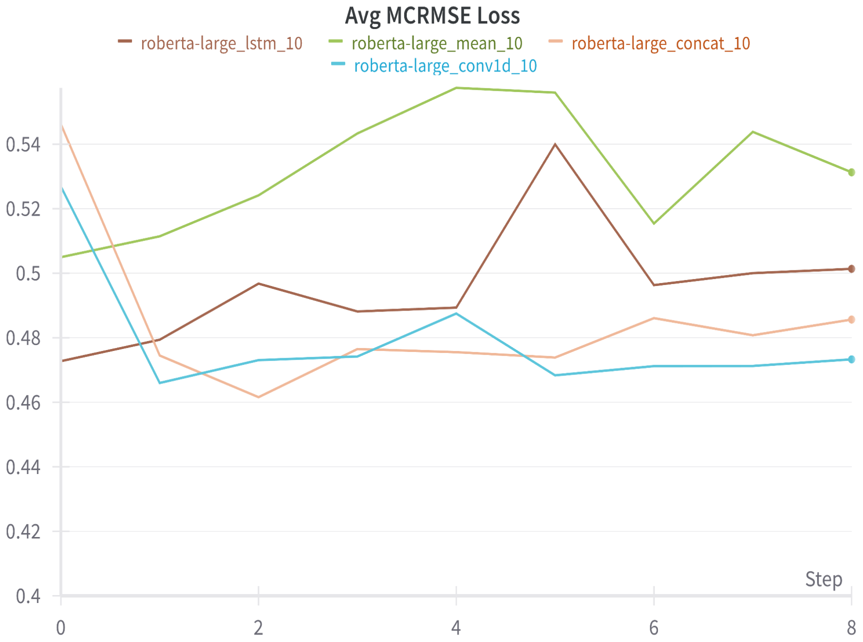
The modeling was done only after the data cleaning step, which was removing lowercase, whitespace, numbers, stop words, and POS tagging. The accuracy isn’t very good, but at the same time not too bad. However, this accuracy isn’t too crucial since this is just a first step before moving on to more complex NLP models.

4.2. Comparing a model with different poolings

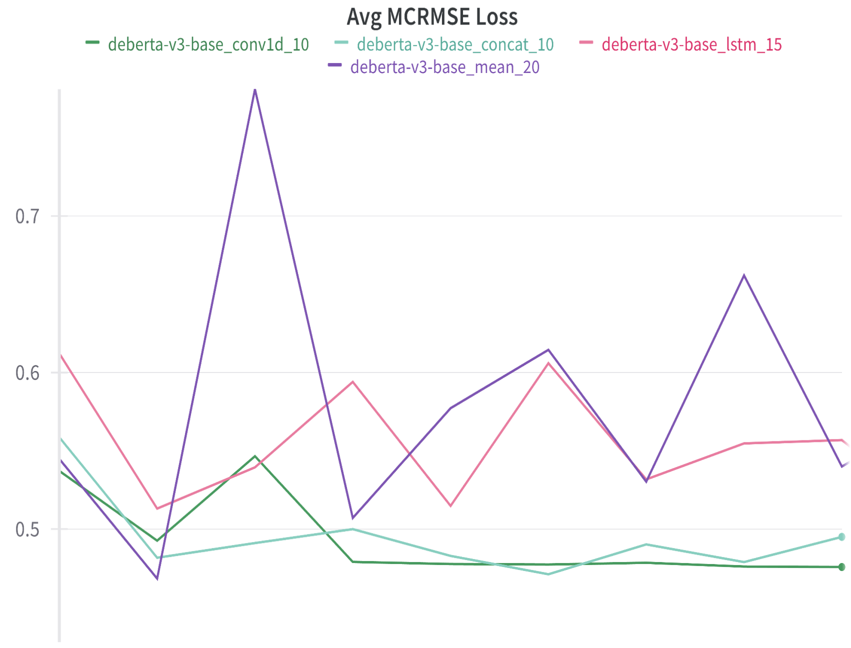
The result compares a model after pre-training with different poolings results is shown below, which is generated by WandB.



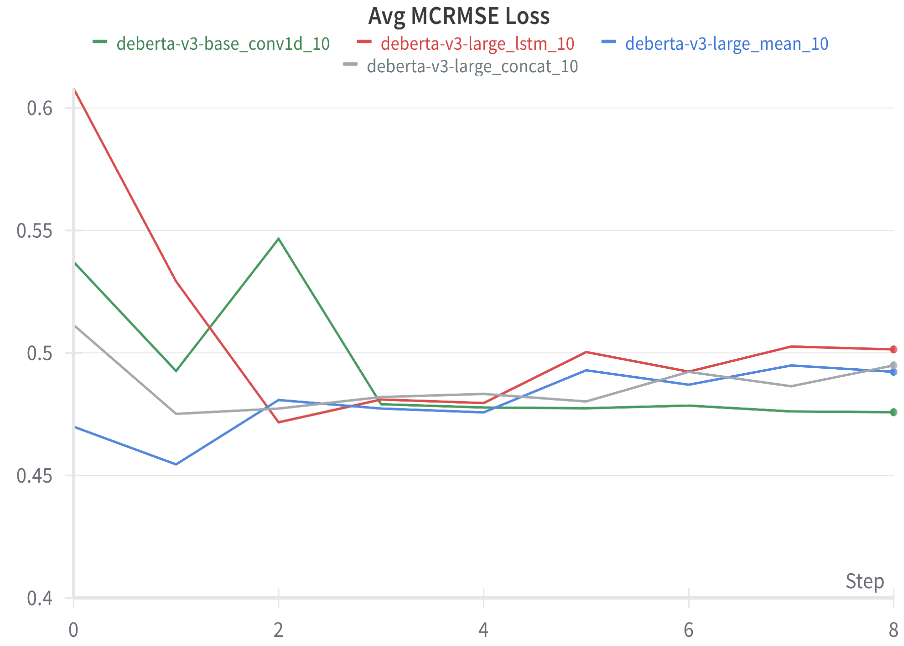
Here, we can see that for the Bert-base-uncased model, with combination of Concat pooling gives the best result of avg MCRMSE loss rate of 0.48.



Next, with the Roberta-large model, with the combination of Conv1D gives the best result of avg MCRMSE loss rate of 0.46.

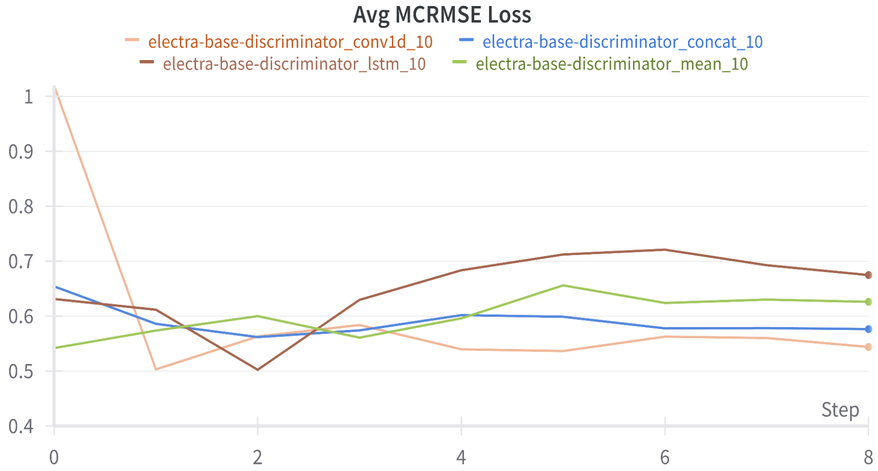


Next, with the Deberta-v3-base model, with the combination of Conv1D gives the best result of avg MCRMSE loss rate of 0.47.



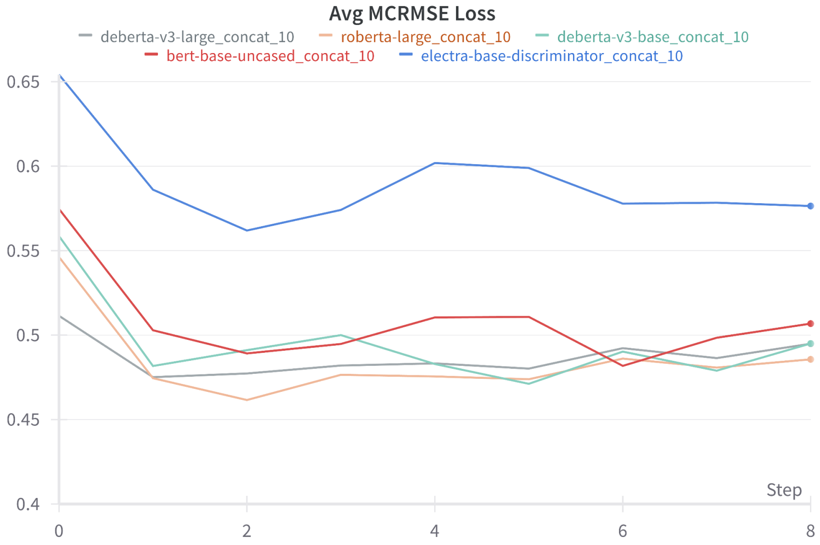
Next, with the Deberta-v3-large model, with the combination of Conv1D gives the best result of avg MCRMSE loss rate of 0.46.

Lastly, with the Electra-base-discriminator model, with the combination of Conv1D gives the best result of avg MCRMSE loss rate of 0.5.

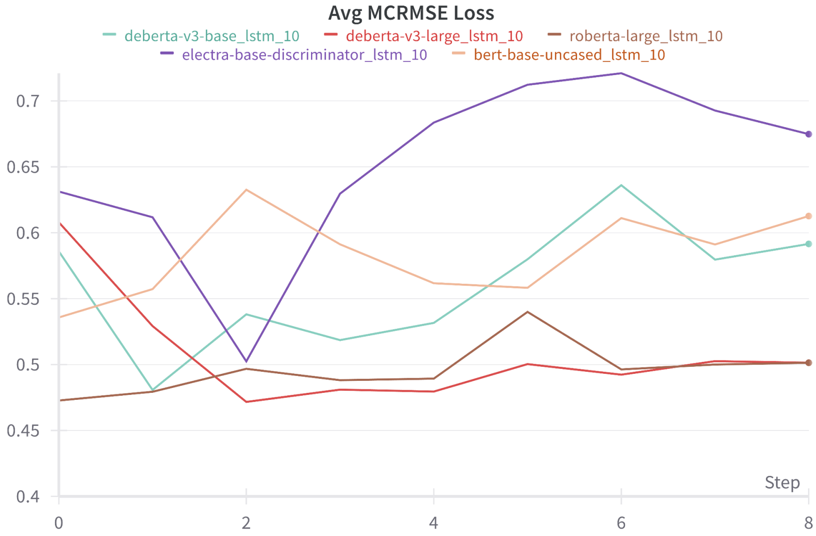


4.3. Comparing a pooling with different models

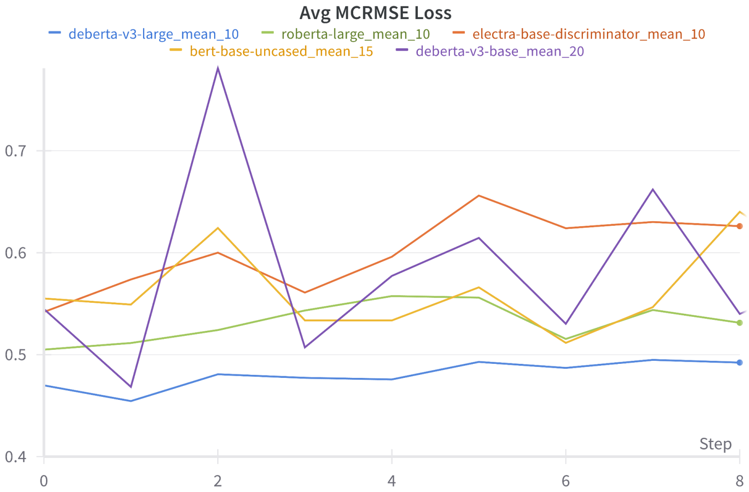
The result compares a pooling after pre-training with different model results is shown below, which is generated by WandB.



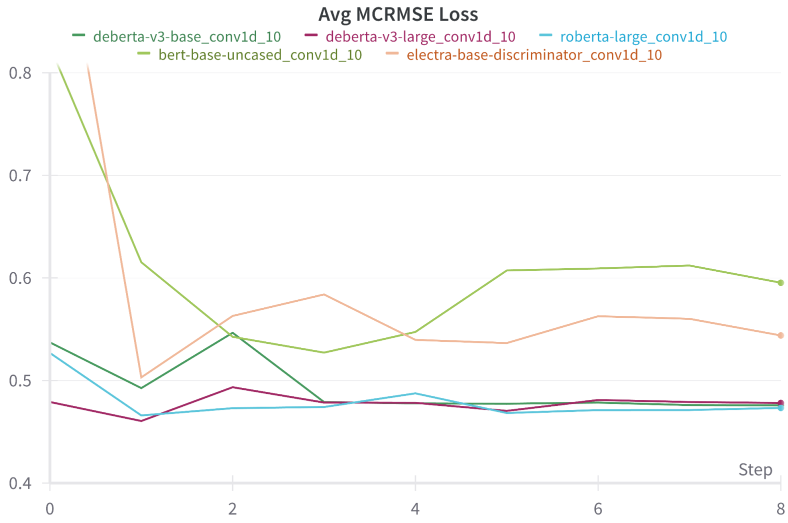
Based on this graph, with the Concat pooling, the modeling performing best is Roberta-large model.



Based on this graph, with the LSTM pooling, the two models that have similar performance are Deberta-v3-large and Bert-base models.



Based on this graph, with the Mean pooling, the Deberta-v3-large seems to work best.



Lastly, the Conv1D pooling, the Roberta-large and Deberta-v3-large models seems to work best.

In summary, the results from different backbone models paired with various pooling techniques, measured by average MCRMSE loss, reveal insightful trends. Notably, the effectiveness of a pooling strategy varies significantly depending on the underlying backbone model.

For BERT-base-uncased, Concat Pooling emerged as the most effective, likely due to its proficiency in integrating multifaceted layer information. Interestingly, the more complex LSTM Pooling was less effective, suggesting a possible misalignment with BERT-base's output structure. Mean and Conv1D Pooling showed moderate success. In the case of Electra-base-discriminator, LSTM and Conv1D Pooling outperformed others, indicating their compatibility with Electra's unique representations. Conversely, Mean and Concat Pooling were less effective, possibly due to a mismatch with Electra's attention mechanism.Roberta-large showed a preference for Concat Pooling, excelling in leveraging its rich, layered representations. Conv1D also performed well, indicating its effectiveness with high-dimensional outputs, while Mean and LSTM Pooling fell short. With Deberta-v3-base, Mean Pooling led the way, suggesting that straightforward averaging is sufficient for capturing Deberta's outputs. LSTM, Concat, and Conv1D Pooling offered comparable performances but didn’t significantly improve results. Finally, for Deberta-v3-large, Mean and Conv1D Pooling stood out, effectively capturing the complex representations of this larger model. Concat and LSTM Pooling didn’t fare as well, potentially due to challenges in managing Deberta-v3-large's high-dimensional outputs. In essence, the compatibility of pooling methods with different backbone models is key. Larger, more complex models like Roberta-large and Deberta-v3-large benefit from simpler pooling approaches like Mean and Conv1D. Conversely, models like BERT-base-uncased gain more from Concat pooling, which effectively integrates information across layers.

4.4. Pre-training vs. Fine-tuning Results

	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

Table 1. Pre-training Results

	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

Table 2. Fine-tuning Results

The fine-tuning phase of our experiment, focused on larger backbone models, reveals intriguing insights when compared to the pre-training results. We chose to fine-tune only larger models like Roberta-large and Deberta-v3 variants, as they have more parameters and complexity, offering greater scope for refinement and optimization through fine-tuning.

In Roberta-large, Concat Pooling significantly outperformed other methods with a score of 0.41, suggesting an excellent synergy between fine-tuning and its ability to leverage information from multiple layers. However, its Mean Pooling score of 0.48 indicates a drop in performance compared to pretraining, possibly due to oversimplification in capturing nuances. Conv1D Pooling also showed notable improvement, aligning well with Roberta's complex representations. LSTM Pooling lagged behind, perhaps due to its complexity not aligning as effectively with the Roberta architecture during fine-tuning. For Deberta-v3-base, Mean Pooling achieved a respectable score of 0.44, suggesting that average pooling captures Deberta's outputs well even after fine-tuning. However, LSTM Pooling scored 0.51, indicating a potential mismatch or overfitting. Conv1D and Concat Pooling showed balanced performance, with Conv1D marginally leading, reflecting its effectiveness in handling Deberta's intricate features. Deberta-v3-large showed remarkable results with Mean Pooling leading at 0.3986, suggesting high overfitting. LSTM Pooling followed suit with a decent performance, indicating its efficacy in handling the complexities post-fine-tuning. Concat and Conv1D Pooling also performed well, although slightly over the threshold of potential overfitting at 0.43 and 0.41, respectively.

Comparing these results with the pre-training phase, it's evident that fine-tuning significantly impacts performance, especially in more complex models like Deberta-v3-large, where sophisticated pooling techniques align well post-fine-tuning. The variance in performance across pooling methods also highlights the nuanced interplay between model architecture and pooling strategy, especially in the context of fine-tuning for optimized performance.

5. Summary and conclusions

Our venture into NLP aimed to elevate the language assessment of English Language Learners (ELLs) by leveraging advanced techniques. Employing from a classical logistic model to transformer-based models like BERT, RoBERTa, DeBERTa, and ELECTRA, along with diverse pooling methods, we effectively captured the nuances of natural language in argumentative essays from the ELLIPSE corpus. This facilitated a multi-label regression approach, predicting proficiency scores with meticulous data preprocessing, model training, and differential learning rates, optimizing using MSE and the custom MCRMSE metric.

Our outcomes highlighted the potential of complex NLP models in education, showcasing their promising role in language assessment for learners and educators. The advanced transformers enabled a nuanced understanding, significantly enhancing assessment accuracy and paving the way for automated scoring and language proficiency assessments, particularly for ELLs.

This project marks a significant step in NLP research for education, not just assessing proficiency but propelling further advancements. Our methodologies inform the development of refined assessment tools, enriching language learning processes. It underscores the transformative potential of NLP in educational assessment, showcasing the pathways opened for its impactful integration.

Looking ahead, enhancing model performance involves a multifaceted approach. Incorporating pseudo-labeling during pretraining and fine-tuning on remaining original data can enrich the model's adaptability. Experimenting with a broader range of pooling methods and exploring diverse backbone architectures will uncover more tailored strategies for effective model designs, especially for intricate NLP tasks.

6. Calculate the percentage of the code

Based on the calculation example in the instruction, around 34% is my own code.

7. References

- Speech and Language Processing, Third Edition, by Daniel Jurafsky and James H. Martin, 2020
- 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