DATS 6312

Natural Language Processing

Prof. Amir Jafari

Evaluating Language Knowledge of ELL Students

TEAM 8

Kyuri Kim, Sanchit Vijay

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 Dataset
2.1. Data Information
The dataset utilized in this project is the ELLIPSE corpus, a comprehensive collection of argumentative essays crafted by English Language Learners (ELLs) from 8th to 12th

grade. This rich dataset, provided by Vanderbilt University, is specifically designed to assess various aspects of English language proficiency among students learning English as a second language. The corpus comprises several key components, each contributing to a thorough evaluation of language skills.

The core of the dataset consists of argumentative essays. These essays are reflective of the students' ability to articulate opinions, construct arguments, and utilize the English language effectively in written form.

Each essay is meticulously scored across six analytical dimensions, crucial for language proficiency assessment. These dimensions include:

- Cohesion: Evaluates the use of connectives and transitions to ensure the text flows logically.
- Syntax: Assesses the complexity and correctness of sentence structures.
- Vocabulary: Measures the range and appropriateness of word choice.
- Phraseology: Analyzes the use of idiomatic expressions and collocations.
- Grammar: Examines the accuracy and diversity of grammatical constructions.
- Conventions: Checks adherence to writing norms, including spelling, punctuation, and capitalization.

These analytical dimensions are assigned a score ranging from 1.0 to 5.0, in 0.5 increments, signifying the proficiency level in that particular aspect. Along with the essays and scores, the dataset includes metadata such as grade levels, which offers additional context and aids in nuanced analysis.

The ELLIPSE corpus stands out for its focus on argumentative essays, which are particularly challenging yet insightful for language proficiency evaluation. The dataset's comprehensive scoring system provides a multi-dimensional view of each student's language abilities, making it an invaluable resource for developing a model that can accurately assess and predict English writing proficiency among ELLs.

3. Description of the NLP Model and Algorithm

In our project, we started with a classical model, which is logistic regression and then incorporated a diverse array of state-of-the-art transformer-based models, each renowned for distinct characteristics and strengths.

3.1. 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.

3.2. BERT-base-uncased

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 $\Pr(\text{wordi} \mid \text{context})\Pr(\text{word}_i \mid \text{context})$.

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 $\Pr(\text{IsNext} \mid \text{sentence1},\text{sentence2})\Pr(\text{IsNext} \mid \text{sentence}_1,\text{sentence}_2)$.

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:

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

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.

3.3. Electra Base Discriminator

The Electra Base Discriminator introduces an innovative pre-training method centered on discerning between 'real' and 'fake' words in sentences. By concentrating on identifying these substitutions, the model hones its capacity to comprehend intricate textual nuances.

This model's specialty lies in its keenness to identify artificial replacements within sentences, enabling it to capture fine-grained textual intricacies effectively. In our project, the Electra Base Discriminator's proficiency in recognizing subtle language patterns proves highly valuable. It plays a crucial role in evaluating and understanding the nuanced linguistic elements present in ELL essays, particularly in aspects like syntax and phraseology.

3.4. RoBERTa Large

RoBERTa Large, an optimized rendition of BERT, enhances pre-training methodologies to elevate overall performance. This 'large' variant boasts additional layers and parameters, enabling a more profound contextual comprehension.

This model's augmented capacity for contextual analysis proves highly effective, especially in comprehending intricate sentence structures and diverse vocabulary within ELL compositions. Its enhanced capabilities significantly contribute to understanding the complexities inherent in language, aiding in the assessment of English Language Learner writings.

3.5. DeBERTa v3 Base

DeBERTa, integrating a disentangled attention mechanism, refines the understanding of word relationships within sentences, surpassing BERT's capabilities. The v3 base version strikes a balance between size and computational efficiency, offering an optimal model for various tasks.

Its advanced attention mechanism plays a pivotal role in precisely evaluating intricate elements like cohesion and grammar within student essays. This feature enhances the model's ability to discern nuanced language structures, facilitating more accurate assessments in educational contexts.

3.6. DeBERTa v3 Large

The larger variant, DeBERTa v3 Large, amplifies the model's capacity with additional layers and parameters, significantly enhancing its ability to comprehend language intricacies. With this expanded depth in contextual analysis, the model excels in capturing and evaluating the multifaceted linguistic nuances present in ELL essays.

All these models have been pre-trained on extensive text corpora, allowing them to understand a wide range of linguistic patterns and contexts. When fine-tuned on our

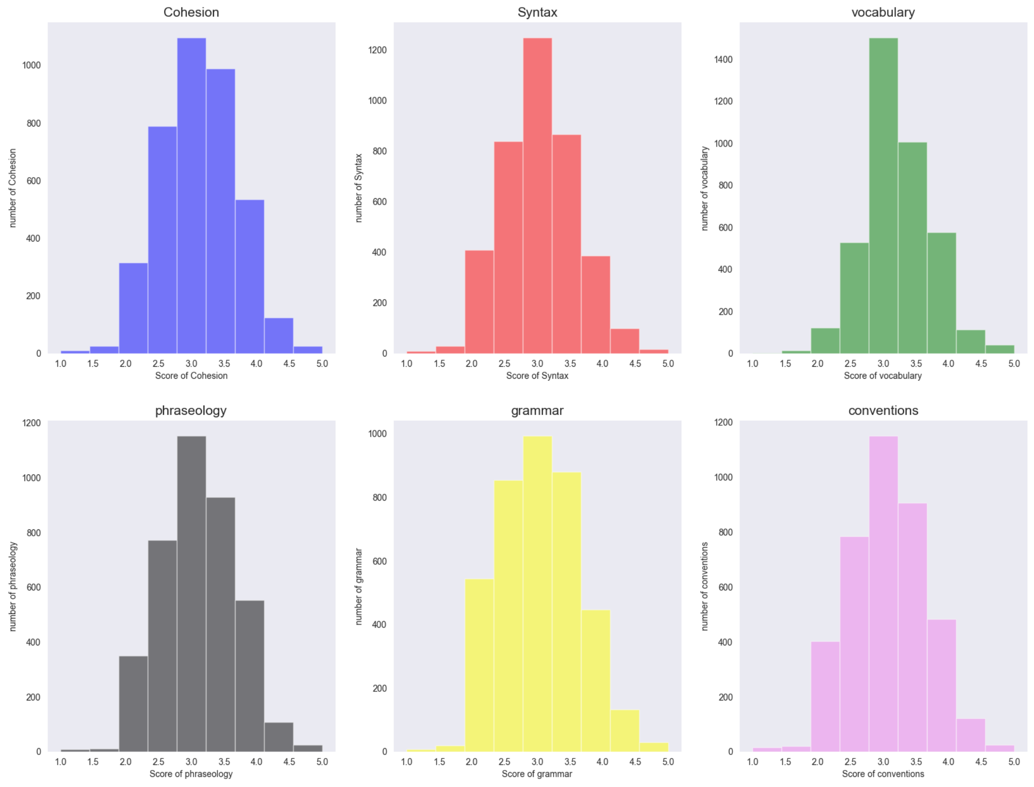
ELL essay dataset, these models adapt to the specific linguistic characteristics of ELL writings. This includes understanding varying levels of language proficiency, identifying common errors made by language learners, and assessing the use of language in an argumentative context. The choice of diverse models also allows us to compare their effectiveness in this unique application, ensuring we utilize the best possible tool for assessing language proficiency in educational settings.

4. Experimental setup

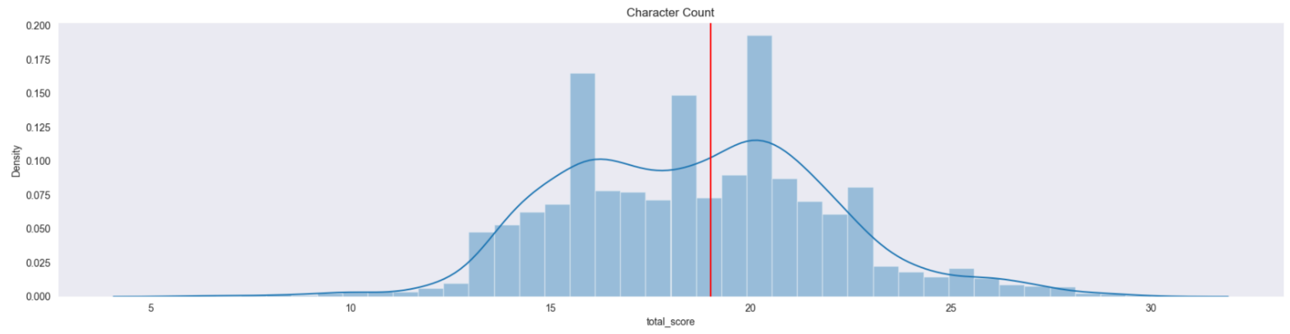
Our experimental setup are designed as following:

4.1. EDA Analysis

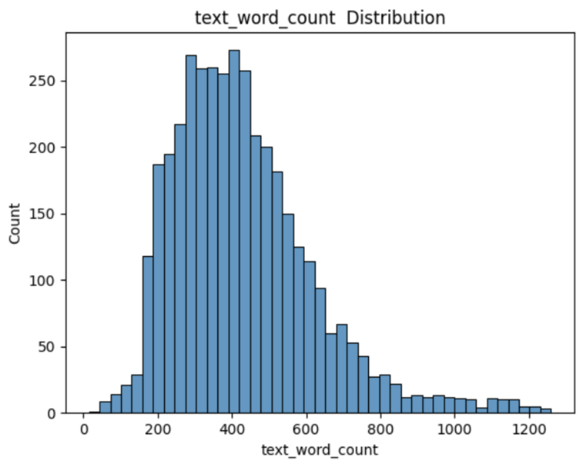
Exploratory Data Analysis (EDA) is an important first step in any data analysis project. Our EDA on the ELLIPSE corpus revealed insightful patterns and characteristics inherent in the students' essays. We examined the six numerical variables which are scoring dimensions. Here are histograms of the six scoring dimensions:

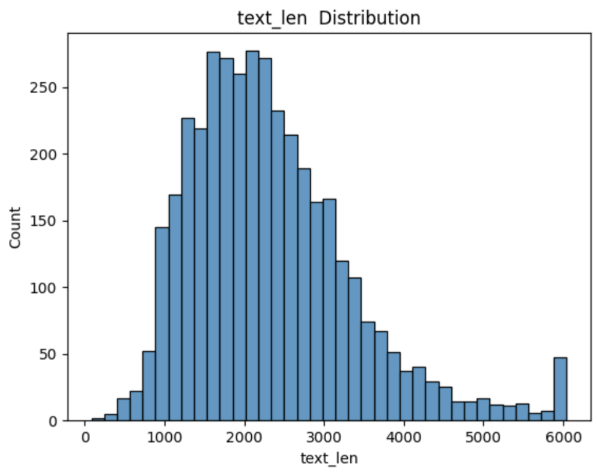


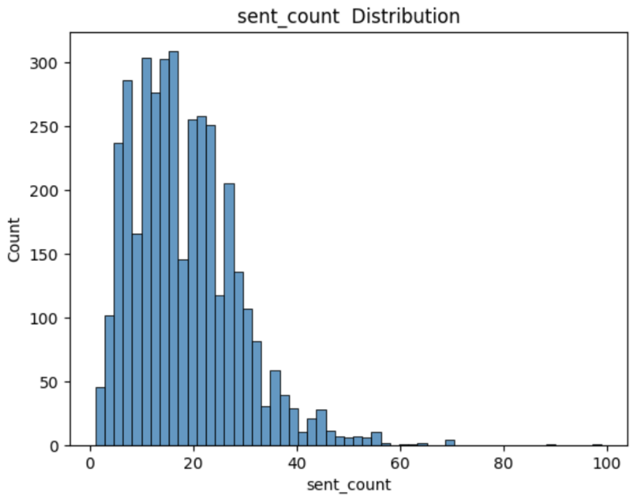
We can see that all the histograms for the six dimensions are normally distributed.



This last histogram plot shows distribution of the total score. Total score is calculated by adding all of the six dimensions per essay.

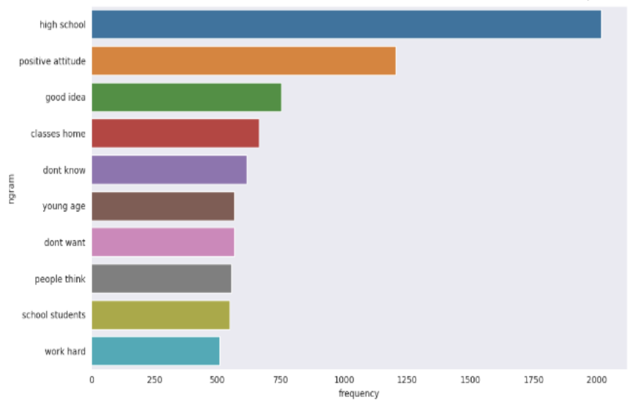


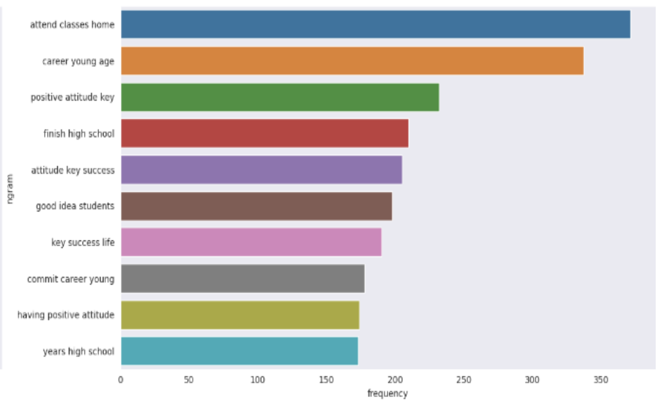


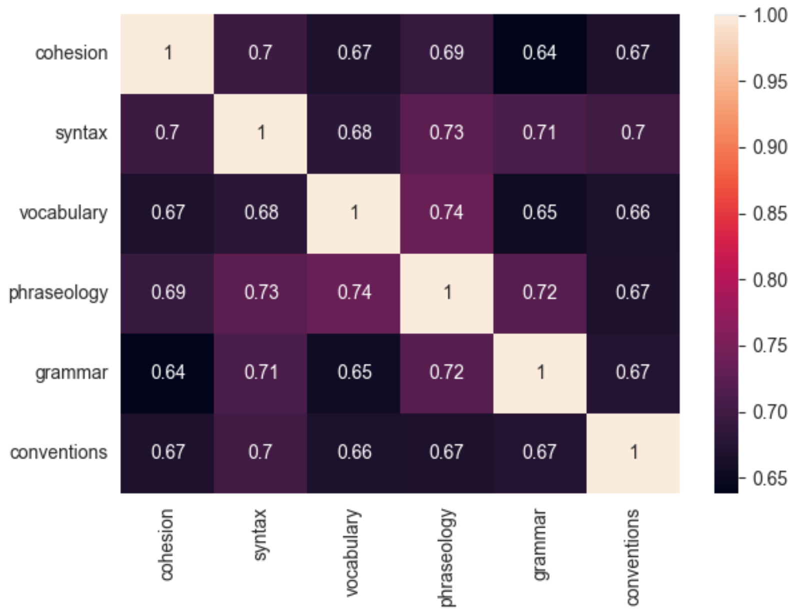


Also, we have more histograms here for the number of words, length of text, and number of sentences in the essay.

Next, here is the result of the most frequent ngrams. This analysis provides insights into the language structure, common phrases, or sequences prevalent in the dataset. In our dataset, 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.

4.2. 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.

4.2.1. 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.

4.2.2. 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.

4.2.3. Splitting Data for Training, Validation, and Unseen Testing

We have strategically partitioned our dataset into different segments for training, validation, and unseen testing. 99% of the data is used for training (60%) and validation (40%). This split ensures that our models are trained on a substantial amount of data, allowing them to learn the nuances of language proficiency effectively. The validation set is crucial for fine-tuning model parameters and for preventing overfitting.

The remaining 1% of the data is set aside as unseen data. This subset is crucial for demonstrating the model's performance in real-world scenarios, such as in our Streamlit app. By using data that the model has never seen before for demonstration purposes, we can provide a more accurate representation of the model's capabilities in assessing language proficiency in new, unobserved essays.

Overall, the careful preprocessing of our data, along with the thoughtful partitioning for training, validation, and unseen testing, ensures that our models are well-trained, thoroughly evaluated, and ready for practical application.

4.3. Pooling

Pooling techniques play a critical role in summarizing and extracting essential features from the outputs of transformer models. For our project, we have utilized four different pooling methods: Mean pooling, LSTM pooling, Concat pooling, and Conv1D pooling. Each of these pooling methods offers unique advantages and works effectively with our transformer models.

4.3.1. Mean Pooling

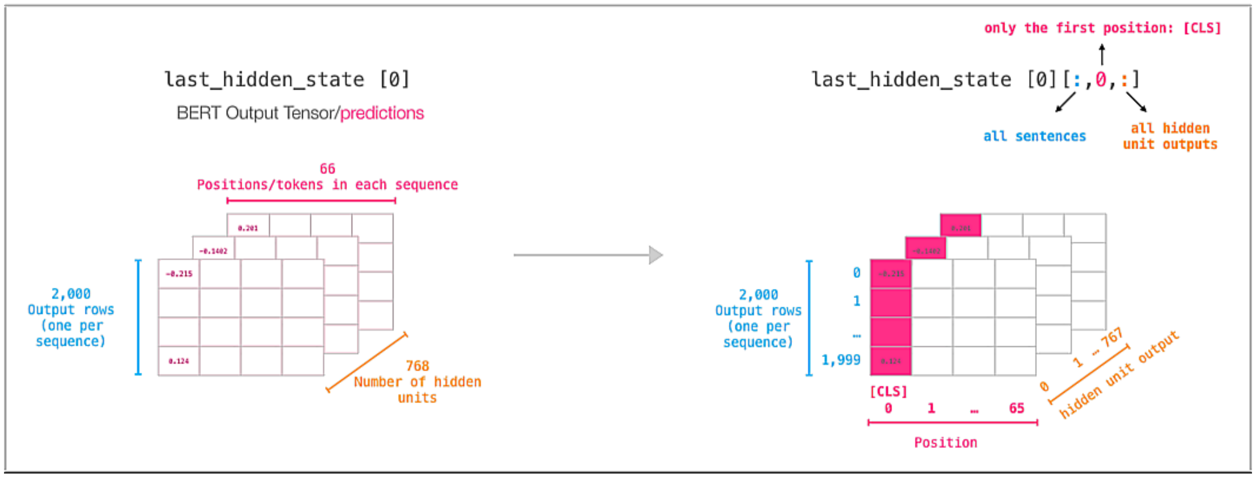
Mean pooling involves taking the average of all token embeddings in the output sequence of a transformer model. This method provides a simple yet effective way to condense the information across the entire input sequence into a single fixed-size representation. It's particularly useful for capturing the overall semantic meaning of an essay, making it suitable for assessing general language proficiency and comprehension.

4.3.2. LSTM Pooling

LSTM pooling uses a Long Short-Term Memory (LSTM) network to process the sequence of embeddings from the transformer output. This approach captures sequential dependencies and contextual relationships between words. The ability of LSTM to understand sequence and context makes it ideal for assessing linguistic elements like syntax and cohesion in student essays, which are crucial for language proficiency.

4.3.3. Concat Pooling

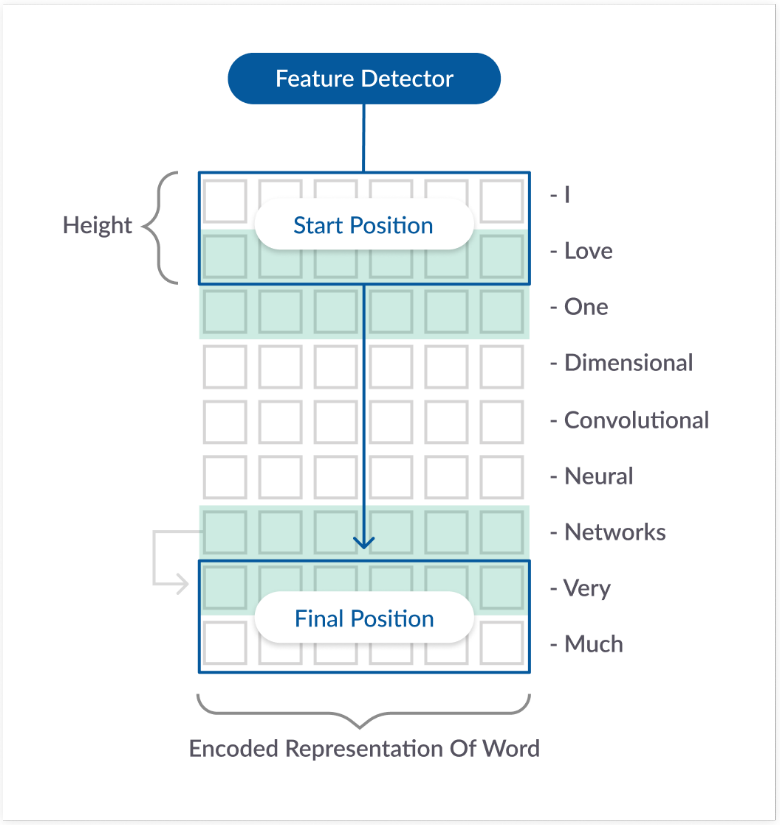
Concat pooling involves concatenating the last hidden states of several layers of the transformer model. This method provides a richer representation by combining information from different abstraction levels. By leveraging insights from multiple layers, concat pooling is adept at capturing both high-level concepts and finer linguistic details, offering a comprehensive assessment of various language skills.


last_hidden_state [0]
BERT Output Tensor/predictions
66 Positions/tokens in each sequence
2,000 Output rows (one per sequence)
768 Number of hidden units
only the first position: [CLS]
last_hidden_state [0][:,0,:]
all sentences
all hidden unit outputs
0
1
...
1,999
[CLS]
0 1 ... 65
Position
0 1 ... 767
hidden unit output

4.3.4. Conv1D Pooling

Conv1D pooling applies a one-dimensional convolutional neural network over the transformer output sequence. This method is effective in capturing local contextual features within the sequence. Its ability to focus on local dependencies and patterns makes it particularly suitable for identifying specific linguistic features like phraseology and vocabulary usage in ELL essays.

1D CONVOLUTIONAL - EXAMPLE



Each pooling technique brings a different perspective in processing and interpreting the output of transformer models. When paired with our diverse set of transformer models, they complement the inherent strengths of each model. For instance, mean pooling, with its simplicity, pairs well with all models to provide a baseline understanding of the text. In contrast, LSTM pooling enhances models like BERT and RoBERTa by adding an additional layer of sequence understanding. Concat pooling is particularly effective with models like DeBERTa, which already have a sophisticated understanding of different word relationships. Conv1D pooling, with its focus on local features, complements Electra's discriminator capabilities in identifying fine-grained text details.

When applied to our dataset of ELL essays, these pooling techniques help in distilling the essential linguistic features relevant to language proficiency assessment. They enable the models to focus on different aspects of language use, from overall coherence and grammar to specific phraseology and vocabulary, providing a well-rounded evaluation of student essays.

4.4. Training Setup

4.4.1. Evaluation Metrics

The task at hand is a regression problem rather than a traditional classification one. Each essay is scored on several analytic measures, each with a continuous range of scores. This setup aligns with regression, where the goal is to predict continuous or quasi-continuous values.

MSE loss is appropriate for this regression task because it penalizes larger errors more severely by squaring the difference between the actual and predicted values. This characteristic helps in fine-tuning the model's predictions to be as close as possible to the actual scores.

While MSE loss is used during training, for validation, we utilize both MSE loss and Mean Column wise Root Mean Squared Error (MCRMSE). MCRMSE provides an aggregate measure of error across all labels, offering a holistic view of the model's performance. It's especially useful in our multi-label setup as it accounts for the variability in the model's prediction accuracy across different scoring dimensions like cohesion, syntax, etc.

4.4.2. Optimizer

This approach involves using different learning rates for different parts of the model. In our case, we have distinct learning rates for the encoder, embeddings, and decoder.

- Encoder (2e-5): The encoder, being a crucial component for understanding text representations, requires a careful, moderate learning rate to adjust weights effectively without missing subtleties in the text.
- Embeddings (7e-6): Embeddings form the base representations of words; a lower learning rate ensures that these representations are fine-tuned delicately, preserving their semantic richness.
- Decoder (3e-5): The decoder is responsible for generating the final scores. A slightly higher learning rate here allows for more flexible adjustments in the later stages of the model, ensuring better alignment with the regression targets.

Using differential learning rates allows each part of the model to learn at a pace suitable to its role and complexity. This method can lead to better overall performance as it caters to the specific learning needs of different components within the model.

4.4.3. Learning Rate Scheduler

The training process incorporates a Cosine Schedule with Warmup, a scheduler that dynamically adjusts the learning rate throughout training. This schedule initiates with a warm-up phase where the learning rate gradually increases, followed by a cosine annealing schedule.

During the warm-up phase, the learning rate's initial ascent enables the model to converge faster in the early stages of training. This is advantageous as it assists the optimizer in escaping local minima, facilitating a more efficient optimization process.

Following the warm-up, the learning rate undergoes a gradual decrease according to a cosine curve during the cosine annealing phase. This approach proves beneficial for fine-tuning the model by making smaller adjustments as it approaches convergence.

Consequently, this mitigates the risk of overshooting the optimal point in the loss landscape, contributing to a more refined and stabilized convergence.

The overall impact of combining a warm-up phase with cosine annealing is a more effective and stable training process. This is particularly advantageous for complex models engaged in nuanced tasks, such as ours. The approach ensures that the model undergoes rapid learning in the initial phases and then refines its understanding more subtly in the later stages, leading to enhanced performance and convergence.

4.5. Logging

In our project, Weights and Biases (WandB) plays a key role in experiment tracking and results visualization. WandB systematically records a variety of experiments, facilitating easy comparisons and replication of specific model versions and hyperparameter settings. This capability proves invaluable for comprehending the impact of each modification on the model's performance.

Moreover, WandB offers real-time logging of key metrics, such as loss and accuracy, accompanied by sophisticated visualization tools. This real-time monitoring enables us to gain insights into the model's performance, facilitating informed decision-making throughout the development process.

In summary, Weights and Biases significantly contributes to our project's efficiency by providing a comprehensive platform for experiment tracking and sharing, thereby enhancing our ability to understand, compare, and optimize machine learning experiments and outcomes.

4.6. Hyperparameters

The hyperparameter settings provided play a pivotal role in configuring and optimizing our model for performance. The inclusion of a seed value (42) ensures reproducibility in our experiments, promoting consistency across runs. The logger settings are tailored to manage project tracking, incorporating a specific project name and key, with an option to save logs for reference and analysis.

Dataset parameters are carefully defined to handle data aspects, such as batch sizes (12 for training, 32 for validation, optimized for different model types), maximum sequence length (512), and target labels. Model configurations encompass the backbone selection, providing options for gradient checkpointing and various pooling methods. Specific parameters like LSTM settings (e.g., hidden size of 1024) and Conv1D configurations (e.g., 128 filters) are specified for their respective pooling types.

In the pre-training phase, critical parameters include the number of epochs (10), learning rates (encoder_lr: 2e-5, embeddings_lr: 7e-6, decoder_lr: 3e-5), and optimization settings like weight decay (1e-8) and maximum gradient norm (1000). These values are meticulously chosen to strike a balance between learning efficiency and computational constraints, aiming for optimal model performance.

During fine-tuning, a nuanced adjustment of hyperparameters is implemented. Learning rates are decreased (encoder_lr: 2e-6, embeddings_lr: 1.5e-6, decoder_lr: 9e-6), and weight decay is increased (0.01). This strategic approach ensures gradual and precise adjustments to the model weights, avoiding overfitting and enhancing generalization. Simultaneously, higher weight decay in fine-tuning serves to regularize the training process, preventing overfitting by penalizing larger weights, ultimately contributing to the development of a more robust and reliable model.

Overall, these hyperparameters are tailored to manage the model's training process effectively, ensuring efficient learning and robust performance. Below table shows the hyperparameters common for pre-training and fine tuning.

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

5. Results

5.1. Baseline Logistic Regression

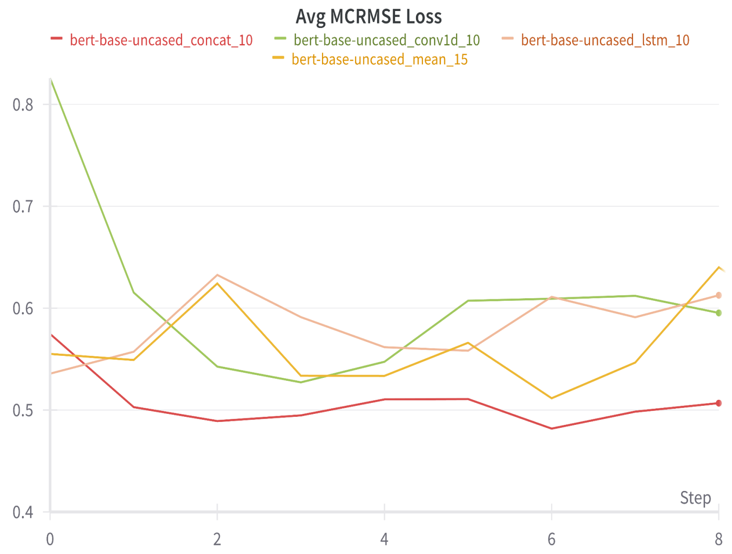
The result of logistic regression is shown as below:

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

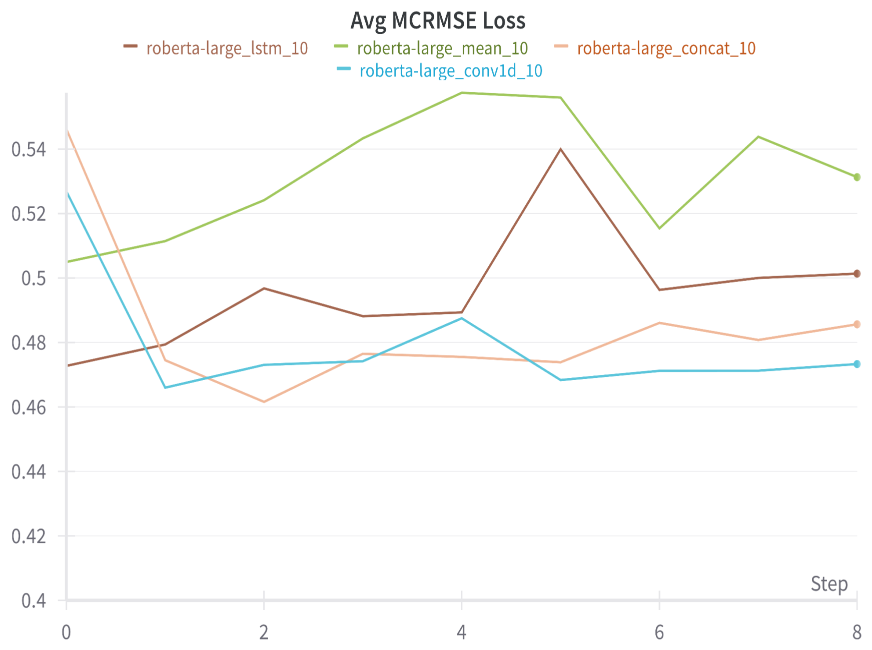
The modeling was done only after the data cleaning step, which was removing lowercase, whitespace, numbers, stop words, and POS tagging. The accuracy isn't very good, but at the same time not too bad. However, this accuracy isn't too crucial since this is just a first step before moving on to more complex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

5.2. Comparing a model with different poolings

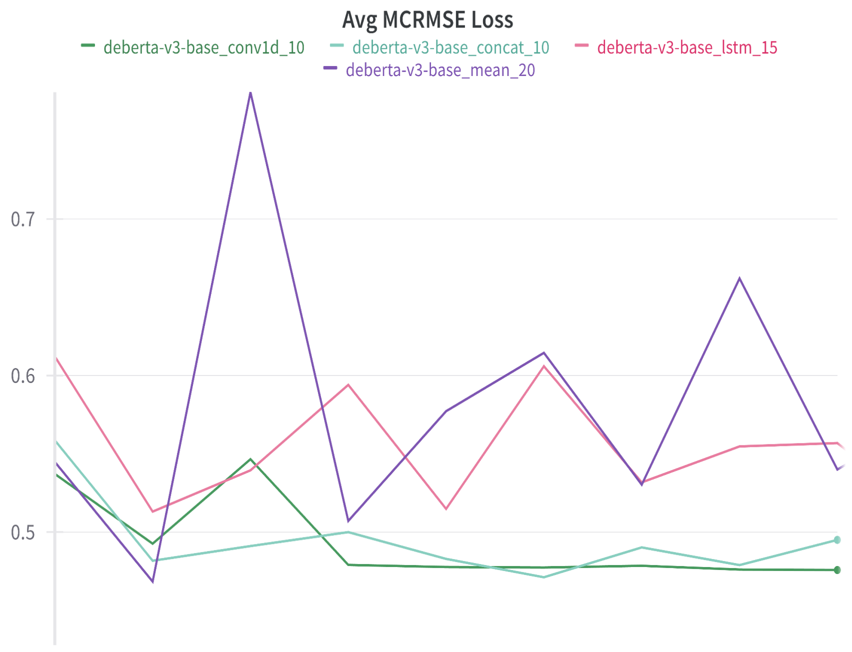
The result compares a model after pre-training with different poolings results is shown below, which is generated by WandB.



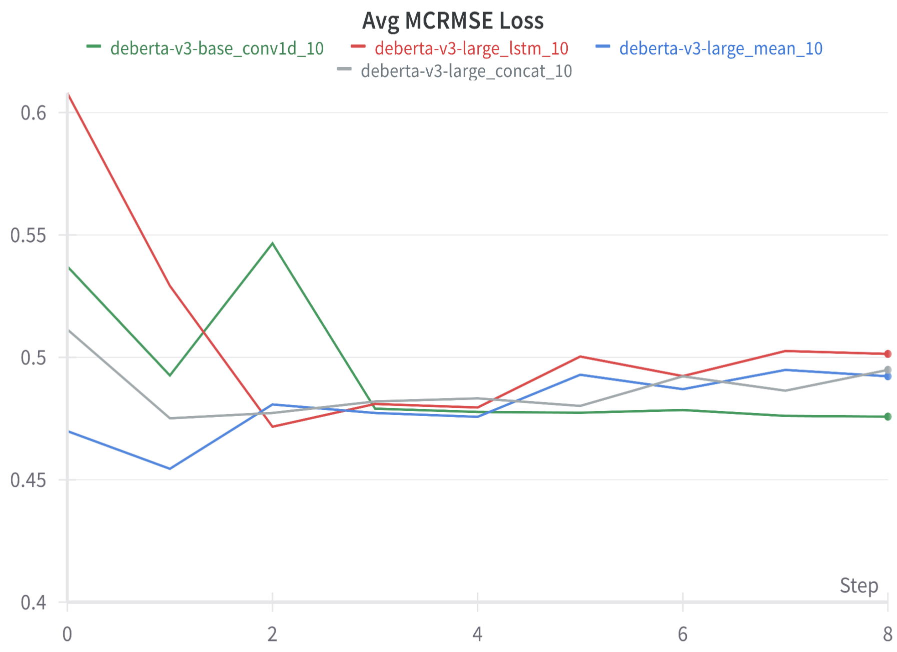
Here, we can see that for the Bert-base-uncased model, with combination of Concat pooling gives the best result of avg MCRMSE loss rate of 0.48.



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

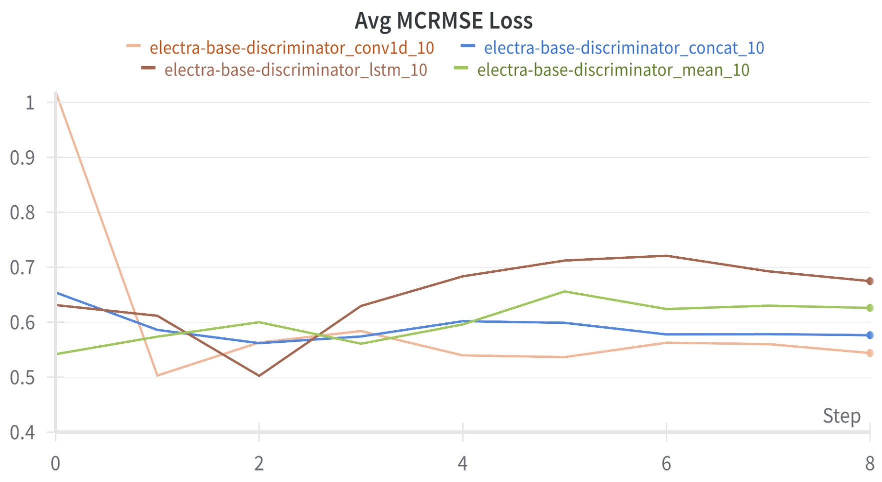


Next, with the Deberta-v3-base model, with the combination of Conv1D gives the best result of avg MCRMSE loss rate of 0.47.



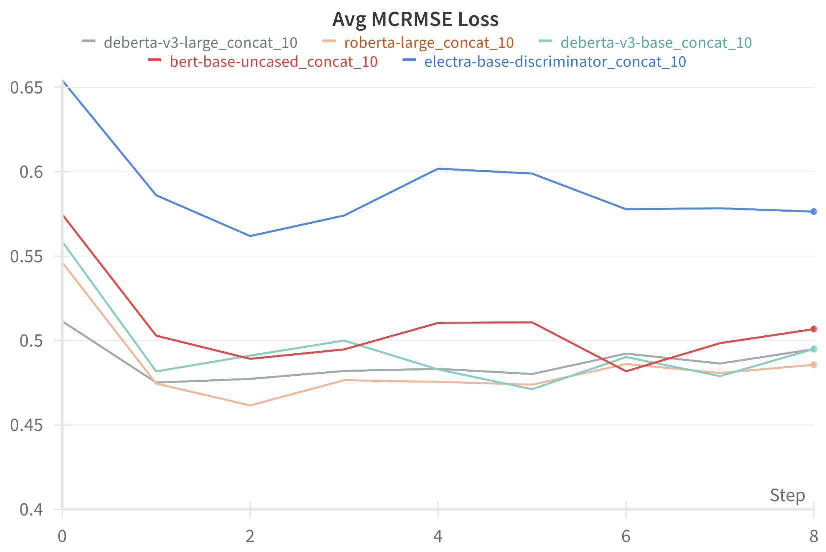
Next, with the Deberta-v3-large model, with the combination of Conv1D gives the best result of avg MCRMSE loss rate of 0.46.

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

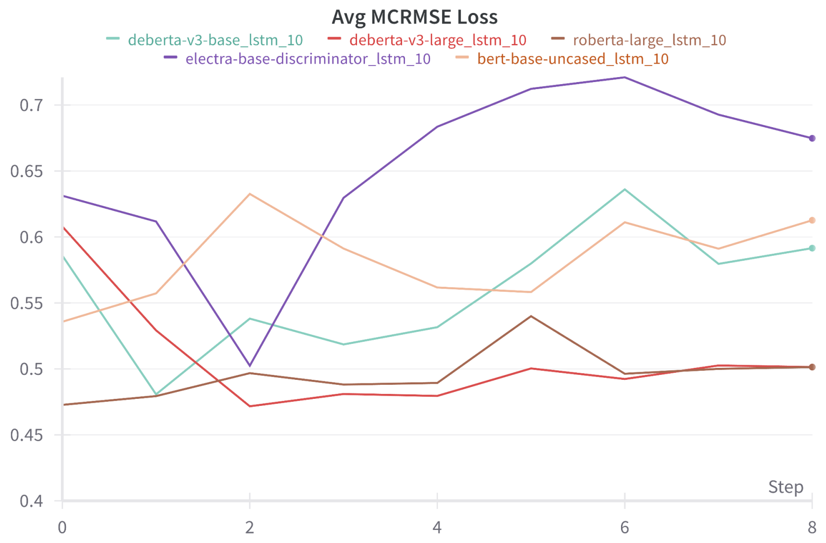


5.3. Comparing a pooling with different models

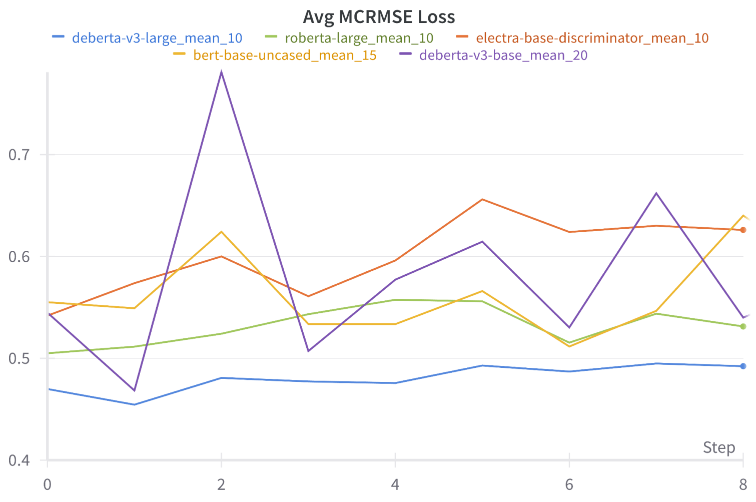
The result compares a pooling after pre-training with different model results is shown below, which is generated by WandB.



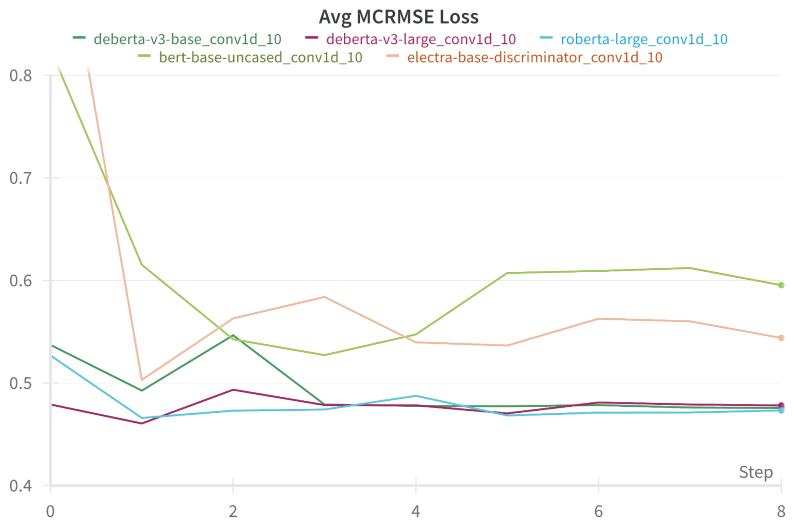
Based on this graph, with the Concat pooling, the modeling performing best is Roberta-large model.



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.

5.4. 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.

6. 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.

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