DATS 6312

Natural Language Processing

Prof. Amir Jafari

***Evaluating Language Knowledge of ELL Students***

TEAM 8

Individual Report

by

Sanchit Vijay

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.

1. **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**

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.

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.

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.

[A white board with text and numbers

Description automatically generated with medium confidence](https://www.kaggle.com/code/rhtsingh/utilizing-transformer-representations-efficiently?scriptVersionId=108281817&cellId=7)

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

[A screenshot of a computer

Description automatically generated](https://www.kaggle.com/code/rhtsingh/utilizing-transformer-representations-efficiently?scriptVersionId=108281817&cellId=19)

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

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.

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.

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.

Conclusion

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 |

1. **Results**

**5.1. Results of pre-training**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 |

**5.2. Results of fine-tuning**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 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 |

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.

The fine-tuning phase of our experiment, focused on larger backbone models, reveals intriguing insights when compared to the pretraining results. I 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 0.4130, suggesting an excellent synergy between fine-tuning and its ability to leverage information from multiple layers. However, its Mean Pooling score of 0.4815 indicates a drop in performance compared to pretraining, possibly due to over-simplification 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 0.4492, suggesting that average pooling captures Deberta's outputs well even after fine-tuning. However, LSTM Pooling scored 0.5125, 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 0.4358 and 0.4131, respectively.

Comparing these results with the pretraining 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.

1. **Comparing a backbone with different poolings**

A graph of a loss

Description automatically generated with medium confidenceA graph of different colored lines

Description automatically generatedA graph of different colored lines

Description automatically generatedA graph of different colored lines

Description automatically generatedA graph of different colored lines

Description automatically generated

1. **Comparing a pooling with different backbones**

**A graph of different colored lines

Description automatically generated**

**A graph of different colored lines

Description automatically generated**

**A graph of different colored lines

Description automatically generated**

**A graph of different colored lines

Description automatically generated**

1. **Summary and Conclusions**

Our project embarked on a pioneering journey, harnessing advanced NLP techniques to elevate the language proficiency assessment of English Language Learners (ELLs). Leveraging state-of-the-art transformer-based models—BERT, RoBERTa, DeBERTa, and ELECTRA—I aimed to grasp the intricacies of natural language. Employing diverse pooling techniques—Mean Pooling, LSTM Pooling, Concat Pooling, and Conv1D Pooling—I effectively captured contextual nuances from transformer model outputs.

The ELLIPSE corpus, comprising argumentative essays by 8th-12th grade ELLs, provided a challenging platform. Tackling this as a multi-label regression problem, I predicted scores for analytic measures like cohesion, syntax, vocabulary, among others.

Meticulous data preprocessing, model training, and differential learning rates optimized our models. I utilized Mean Squared Error (MSE) loss for training, validating models using MSE and a custom metric—average Mean Columnwise Root Mean Square Error (MCRMSE).

The outcomes are multifaceted. Demonstrating the application of complex NLP models to real-world educational challenges, our models exhibited promising language proficiency assessment capabilities, aiding learners and educators. Advanced transformers facilitated a nuanced text understanding, improving assessment accuracy.

This project underscores NLP's potential in educational settings, especially supporting ELLs. Insights gained pave the way for automated essay scoring and language proficiency assessments. Differential learning rates and diverse pooling techniques showcased NLP's adaptability in complex tasks.

In conclusion, this project not only assessed language proficiency but also catalyzed further NLP research in education. Our methodologies and findings inform the development of refined language assessment tools, enhancing language learning processes. The success achieved and pathways opened highlight the transformative potential of NLP in education and assessment.

1. **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.

1. **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>