CommonLit - Evaluate Student Summaries

https://www.kaggle.com/competitions/commonlit-evaluate-student-summaries/overview

GitHub Repo: https://github.com/eladsalama/ML-evaluate-summaries.git

presenting:

Name	Email	Tau User
May Siva	maysiva@mail.tau.ac.il	maysiva
Daniel Palmor	danielpl@tauex.tau.ac.il	danielpl
Elad Salama	eladsalama@mail.tau.ac.il	eladsalama
Ofek Levi	ofeklevi2@mail.tau.ac.il	ofeklevi2

Introduction

The goal of this competition is to assess the quality of summaries written by students in grades 3-12. We built a model that evaluates how well a student represents the main idea and details of a source text guided by a prompt question, as well as the clarity, precision, and fluency of language used in the summary.

This competition marked our first venture into the world of machine learning, and we are thrilled to present the results of our efforts. We developed a model that successfully evaluates student summaries, providing insights into their comprehension and writing skills. Our journey included data preprocessing, feature engineering, model training, and evaluation.

Challenges Faced

One of the major challenges we encountered in this competition was the limited amount of data provided. With only 4 prompts available for training and over 120 prompts in the private test set, it was difficult to ensure that our model could generalize well. This scarcity of data made it easy for the model to overfit, capturing noise rather than meaningful patterns.

Balancing the complexity of the model was another significant challenge. On one hand, a model with low complexity might not capture the intricacies of the data, leading to poor generalization. On the other hand, a highly complex model could easily overfit the limited training data, performing well on the training set but poorly on the unseen test set. Achieving the right balance between these extremes was critical for developing a robust model.

Methodology and Techniques

Throughout the project, we learned about various advanced machine learning techniques and tools, including DeBERTa, LSTM, mean folds predictions, k-fold cross-validation, semi-supervised learning, self-meta pseudo labels, and data augmentation. Our experience with these methods was enlightening and significantly enhanced our understanding of machine learning and NLP. This project has equipped us with valuable skills and insights, and we are excited to apply what we've learned to future endeavors.

Key Ideas that Worked:

1. Replacing RoBERTa with DeBERTa: This improved the performance of our model.
2. Utilizing All Hidden States of DeBERTa: Leveraging the full potential of DeBERTa's hidden states.
3. Using Head Mask: Enhanced the model's ability to focus on important features.
4. Incorporating LSTM: Added recurrent layers to capture sequential dependencies.
5. Multi Sample Dropout: Improved regularization and model robustness.
6. Using Exponential Moving Average (EMA): Helped in stabilizing the training process.
7. Log Transformation on Content and Wording: Improved the distribution and handling of input features.
8. Effective Hyperparameters and Optimizers:
   - Batch size: 4
   - Epochs: 12
   - Optimizer: AdamW
   - Weight decay: 1e-4
   - Scheduler: Cosine
   - Initial Learning Rate: 0.00015
   - Warmup steps: 100

Ideas that Didn't Work:

1. Transformers Other Than DeBERTa: They didn't perform as well.
2. Freezing Layers of DeBERTa: This approach was counterproductive.
3. Using Fewer Than 25 Hidden Layers: Reduced model performance.
4. Randomized Search: Less effective in finding optimal parameters.
5. Loss Functions Other Than MCRMSE: Did not improve results.
6. L2 and L1 Regularization: Ineffective in this context.
7. Combining LGBM with the Existing Model: Despite creating 6000+ features and conducting feature selection, this approach did not yield better results.
8. Meta Pseudo Labels: Ineffective in improving model performance.
9. Data Augmentation with Synonym Replacement and Back Translation: Did not contribute to better performance.
10. Ensemble of the Same Model with Different Seeds: No improvement observed.
11. Mean Fold Prediction: Ineffective.

Expected Compute And Storage Requirements

Files size - 3.45 MB.

Computation power - Kaggle Notebook \ Google Colab is sufficient.

The Data

With limited data at our disposal, we embarked on our project, determined to extract maximum value from the available resources.
Here are the tables we received:

summaries_train.csv – Contains the student summaries and their labels:

student_id - The ID of the student writer.

prompt_id - The ID of the prompt which links to the prompt file.

text - The full text of the student's summary.

content - The content score for the summary. The first target.

wording - The wording score for the summary. The second target.

	student_id	prompt_id	text	content	wording
0	000e8c3c7ddb	814d6b	The third wave was an experimentto see how peo...	0.205683	0.380538
1	0020ae56ffbf	ebad26	They would rub it up with soda to make the sme...	-0.548304	0.506755
2	004e978e639e	3b9047	In Egypt, there were many occupations and soci...	3.128928	4.231226
3	005ab0199905	3b9047	The highest class was Pharaohs these people we...	-0.210614	-0.471415
4	0070c9e7af47	814d6b	The Third Wave developed rapidly because the ...	3.272894	3.219757

prompts_train.csv - Includes four training set prompts. Each prompt comprises the complete summarization assignment given to students.

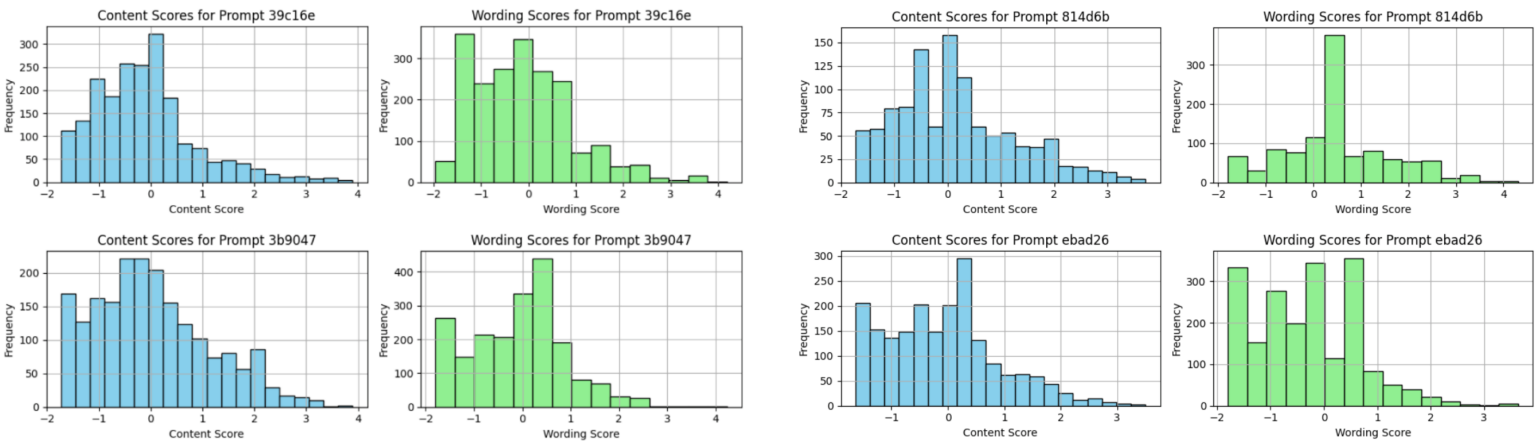
prompt_id - The ID of the prompt which links to the summaries file.

prompt_question - The students guiding question for crafting summaries.

prompt_title - A short-hand title for the prompt.

prompt_text - The full prompt text.

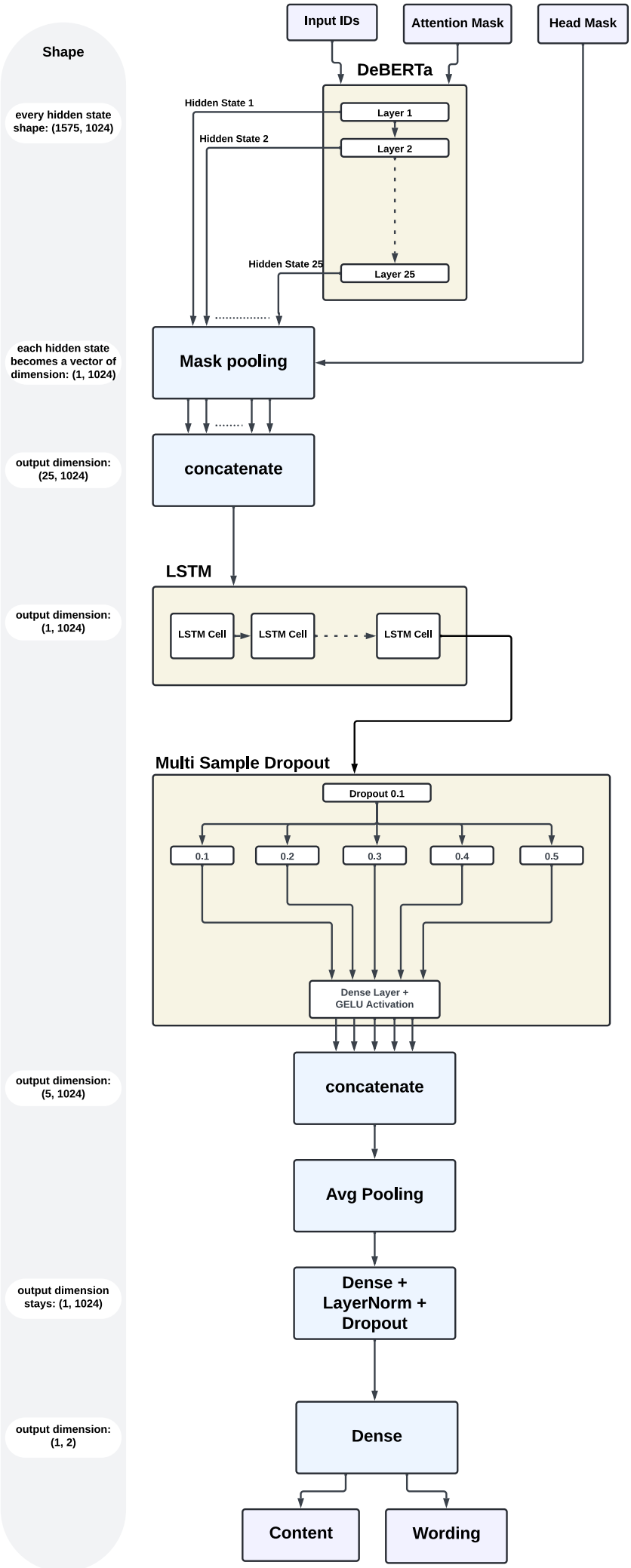
	prompt_id	prompt_question	prompt_title	prompt_text
0	39c16e	Summarize at least 3 elements of an ideal trag...	On Tragedy	Chapter 13 \r\nAs the sequel to what has alrea...
1	3b9047	In complete sentences, summarize the structure...	Egyptian Social Structure	Egyptian society was structured like a pyramid...
2	814d6b	Summarize how the Third Wave developed over su...	The Third Wave	Background \r\nThe Third Wave experiment took ...
3	ebad26	Summarize the various ways the factory would u...	Excerpt from The Jungle	With one member trimming beef in a cannery, an...



As we can see in these figures, there is a distinct distribution of content and wording in each given prompt. This necessitates that our model must emphasize each prompt individually.

Because the test data contains more than 120 prompts, which caught our attention, We realized the need for robustness beyond standard cross-validation practices.

Model Diagram



Lessons learned

This being our first machine learning project, we gained invaluable hands-on experience and deepened our understanding of several key concepts and techniques in the field. Here are some of the significant lessons we learned:

Large Language Models (LLMs):

DeBERTa: We discovered the power of transformer-based models like DeBERTa, which excels at capturing semantic nuances in text.

LSTM: Long Short-Term Memory networks provided us with the ability to model sequential data effectively, capturing dependencies and patterns across the text that simpler models might miss.

Data Preproccesing:

We learned the critical importance of thorough data preprocessing. Addressing missing values, normalizing data, and handling outliers significantly improved our model's performance. This step ensured data quality and consistency, helping our models generalize well to new, unseen data.

Model Optimization with Limited Data:

K-fold Cross-Validation: This technique was crucial in validating our model across different subsets of the data, ensuring that our results were robust and not overfitted to a particular subset.

Self Meta Pseudo Labels: Although did not yield much success. We learned about this innovative approach hoping it would iteratively refine our model's predictions, even with a small training dataset.

Multi Sample Dropout: This technique improved regularization and model robustness by applying dropout multiple times during training, reducing overfitting.

Combining Models:

LGBM: Despite creating over 6000 features and conducting extensive feature selection, integrating LGBM with our existing model did not yield better results. This highlighted the challenges of combining different types of models and the importance of careful feature engineering and selection.

Collaboration and Project Management:

Teamwork taught us the essential value of dividing tasks, sharing insights, and seamlessly integrating different components. This collaborative effort was pivotal for the progress of our project, highlighting the significance of teamwork in achieving our objectives

Progress summary

Method	K-fold cross validation Loss (MCRMSE)
RoBERTa + fully connected layers (Midpoint Review)	0.6
DeBERTa + fully connected layers + LSTM layers + Early Stopping + reduce learning rate on plateau + head mask	0.47
DeBERTa + fully connected layers + 2 LSTM layers + dropout + Early Stopping + cosine decay + head mask	0.443

After reaching 0.443 on the val-loss, we submitted the notebook. After which, we received a private score of 0.566 which indicated that our model introduced data leakage.

The following are the results after fixing the data leakage:

Method	Grouped K-fold Loss (MCRMSE)
DeBERTa + fully connected layers + 2 LSTM layers + dropout + Early Stopping + cosine decay + head mask	0.56
DeBERTa – All hidden states + 1 LSTM layer + Early Stopping + cosine decay + head mask	0.526
DeBERTa – All hidden states + 1 LSTM layer + Early Stopping + cosine decay + head mask + Multi Sample Dropout + Fully Connected Layers	0.51
Log Transformation + DeBERTa – All hidden states + 1 LSTM layer + Early Stopping + cosine decay + head mask + Multi Sample Dropout + Fully Connected Layers	0.47

And the final results are:

Submission	Private Score	Public Score
Random baseline	2.23374	2.13725
First submission	0.56659	0.49590
Final submission (Kaggle bronze medal)	0.48301	0.49939

Additional Ideas for Improvements

In seeking further improvements, incorporating contrastive learning could potentially improve the model's ability to distinguish between different types of student summaries by learning to recognize subtle differences and similarities in the data, thereby enhancing its overall performance.

In addition, it might be worth trying to apply meta-psuedo labels again once our model is more robust.