EECS 595 Final Project Report: AES(Automatic Essay Scoring) for ELLs(English Language Learners)

Yiwei Peng
University of Michigan
ywcpeng@umich.edu

Suhyun Jung
University of Michigan
suhyunj@umich.edu

Youngmin Kim
University of Michigan
yngmkim@umich.edu

Abstract

Traditionally, research on Automatic Essay Scoring(AES) mainly focuses on essays written by native English speakers and has only one type of essay score(overall score). This creates a bias against English Language learners as a second language. In this project, we aim to train a model that focuses on scoring essays written by them and has various types of scores(cohesion, syntax, vocabulary, phraseology, grammar, conventions). In this project, we implemented several state-of-the-art models that perform well in essays written by native English speakers. Also, we developed a novel two-step method for training the combined model. We found that the best prediction models for each type of score are different. In the end, we combined our several models based on the validation result(QWK: Quadratic Weighted Kappa) and achieved an average result(QWK) of 0.669 in the task.

1 Introduction

Automated Essay Scoring (AES) has been a popular domain in natural language processing since the release of the Automated Student Assessment Prize (ASAP) dataset in 2012[1]. However, previously state-of-the-art methods (Peter Phandi and Ng, 2015; Crossley, 2016; Dong and Zhang, 2016; Wei Song and Cheng, 2020; Ruosong Yang and He, 2020; Tirthankar Dasgupta and Saha, 2018; Masaki Uto and Ueno, 2020; Sharma A., 2021) have been more focused on scoring the essays of English native speakers instead of English Language Learners (ELLs). Composing is an important skill when learning English, unfortunately, only a few ELL students are able to hone it, often because writing tasks are rarely assigned in school. English Language Learners (ELLs), students learning English as a second language, are especially affected by the lack of practice. While automated feedback tools make it easier for instructors to assign more writing tasks, they are not trained with ELLs in mind. Existing tools are unable to provide feedback based on the language proficiency of the student, resulting in a final evaluation that may be skewed against the learner.

For ELLs, automatic essay scoring is essential since, compared with native speakers, they usually have fewer chances to have English teachers that can assess their English essays. However, English essays written by ELLs may be more challenging to grade than those written by native speakers since their essays may resemble the grammar or phrases used in their mother tongue. This task is presented in the challenge Feedback Prize - English Language Learning[2], where Vanderbilt University and the Learning Agency Lab released a dataset focusing on essays written by 8th-12th grade ELLs. The goal of our project is to work on this dataset, explore the issues when implementing previous state-of-the-art models on this dataset, and modify or make a model that is more dedicated to scoring essays of ELLs.

2 Datasets

This project focuses on implementing models on the dataset from the Kaggle Competition "Feedback Prize - English Language Learning" released by Vanderbilt University and the Learning Agency Lab in 2022. The dataset is focusing on essays written by 8th-12th grade ELLs. The training dataset comprised 3911 essays with 6 scores for each essay representing cohesion, syntax, vocabulary, phraseology, grammar, and conventions. The scores range from 1.0 to 5.0 in increments of 0.5. Therefore, there are 9 possible values of scores.

Also, there were test datasets, but the Kaggle Competition didn't open test data to public(We can

[1]The Hewlett Foundation: Automated Essay Scoring. Hosted on Kaggle.

[2]Vanderbilt University and the Learning Agency Lab. 2022. Feedback Prize - English Language Learning. Hosted on Kaggle

text_id	Cohesion	Syntax	Vocabulary	Phraseology	Grammar	Conventions
0000C359D63E	3.0	4.0	4.0	3.0	5.0	4.0
000BAD50D026	4.0	4.0	4.0	3.0	5.0	3.0
00367BB2546B	5.0	5.0	4.0	3.0	5.0	4.0

Table 1: Examples of our Dataset scorings

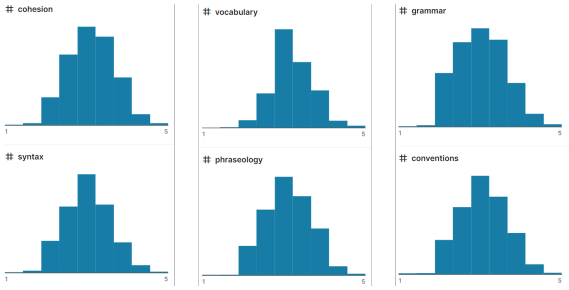

Figure 1: Score distribution of the data

I think that students would benefit from learning at home,because they wont have to change and get up early in the morning to shower and do there hair. ... some teacher dont know how to teach it in then way that students understand it. that causes students to fail and they may repeat the class.

Table 2: Examples of an essay in our Dataset

test our model using that test data only on the Kaggle platform.). Therefore, we divided the training data set into three parts, training, validation, and testing. We used 80% data(3,511 essays) for training, 10%(200 essays) for validation, and 10%(200 essays) for testing.

The average string length of each essay in our dataset is 2,335 and the average word count of each essay in our dataset is 430.

As you can see in Figure 1., all of the scores are distributed roughly normally. The mean and standard deviation of each score is similar. The mean is about 3 and the standard deviation is about 0.66 for all scores. The more detailed statistics for scores are in Table 3.

For the train data set, it is comprised of the 'full_text' of each essay, identified by a unique 'text_id'. Each essay in our dataset is given 7 scores for the seven analytic measures mentioned above. These analytic measures comprise the target for the competition. For the test dataset, we give only the 'full_text' of an essay together with its 'text_id'.

score_name	Mean	Std_dev
Cohesion	3.13	0.66
Syntax	3.03	0.64
Vocabulary	3.24	0.58
Phraseology	3.12	0.66
Grammar	3.03	0.7
Conventions	3.08	0.67

Table 3: Detailed statistics of our Dataset scores

3 Related Works

3.1 Traditional Approaches

Traditional approaches mainly focus on developing effective hand-crafted features. There were a couple of models that we found. First, EASE(Crossley, 2016) (Enhanced AI Scoring Engine) was published in 2015. EASE is a popular and commonly compared AES engine that applies text analysis and feature engineering to several regression models. It is also a library that allows for machine-learning-based classification of textual content. It provides functions that can score arbitrary free text and numeric predictors. Its goal is to provide a high-performance, scalable solution that can predict targets from arbitrary values. The use of hand-crafted features has had immense success on the AES task. Common simple features include sentence length, and word count, there are also other features that need more complex engineering, such as readability, textual, and discourse coherence.

TAACO(Tool for the Automatic Analysis of Cohesion)(Peter Phandi and Ng, 2015) was first mentioned in the paper "The tool for the automatic analysis of text cohesion (TAACO): Automatic assessment of local, global, and text cohesion". It calculates 150 indices indicating different kinds of cohesion, including a number of type-token ratio indices, adjacent overlap indices, and connectives indices. It is a freely available text analysis tool that is easy to use. It allows for the batch processing of text files and developed indices related to text cohesion. TAACO is written in Python, but it is implemented in a way that requires little to no knowledge

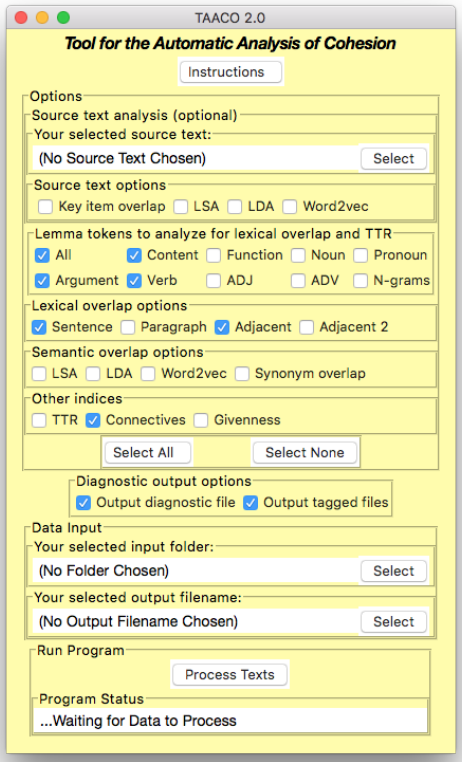

Figure 2: TAACO Ver 2.0

of programming. For a number of indices, the tool incorporates a part-of-speech (POS) tagger from the Natural Language Toolkit and synonym sets from the WordNet lexical database. TAACO differs from other automatic tools that assess cohesion in the way that it reports on a greater number and variety of local, global, and overall text cohesion markers.

3.2 Neural Network Approaches

Since 2016, the Neural Network Model has been broadly used in AES. In the traditional approaches, most researchers focused on how to make good features, but with neural networks, they focused more on the structure of the models instead. Starting from the very first research(Taghipour and Ng, 2016) there were many pieces of research that used neural network models for AES.

Most neural network approaches use LSTM(Long short-term memory), CNN(Convolutional neural network), and their variations(Taghipour and Ng, 2016; Dong and Zhang, 2016; Wei Song and Cheng, 2020). These researchers use multiple neural network models or a hierarchy of models to make better results with the primary features only embeddings of words. They outperform traditional hand-crafted feature approaches.

Also, there were some approaches to using pre-trained models(such as BERT(Jacob Devlin and Toutanova, 2019)) to solve this task. But most of the approaches achieved similar results compared to other neural network approaches. However, among them, some approaches make better results such as R2Bert(Ruosong Yang and He, 2020), which introduced multi-loss(regression loss and ranking loss) to finetune Bert Model.

There were also mixed approaches that used both neural networks(including BERT) and hand-craft features. Tirthankar Dasgupta and Saha (2018) use two DNNs(one for word embedding, the other for the hand-craft features) and concatenate the output of each DNN to pass it to the activation function. There are also other researches using both BERT(Masaki Uto and Ueno, 2020) and hand-craft features.

In the challenge Feedback Prize - English Language Learning, most approaches use DeBERTa (Decoding-enhanced BERT with Disentangled Attention(He Pengcheng, 2021)), which is developed by Microsoft and outperforms the original BERT in various NLP tasks. The highest score among teams that have revealed their approach is also using DeBERTa.

3.2.1 A Neural Approach to Automated Essay Scoring(Taghipour and Ng, 2016)

This very first neural network approach on AES found that even if they use only the text itself (one-hot word embeddings) as the features it outperformed the base-line traditional AES Systems (EASE, which got 3rd prize from the ASAP competition). When using only LSTM+CNN and pre-trained word embeddings of text as inputs, they outperformed EASE by around 6% in terms of quadratic weighted Kappa. When using only LSTM, it also outperformed EASE by about 5%.

3.2.2 Enhancing Automated Essay Scoring Performance via Fine-tuning Pre-trained Language Models with Combination of Regression and Ranking(R2BERT)(Ruosong Yang and He, 2020)

Yang et al.[8] trained a model based on pre-trained Bert and fine-tuned it on the loss combined with

regression loss and batch-wise ranking loss. They exploit the mean square error function for regression loss and use the loss function proposed by ListNet(Zhe Cao and Li, 2007) for batch-wise ranking loss. When combining the two losses into one loss function, they followed Mingrui Wu and Zha (2009) to let the relative weight of ranking loss decrease as the model trains. The model achieved an average Quadratic Weighted Kappa (QWK) score of 0.794 on the 8 prompts of the ASAP dataset. To our best knowledge, this is the highest average score to date.

3.2.3 Feature Enhanced Capsule Networks for Robust Automatic Essay Scoring(Sharma A., 2021)

This method was published in 2021. They compared their model with many recent or commonly used models, including R2BERT, which we just mentioned. The architecture of their model consists of two independent pipelines, which are CapsRater and FeatureCapture. In FeatureCapture, they used the XGBClassifier with the GBTree booster method. The max depth parameter is set to 6, the objective function to multi-softprob, and n_estimators to 1000. For the CapsRater pipeline, they used the standard, cross-entropy loss function. They reported the results on both of these models alone. However, the best-performing model uses a combination of the two methods. To combine these models, they took the mean over their class-wise probabilities and passed the output through a final dense layer to get the resultant score vector. According to the result of the paper, CR+FC outperforms R2BERT in most of the probs of ASAP. The average score it got on ASAP is 0.809, compared with R2BERT's 0.794.

4 Approaches

The goal of this project is to develop a model dedicated to the task of scoring essays of ELLs. First, we did survey for the state-of-the-art models which were well-performed within the dataset of native speakers' essays, and we found several models as a result of the survey. Then, we implemented the previous state-of-the-art methods of the ASAP task to the essays written by ELLs. Finally, we tried to modify the methods and parameters and concatenated the models to get better performance. We also made a combined version of the model that is trained in a two-step manner. In the dataset consisting of essays written by ELLs, we have the scores

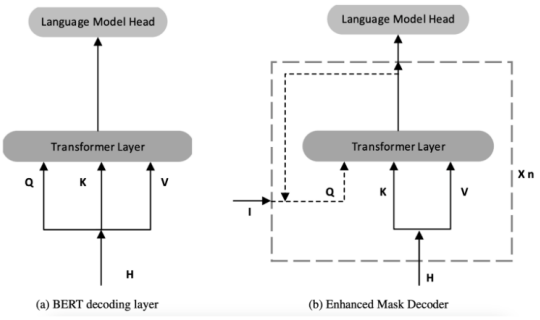

Figure 3: Deberta Model

in 6 different categories. After the presentation, we trained with several different epochs and compared the results of the scores.

4.1 Baseline(DeBERTa)

To be more specific with the models, We used DeBERTa as our base language model because it is the newest state-of-the-art in many NLP tasks. DeBERTa is a Transformer-based neural language model that aims to improve the BERT and RoBERTa models with two techniques: a disentangled attention mechanism and an enhanced mask decoder. Among DeBERTa, We used two pre-trained models, 'DeBERTa-v3-base' and 'DeBERTa-v3-large'.

4.2 Handcrafted Features

We also tried three different models with handcrafted features, which are made by 'Lingfeat(Lee et al., 2021)'. Lingfeat is a Python research package for various handcrafted linguistic features which is introduced by a paper named 'Pushing on Text Readability Assessment: A Transformer Meets Handcrafted Linguistic Features' from EMNLP 2021. Lingfeat makes all 255 different features for our model. These features can be divided into five broad linguistic branches:

First is Advanced Semantic (AdSem). This is for measuring the complexity of meaning structures (Not working in some cases. Working on this issue.) such as Semantic Richness, Noise, and Clarity from trained LDA models (included, no training required).

The second one is Discourse (Disco). This is for measuring coherence/cohesion Entity Counts, Entity Grid, and Local Coherence score.

The third is Syntactic (Synta). This is for measuring the complexity of grammar and structure

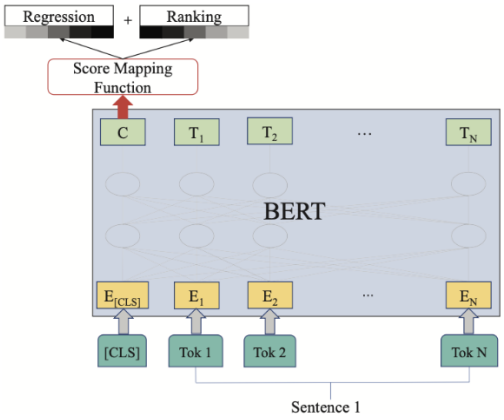

Figure 4: R2Bert Model

Phrasal Counts (e.g. Noun Phrase), Part-of-Speech Counts, and Tree Structure.

The fourth is Lexico Semantic (LxSem). This is for measuring word/phrasal-specific difficulty such as Type Token Ratio, Variation Score (e.g. Verb Variation), Age-of-Acquisition, and SubtlexUS Frequency.

The last one is Shallow Traditional (ShTra). These are the traditional features/formulas for text difficulty such as Basic Average Counts (words per sentence), Flesch-Kincaid Reading Ease, Smog, Gunning Fog.

4.3 R2BERT

In R2BERT, the author proposed a loss function that combined the ranking and regression loss to help the model. At the beginning of training, we put more weight on the ranking loss (ranking within a batch), which helped the model to learn faster in the beginning. The paper achieved one of the highest average accuracies on the ASAP dataset to date. The loss is calculated with formula 1 in below, where the τ_e represents the weight. The weight calculation is followed by Formula 2, where τ_e is a σ function about e calculated as Formula (2). In Formula 2, 'E' is the total epochs, and 'e' is the current epoch, 'y' is a hyper-parameter which is chosen such that 'τ_1' = 0.000001.

$$L = \tau_e * L_m + (1 - \tau_e) * L_r \quad (1)$$

$$\tau_e = \frac{1}{1 + exp(y(E/2 - e))} \quad (2)$$

4.4 Combined Models

We also tried combining the pre-trained models and hand-crafted features. Usually, training neural networks with hand-crafted features requires more epochs than using pre-trained models. However, training too many epochs with a pre-trained model may spoil the original pre-trained weights. As a result, we trained the model in a two-step manner.

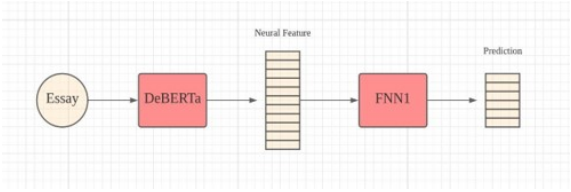

Figure 5: Step one of training our combined method

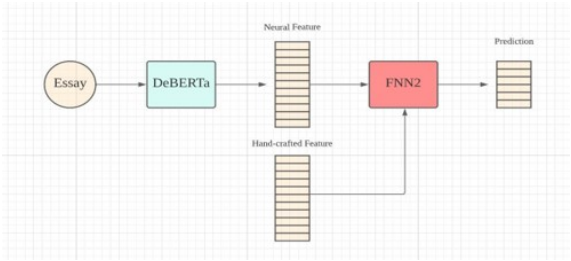

Figure 6: Step two of training our combined method

In the first step, we fine-tuned the DeBERTa for our task. We input the essays (tokenized) into the DeBERTa model and get the feature vector. Then, we passed the feature vectors into a downstream feed-forward neural network to predict our scores for each class.

In the second step, we freezed the pre-trained DeBERTa (we do not compute the gradient for this part). After passing through DeBERTa, we concatenated the output feature from DeBERTa with the hand-crafted features. Then, we train a new downstream layer.

5 Evaluation

5.1 Evaluation Metric

Following the previous works on the ASAP dataset, we use Quadratic Weighted Kappa (QWK) as a performance evaluation metric. QWK is an index measuring the agreement between the prediction and a set of multiclass labels. The score range from 0 to 1.0. The score is 1.0 when the prediction perfectly aligns with the ground truth and the score is 0 for random agreement between prediction and ground truth.

To calculate QWK, we first have to calculate the

Model	Coh	Syn	Voc	Phr	Gra	Con	Average
Base(DeBERTa-base)	0.589	0.648	0.630	0.614	**0.731**	0.666	0.646
Base(DeBERTa-large)	0.588	0.657	**0.639**	0.622	0.729	**0.706**	0.657
R2Bert(DeBERTa-base)	0.605	**0.685**	0.615	0.627	0.694	0.691	0.653
R2Bert(DeBERTa-large)	**0.607**	0.680	0.610	**0.646**	0.703	0.702	**0.658**
Handcraft(Neural)	0.406	0.425	0.334	0.337	0.318	0.407	0.371
Handcraft(SVM)	0.330	0.346	0.344	0.298	0.336	0.321	0.329
Handcraft(Lasso)	0.217	0.293	0.287	0.247	0.137	0.262	0.241
Handcraft+Base	0.465	0.296	0.355	0.378	0.321	0.514	0.388
Handcraft+R2Bert	0.460	0.485	0.527	0.447	0.592	0.604	0.519

Table 4: Result from our Models

weight matrix W according to the formula:

$$W_{ij} = \frac{(i-j)^2}{(N-1)^2}$$

Where i is the ground truth and j is the prediction, and N is the number of possible ratings (labels).

After that, we'll have to calculate an NxN matrix O, O_{ij} corresponds to the number of cases with the ground truth score i and got predicted score j.

An NxN histogram matrix of expected outcomes, E, is calculated as the outer product between the ground truth histogram vector and the predicted histogram vector. Also, we normalize them such that E and O have the same sum.

After calculating those three matrices, the quadratic weighted kappa is calculated with the following formula:

$$k = 1 - \frac{\sum_{i,j} W_{ij} O_{ij}}{\sum_{i,j} W_{ij} E_{ij}}$$

Usually, a score of 0.6 is considered a good score. However, the criteria may differ from case to case. As a reference, in the ASAP dataset, R2BERT achieved an average score of 0.794, which is one of the highest average score for now on the task. Using only LSTM achieved an average score of 0.275 on the task, and using LSTM with mean pooling achieved an average score of 0.602. (Ruosong Yang and He, 2020)

5.2 Experiment Settings

We evaluate our methods on the dataset of 3911 essays. We divide our data into training, validation, and testing sets with an 80%, 10%, and 10% ratio.

Most of the result we report here is trained with 4 epochs on the training set. The handcraft feature with neural regression is trained with 400 epochs. Whereas the Handcraft + DeBERTa is trained with 4 epochs on the pretrained part and 10 epochs on the downstream after concatenating with the handcrafted features. We use the validation data to find the model with the best score within these 4 epochs.

We implemented our models with Pytorch's version of the pre-trained DeBERTa model. The batch size sets to 8 in general and we used the weight decay technique along training to adjust the learning rate during training. When working with handcrafted features, we also normalized all elements from LingFeat to standard normal distribution to avoid overflow during downstream calculation.

We use the Great Lakes, a computing cluster, for training and testing. We ran our model on Nvidia A40 GPU and allocated 48GB GPU memory. It takes about 40 minutes to run 4 epochs using the DeBERTa-base pre-trained model. For DeBERTa-large pre-trained model, it took about an hour and a half.

5.3 Result

As shown in the table, the average score is higher in R2BERT. In detail, the Grammar, Vocabulary, and Conventions scores are better predicted in our base model, and the Phraseology, Syntax, and Cohesion scores are better predicted in the R2BERT model. We believe this is not due to chance, since we found a similar trends from both the validation dataset and the testing dataset. This shows that we can potentially produce better results by using different models for different kinds of scores.

Besides, the results of handcrafted models are all outperformed by the models using pre-trained models.

In the case of combined models. We can see that the ranking loss from R2BERT helped a lot when training the combined model.

ID	Models	P1	P2	P3	P4	P5	P6	P7	P8	Avg
1	EASE(SVR) (Phandi:2015)	0.781	0.621	0.630	0.749	0.782	0.771	0.727	0.534	0.699
2	CNN+LSTM (Tagh:2016)	0.821	0.688	0.694	0.805	0.807	0.819	0.808	0.644	0.761
3	R2BERT (Yang:2020)	0.817	0.719	0.698	0.845	0.841	0.847	0.839	0.744	0.794
4	BERT+Hand features (Uto:2020)	0.852	0.651	0.804	0.888	0.885	0.817	0.864	0.645	0.801

Table 5: QWK scores of the previous works on ASAP dataset

Prompt	#Essays	Avg length	Scores
1	1,783	350	2–12
2	1,800	350	1–6
3	1,726	150	0–3
4	1,772	150	0–3
5	1,805	150	0–4
6	1,800	150	0–4
7	1,569	250	0–30
8	723	650	0–60

Table 6: Statistics of the ASAP dataset

6 Discussion of the Result

6.1 Our Results

First of all, from the related works, we knew that the models based on neural networks perform better than the ones based on hand-crafted features. Besides, based on the result, we can conclude that the state-of-the-art models for the ASAP dataset also applied smoothly to our problem.

As you can see in the Table 1(Result from our Models), the state-of-art BERT family model(DeBERTa base and large) makes quite a good results, with more than 0.65 average QWK scores. Some might think that it is worse than the previous works which made about 0.8 QWK scores on ASAP datasets. However, since there are 9 possible scores(1.0 to 5.0, 0.5 steps), we expected the QWK will be worse than the previous works based on the ASAP data.

6.2 Comparing with the ASAP task

As you can see in the ASAP dataset table(Table 6) most scores have a smaller range than our datasets. Also, the QWK score tends to decrease if there are many possible score values, as you can see the trends in table 5 (results from various papers using ASAP datasets).

Moreover, we can see that the R2BERT model shows worse results with our datasets compared with the one with ASAP. In our datasets, QWK score from the R2BERT output is 0.658 at the highest which is way worse than that from the ASAP datasets which were 0.794 on average. We think that the reason for this difference is the characteristics of the datasets. First of all, as we explained earlier that there are more possible values of the score in our dataset, mostly. However, we can see that for the data with 60 possible values(prompt 8) in the ASAP dataset, the QWK is 0.744 which is much greater than our result, and it is still worse than other ASAP prompt results. Furthermore, our dataset has a different average length(our dataset's average length(# of words) is 450) and has six different scores. In addition, the text of our dataset is written by 8th-12th grade students learning English as a second language, known as English Language Learners, but ASAP's essays are written by native speakers. ELLs may come from all around the world, resulting in a more diverse dataset. Besides, we did not have the chance to put much effort into optimizing the parameters of our R2BERT model, this also can be one reason.

The results that handcrafted models are bad are also interesting. We assume that the model using only hand-crafted features cannot make a good result, and the result is aligned with our expectations. However, we thought that the results from Bert+Handcraft can outperform the Bert-only Model. But it was not the case. We thought that it is because we still need more epochs in the second step. Bert needs only a small number of epochs and the handcrafted model needs more epochs. We trained both steps within 10 epochs for now. We plan to make an more optimized Bert+Handcraft model in the future.

6.3 Assumptions

Finally, we want to explain our assumptions and related result. We assumed that each score is assessed from different aspects of natural language.

	Cohesion	Syntax	Vocabulary	Phraseology	Grammar	Conventions	Average
Score	0.607	0.685	0.639	0.646	0.731	0.706	0.669
Selected Model	R2Bert large	R2Bert base	Base large	R2Bert large	Base base	Base large	

Table 7: The Best Results of our models

Therefore, we tried to apply many different models for each score to find a model that catches their aspects well.

We confirmed our assumptions are true. Different models make different results and the best QWK score comes from all different models. As we wrote down in Section 5. The average score is higher in R2BERT. However, in detail, the Grammar, Vocabulary, and Conventions scores are better predicted in our base model. For Phraseology, Syntax, and Cohesion scores the R2BERT model predicted better. This shows what we should do in the following research. We should try to find the best model for each score and make a sound explanation for that. So far, our result shows that in grammar scores, Bert(DeBERTa) makes a good performance, however, for different scores BERT-based model can't predict well. The following research should find a better model for other scores, this could be other kinds of neural network models or some nice handcrafted features, also a mixed model.

6.4 Future Works

Due to the time limit for this project, we believe that there is still a lot more work that can be done in this project in the future. First of all, as we mentioned, we aim to try more epochs when training the second step of the combined model. Secondly, we can select the best model for each score based on the validation dataset and use the hill climbing techniques to choose our ensemble.

7 Conclusion

We worked on the AES(Automated Essay Scoring) Projects for a month. We tried several different models. One was Deberta(v3-base and v3-large) which was our baseline model. The second one was R2Bert, which had a somewhat interesting loss function. The third one was a hand-crafted model which was made with 255 different features. We used Lingfeat for the features and we tried SVM, Lasso, and Neural Network regression for hand-crafted ones. Lastly, we tried our unique concatenated model.

Following the papers on the ASAP dataset, we used Quadratic Weighted Kappa, which is known as QWK, as a performance metric. As we already mentioned before, the overall score is slightly higher in R2Bert. However, in detail, Grammar and Vocabulary scores are better in base. We found a similar trend from the validation dataset, too. This showed that we can produce better results using different models for different scores. Besides, results of handcrafted models show that the only-handcrafted models are much worse than the Bert. In the case of combined models. The loss function from R2BERT helps a lot when training the combined model.

As we can predict each score with different models, we made a table to show the best results for each score.

8 Acknowledgement

We would like to thank the Advanced Research Computing Technology Services at the University of Michigan for providing computing power on the Great Lakes.

For the dataset we used, we would like to thank Vanderbilt University and the Learning Agency Lab for the dataset. Also, Vanderbilt University and the Learning Agency Lab would like to thank the Bill & Melinda Gates Foundation, Schmidt Futures, and Chan Zuckerberg Initiative for their support in making this work possible.

References

Kyle K. & McNamara D.S Crossley, S.A. 2016. The tool for the automatic analysis of text cohesion (taaco): Automatic assessment of local, global, and text cohesion. *Journal of the Association for Computing Machinery*, 48:1227–1237.

Fei Dong and Yue Zhang. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, page 1072–1077.

Gao Jianfeng Chen Weizhu He Pengcheng, Liu Xiaodong. 2021. Deberta: Decoding-enhanced bert with disentangled attention. arXiv:2006.03654.

Kenton Lee Jacob Devlin, Ming-Wei Chang and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4171–4186.

Bruce W. Lee, Yoo Sung Jang, and Jason Lee. 2021. Pushing on text readability assessment: A transformer meets handcrafted linguistic features. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Yikuan Xie Masaki Uto and Maomi Ueno. 2020. Neural automated essay scoring incorporating handcrafted features. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6077–6088.

Zhaohui Zheng Mingrui Wu, Yi Chang and Hongyuan Zha. 2009. Smoothing dcg for learning to rank: A novel approach using smoothed hinge functions. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1923–1926.

Kian Ming A. Chai Peter Phandi and Hwee Tou Ng. 2015. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, page 431–439.

Zhiyuan Wen Youzheng Wu Ruosong Yang, Jiannong Cao and Xiaodong He. 2020. Enhancing automated essay scoring performance via fine-tuning pre-trained language models with combination of regression and ranking. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1560–1569.

Kapoor R Sharma A., Kabra A. 2021. Feature enhanced capsule networks for robust automatic essay scoring. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, arXiv:1503.06733:365–380.

Kaveh Taghipour and Hwee Tou Ng. 2016. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, page 1882–1891.

Lipika Dey Tirthankar Dasgupta, Abir Naskar and Rupsa Saha. 2018. Augmenting textual qualitative features in deep convolution recurrent neural network for automatic essay scoring. In *Proceedings of the 5th Workshop on Natural Language Processing Techniques for Educational Applications*, pages 93–102.

Ruiji Fu Lizhen Liu-Ting Liu Wei Song, Kai Zhang and Miaomiao Cheng. 2020. Multi-stage pre-training for automated chinese essay scoring. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, page 6723–6733.

Tie-Yan Liu Ming-Feng Tsai Zhe Cao, Tao Qin and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning*, pages 129–136.