

Automated Essay Scoring 2.0

Project Group - 15

Yu Tao



Introduction

Essay writing is an important method to evaluate student learning performance. It is also time-consuming for educators to grade manually.

A reliable essay scoring system can help educators save grading time and provide students with in-time feedbacks, making it highly worthy of research.

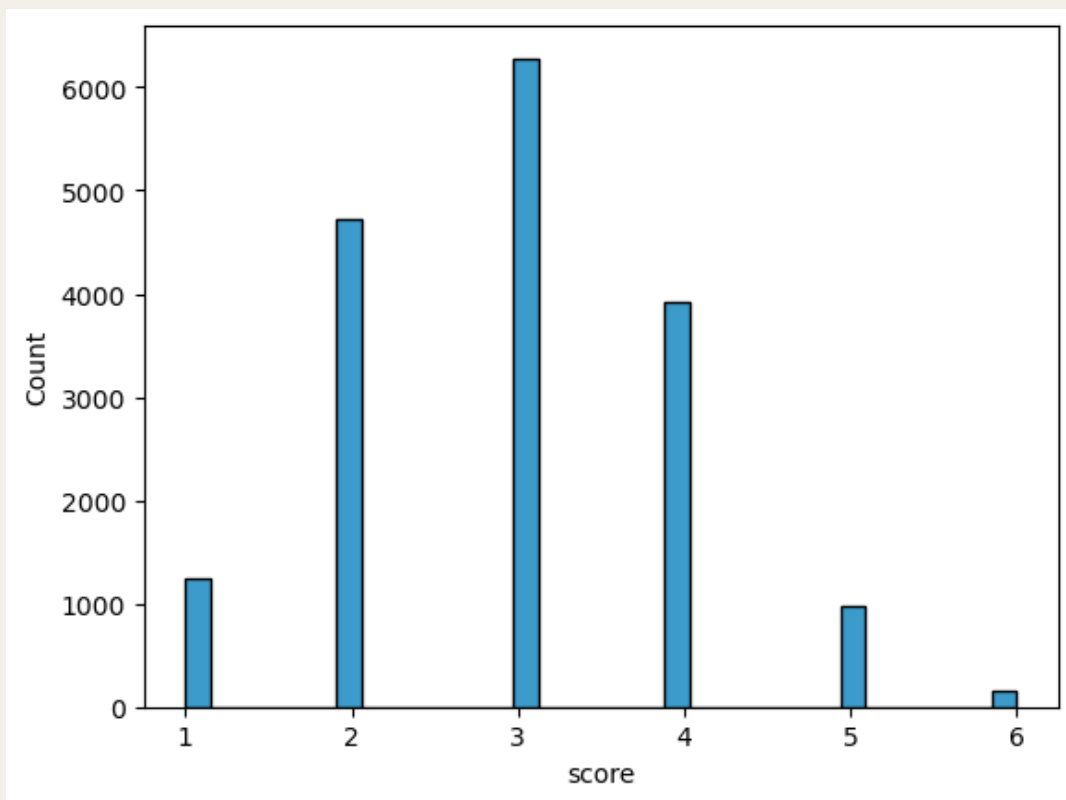
The goal of this project is to develop a model that assists the essay grading system.

Data

- This is a Kaggle Competition started in April 2024. Datasets can be directly downloaded from <https://www.kaggle.com/competitions/learning-agency-lab-automated-essay-scoring-2/overview>
- The dataset comprises about 24,000 student-written essays. 17,000 records are used for training and validating.
- Essays (text) and scores (int) are used as training data.
- Each essay was scored on a scale of 1 (minimum) to 6 (maximum).
- A score of 6 : critical thinking, solid support to its position, well organized, skillful use of language, free of grammar errors.
- A score of 1 : no evidence to support its position, disorganized, vocabulary and sentence structure problems.

Data

A closer look



Topic	Keywords
1	Venus, Planet, Earth
2	Cars, Driverless, Driver
3	Animals, Cowboys, Seagoing
4	Electoral, Vote, President
5	Students, Technology, Computer

Evaluation Method

Quadratic Weighted Kappa (QWK)

- A metric for ordinal problems because it considers the ordering of classes.
- Measures the **agreement** between two ratings (predictions v.s. ground truth).
- The value typically varies from 0 (random agreement) to 1 (complete agreement).

Sl. No	Range of Quadratic Weighted Kappa	Concordance
1	Negative	poor
2	0.01–0.20	slight
3	0.21–0.40	fair
4	0.41–0.60	moderate
5	0.61–0.80	substantial
6	0.81–1	almost perfect

Model – 1

LSTM

- Taking top 10,000 vocabulary from corpus.
- Truncating text to 500 words.
- Embedding tokens to 32 dimensions.

Layer (type)	Output Shape	Param #
embedding_11 (Embedding)	(None, 500, 32)	320000
lstm_11 (LSTM)	(None, 64)	24832
dense_11 (Dense)	(None, 6)	390
Total params: 345222 (1.32 MB)		
Trainable params: 345222 (1.32 MB)		
Non-trainable params: 0 (0.00 Byte)		

Model – 2

DistilBERT

- DistilBERT has only half of BERT layers.
- 60% faster than BERT during inference.
- 40% smaller than BERT.
- Retains 97% of BERT performance



Model – 2

DistilBERT

Customized DistilBERT model with one extra dense layer with ReLU activation.

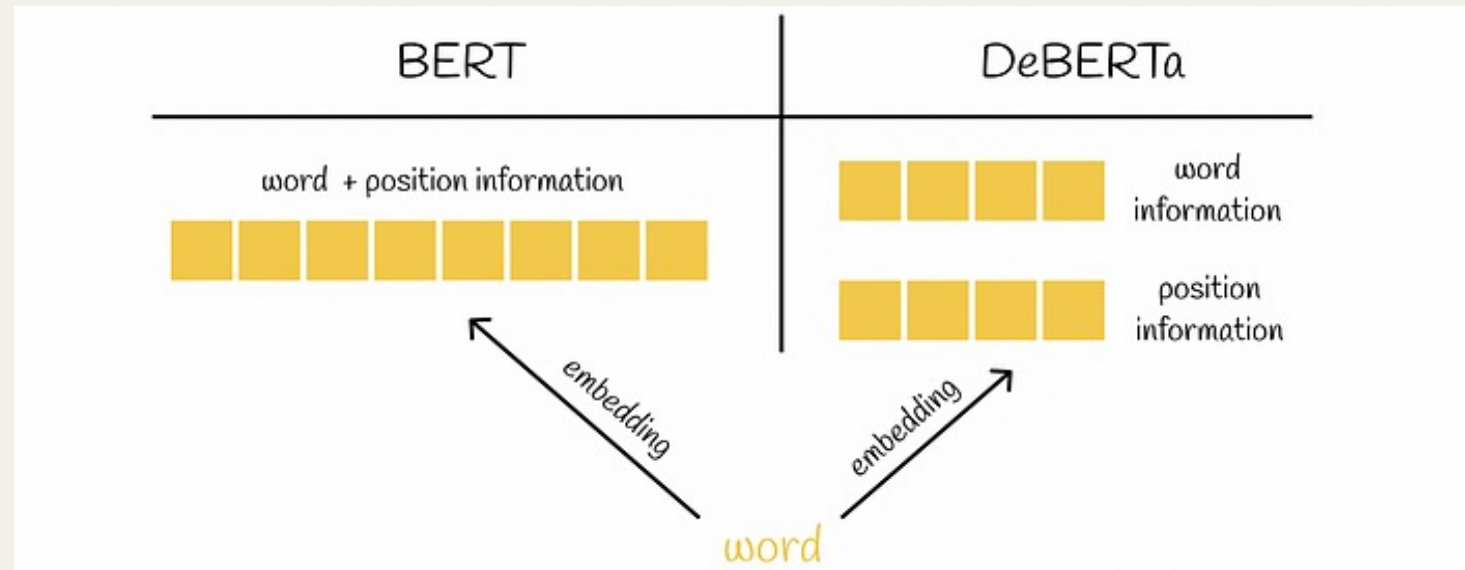
```
class DistillBERTClass(torch.nn.Module):
    def __init__(self):
        super(DistillBERTClass, self).__init__()
        self.l1 = DistilBertModel.from_pretrained("distilbert-base-uncased")
        self.pre_classifier = torch.nn.Linear(768, 768)
        self.dropout = torch.nn.Dropout(0.3)
        self.classifier = torch.nn.Linear(768, 6)

    def forward(self, input_ids, attention_mask):
        output_1 = self.l1(input_ids=input_ids, attention_mask=attention_mask)
        hidden_state = output_1[0]
        pooler = hidden_state[:, 0]
        pooler = self.pre_classifier(pooler)
        pooler = torch.nn.ReLU()(pooler)
        pooler = self.dropout(pooler)
        output = self.classifier(pooler)
        return output
```


Model -3

DeBERTa (Decoding-Enhanced BERT with Disentangled Attention)

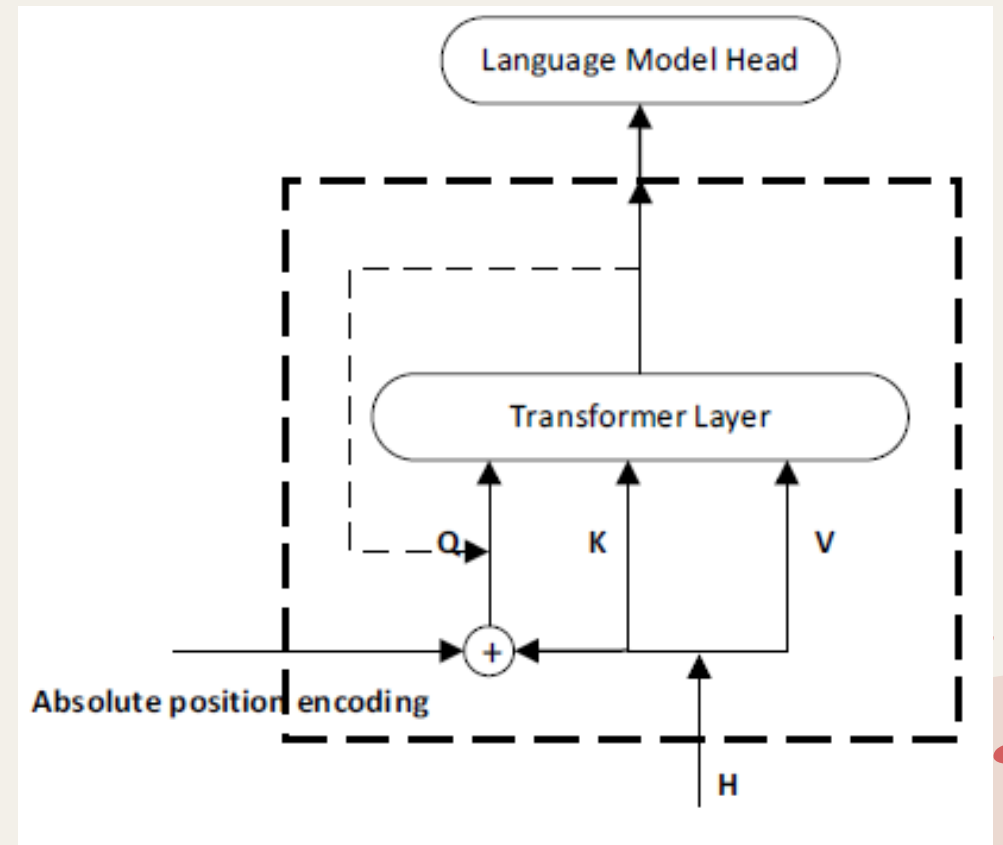
- In the original Transformer block, each token is represented by a single vector which contains information about word and position.
- The disadvantage of this approach is potential information loss: the model might not differentiate whether a word itself or its position gives more importance to a certain embedded vector component.



Model -3









DeBERTa (Decoding-Enhanced BERT with Disentangled Attention)

- Absolute position is incorporated in decoding layer.
- Given a sentence "a new store opened beside the new mall".
- Introducing absolute position enables the model to distinguish the store and the mall as two different instances.

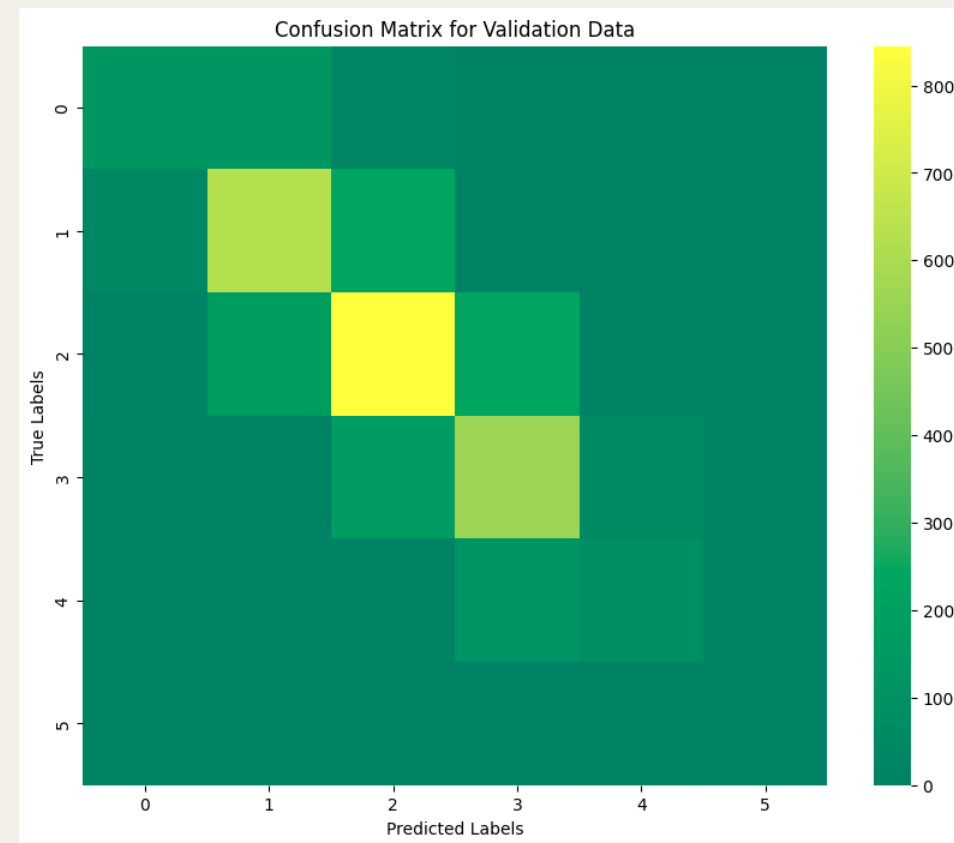


Modeling Results

Model	Train/Validation Loss	Validation QWK
LSTM	0.90/1.08	0.70
DistilBERT	0.76/1.02	0.73
DeBERTa	0.76/0.86	0.80

#	Team	Members	Score	Entries	Last	Join
1	Qihang Wang			0.819	30	2d
2	Jules			0.818	47	9h
3	Cling			0.818	73	4h
4	Aindriú			0.818	63	9h

Modeling Results



Result Comparison: DistilBERT (left) & DeBERTa (right)

Conclusions & Future Works

- The study indicates that using deep learning models for essay scoring is possible.
- Deep learning models are able to detect patterns that influence essay scores.
- Transformer-based models outperform RNN models.
- Due to the imbalance distribution between scores, high-scoring essays are difficult to predict, downsampling should be considered as a further step.
- Try the customized DeBERTa model to see if there is any improvement.

References

1. <https://www.kaggle.com/competitions/learning-agency-lab-automated-essay-scoring-2/overview>
2. https://www.researchgate.net/figure/Interpretation-of-quadratic-weighted-kappa_tbl1_336574571
3. <https://towardsdatascience.com/distilbert-11c8810d29fc>
4. <https://towardsdatascience.com/large-language-models-deberta-decoding-enhanced-bert-with-disentangled-attention-90016668db4b>