Neural Networks Project Report

Automated Essay Feedback

D.Nithish - IMT2014016
V. Keerthi Chandra - IMT2014064

---

1 Introduction

Our goal is to create a system that gives the essay feedback by identifying the stronger and weaker parts of the essay. This could help students improve their writing and prepare for standardized tests.

We are using a dataset of 1800 graded student essays provided by the Hewlett Foundation on Kaggle. The essays are between 150 and 550 words, written by 7th-10th grade students across 8 different topics. Approximately 1300 samples of the dataset have their scores among 6,7 and 8. This does not represent the class imbalance, rather it shows the extensive nature of the grading scheme for essays. This is a standard dataset for an essay grading problem but it only gives the final grade to an essay. It is challenging to turn this into a more grained feedback.

2 Pre-Processing

Essays have a minimum score of 0 and maximum score of 12. So we encode the essay scores into the one-hot vectors. We then process the essay in following steps.

1. Using spaCy (free, open-source library for advanced Natural Language Processing (NLP) in Python.) we parse each essay into sentences.
2. We then tokenize sentences into words. Then we build the vocabulary of obtained words.
3. We find the embeddings for each word in the vocabulary and then save it to the pickle file to avoid computational complexity on each run.
4. Finally we translate the words into the indexes. We added the 0 index to be the "UNK" unknown word.

3 Baseline Model

3.a Description

We used tensorflow to build our model. We used this baseline model to predict scores for the given essays rather than giving feedback. We tokenize the essay into words and then find the embeddings for each word. We then feed these embeddings to the multi layered LSTM. Then we feed the output of this LSTM layer to the fully connected layer. At last softmax layer is used to predict the score between 0 to 12(argmax of logit probabilities)

Since we don't have the annotations of stronger and weaker sentences for each essay we stick to the other metrics. Our error function is mean squared error between the predicted scores and the actual scores. We used the following parameters for executing different kinds of models to maintain consistency.

3.b Parameters and Choices

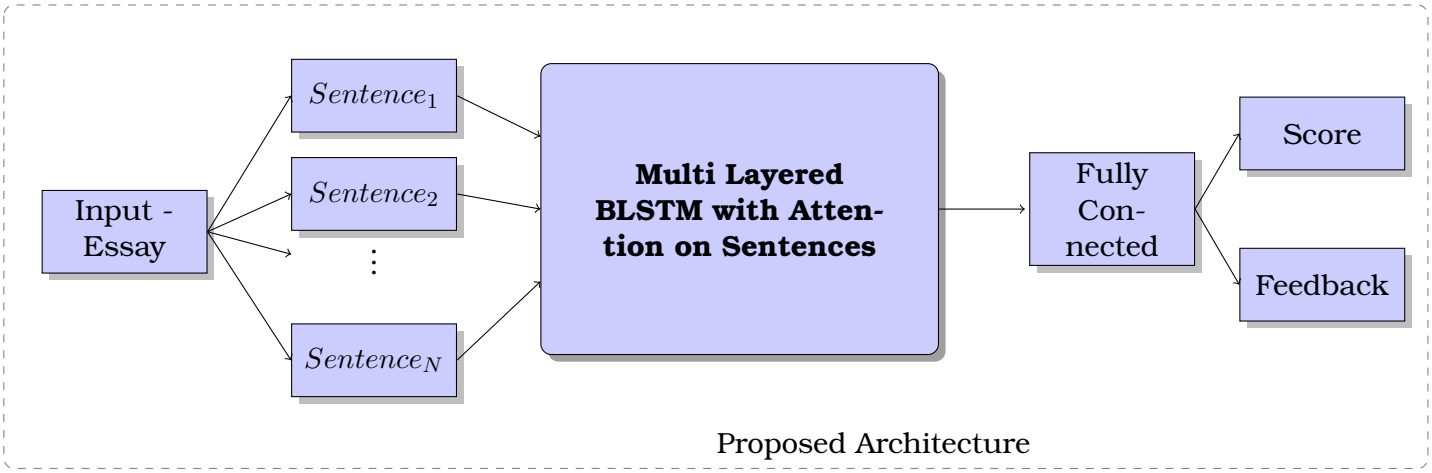
1. Hardware $\longrightarrow$ Intel AI Dev Cloud
2. Optimizer = Adam with learning rate $\longrightarrow$ 1e-4
3. batchsize $\longrightarrow$ 128
4. Number of layers $\longrightarrow$ 2
5. Hidden size $\longrightarrow$ 128
6. Maximum number of sentences in essay $\longrightarrow$ 20
7. Maximum number of words in a sentence $\longrightarrow$ 50
8. Embedding size $\longrightarrow$ 300
9. Number of epochs $\longrightarrow$ 30
10. Dropout $\longrightarrow$ 0.5

4 Models for Feedback Mechanism

We proposed two different viewpoints to give feedback to the essay. One being, we respond with stronger and weaker sentences of an essay. While the other, responds by spotting the best chunk of an essay.

4.a Finding the Strong/Weak Sentences

First we get the embeddings for each word using embedding matrix(built using spaCy). Then we feed them into the multi layered bidirectional LSTM. In order to capture both the forward and backward input we concatenate the first and last outputs of bidirectional LSTM to create sentence level encoding.



Proposed Architecture

We then use the attention mechanism to calculate the weights(probabilities) of each sentence in the essay. We interpret the weights of sentence as follows - High value corresponds to the stronger sentence and Low value to the weaker sentence. We output the stronger and weaker sentences using the weights that are assigned to sentences in an essay.

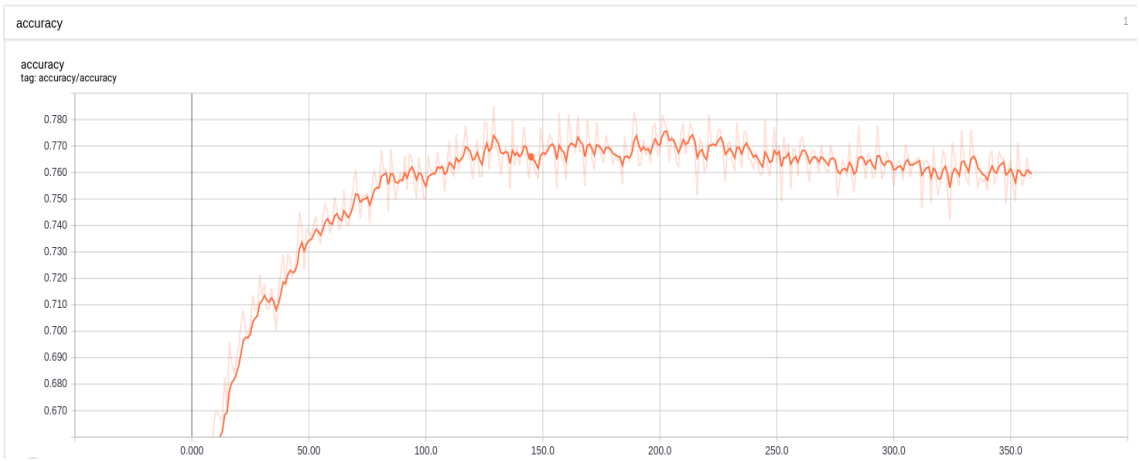


Figure 1: Training Accuracy

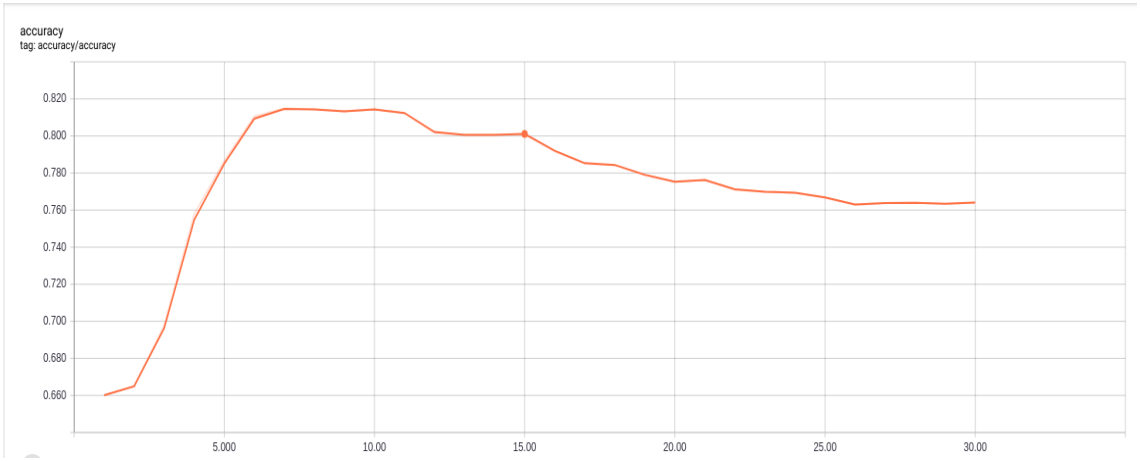


Figure 2: Testing Accuracy

4.b Identifying the Best chunk of an Essay

We split our essays into a small number of chunks. The size of each chunk, number of chunks to split are dependent on the size of an essay. We have restricted ourselves to split an essay to 2-3 chunks only. This limitation is needed during prediction, to create consistency in the input format of the essays. With the essay chunks in hand, we score each chunk of the essay. The scoring function remains same as that of our first feed back mechanism.

This approach had few problems bound with it. The essay sizes differ a lot higher than expected. So we ended up with highly variant chunk sizes. So to maintain consistency we have to pad these chunks or shorten them. So maintaining consistency in the input sizes of chunks is not ideal. To make it complete, we have to manually break these essays into required number of chunks and validate. To conclude, we don't have ground truth that decides the validity of our identified chunk.

5 Conclusion

Our perspective of identifying stronger and weaker sentences as a feedback mechanism is assuring. The promising behaviour is because of our utilization of attention model on the individual sentences of an essay. But our model is limited by the ground truth of the essays. This constraint reflected largely on the scoring mechanism. Our other feedback mechanism mainly

focuses on identifying a chunk of an essay. We consider this approach delicate, as it cannot accumulate various smaller sentences of an essay which are lodged in different chunks of an essay. Beyond that, we would want to explore with hyperparameter tuning like regularization, optimizer, learning rate etc.

References

[1] Data Set

[2] Code GitHub Repo

[3] Spacy NLP Library