Automatic Essay Grading and Feedback

Patrick Myers, Gaurav Kumar Jindal, Sanchit Sinha, Aniruddha Dave
Department of Computer Science
University of Virginia
Charlottesville, VA 22903

1 Problem Statement

Writing essays is a key component of many curricula across most academic subjects. Unlike other forms of evaluation, such as multiple choice tests, essays allow students to demonstrate their knowledge of a given subject in greater detail. However, unlike multiple choice tests, essays have no clear right or wrong answers and thus, they have traditionally been graded by people, wasting valuable time. Automatic essay grading is a Natural Language Processing (NLP) task whose goal is to automate the process of assigning scores to written essays. Advances in NLP over the years have allowed automatic essay grading to become more and more feasible and it is possible that such systems may eventually replace human graders in some courses.

One of the primary limitations that current automated essay grading systems face in comparison to human graders is their inability to provide any form of feedback beyond the scores that they have generated for an input essay. This is an issue for two reasons. Firstly, the lack of any feedback provided by such a system denies students the ability to learn from the mistakes they made on the essay so that they might earn a better score on their next written assignment. Secondly, having no feedback as to how a computer algorithm assigned an essay score could decrease both students' and teachers' trust in the legitimacy of such scores, decreasing the likelihood of fully adopting such a system. Although we may be far from being able to generate essay score feedback at the same level as a human grader, automated grading systems will need to be able to provide some level of specific feedback before most teachers will be willing to consider incorporating them into their classes. A few methods to draw inspiration from are [1] [2] [3] [4] [5] .

2 Proposed Method

We aim to address this issue by providing more specific feedback for automatically graded essays. To achieve this, we use an essay dataset - "The Hewlett Foundation: Automated Essay Scoring"[1] dataset. We used a subset of this dataset, set eight, which is labeled with scores across six different rubric categories: ideas/content, organization, voice, word choice, sentence fluency, and conventions.
First, we perform some basic pre-processing on each essay, including punctuation removal, stopword removal, and tokenization. This is done by the tokenizer and stopword functions from the NLTK

[1] `https://www.kaggle.com/c/asap-aes`

Preprint. Under review.

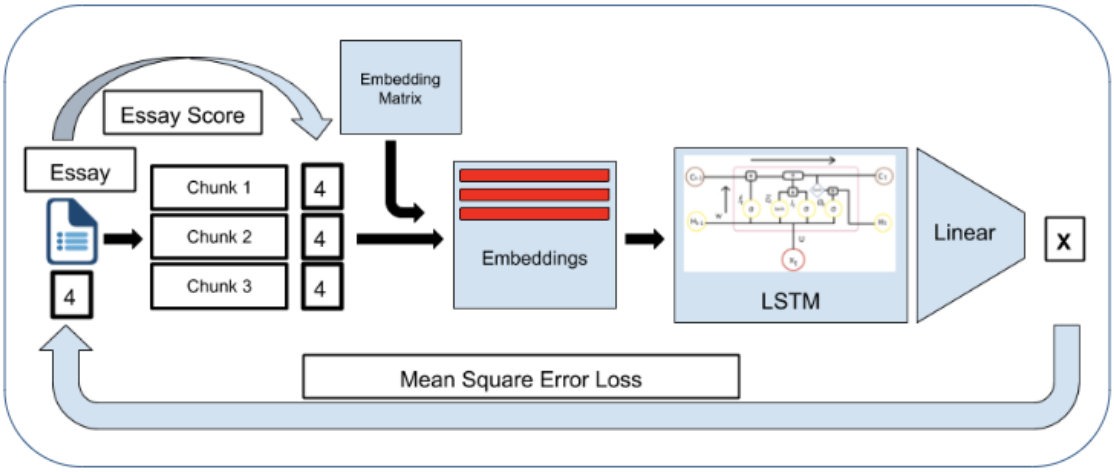


Figure 1: The proposed model architecture

library.

Next, we split each essay into three chunks of equal word length to represent the introduction, body, and conclusion of the essays. The method of chunking in our case is using the word counts - a third of words in each essay are kept in each chunk. The intuition behind the choice of three chunks is that - each essay can be roughly thought of as containing introduction, body and conclusion of roughly equal lengths.

Once the chunks are constructed, they are transformed to an embedding using an embedding matrix. Once the embeddig of each chunk is obtained, these chunk embeddings are each fed separately through an LSTM to obtain a feature vector. The feature vector is then passed through a fully connected layer and a final single score is obtained. The loss is computed using Mean Square Error with the original score, and the loss value is backpropogated through the entire network. Figure 1 gives a detailed schematic view of the complete architecture of the model.

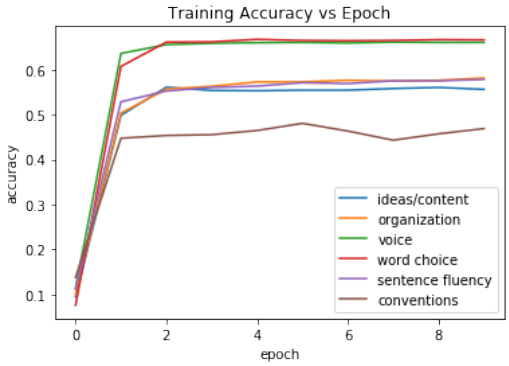
Although essays are labeled with discrete integer scores, the scores are ordered values so we treat this as a regression problem, using mean squared error as our loss function. We train separate models for each of the six rubric categories.

3 Expected Outcome

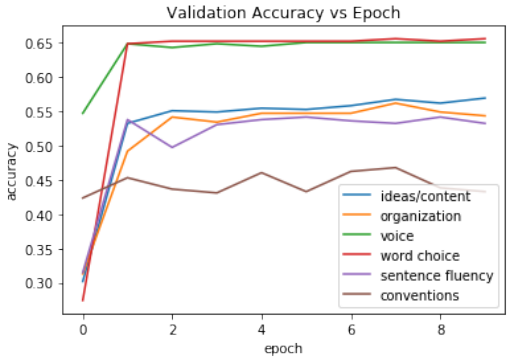
The final output of our system should be a set of six models, one for each of the six rubric categories. Furthermore, the system should be able to generate a matrix of scores for a given essay, presenting a score for each of the six categories across each of the three chunks. This output will enable students to better understand specifically where they struggled and succeeded in writing their essay as well as provide some intuition as to how their essay was graded.

Predicted/ Actual	Word Choice	Voice	Conventions
Chunk 1	3.757 / 4	4.016 / 4	3.450 / 3
Chunk 2	3.800 / 4	4.017 / 4	3.572 / 3
Chunk 3	3.892 / 4	3.970 / 4	3.562 / 3

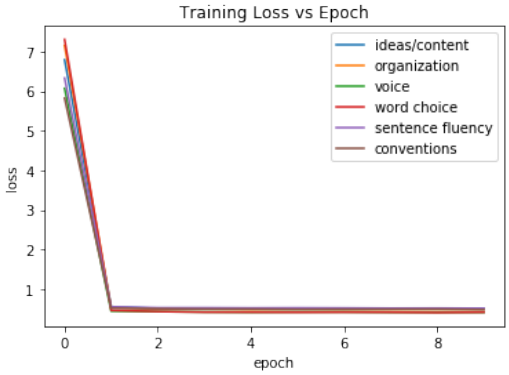
Predictions for a single essay (three of the six rubric categories) and their true values



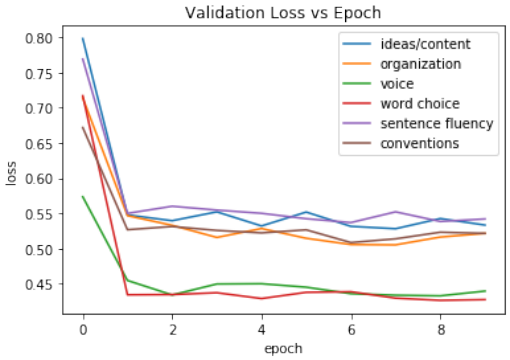
Training accuracy vs Epoch



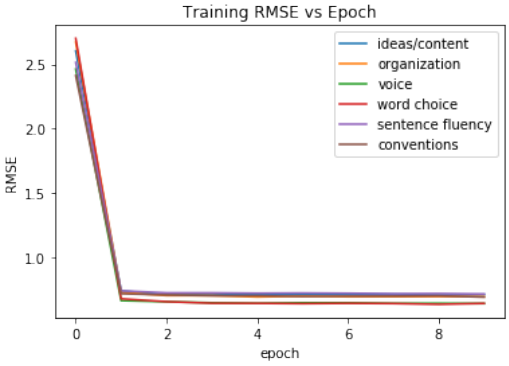
Validation accuracy vs Epoch



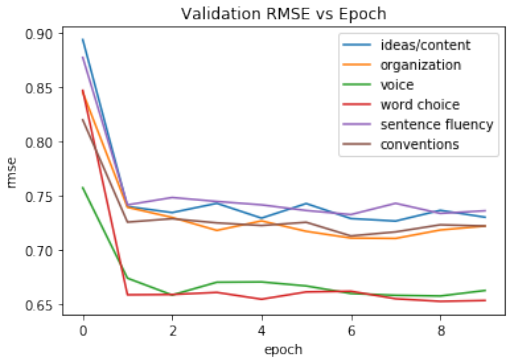
Training loss vs Epoch



Validation loss vs Epoch



Training RMSE vs Epoch



Validation RMSE vs Epoch

4 Results

The following hyper-parameter were used for the final result:

No.	Parameter	Value
1.	Embedding Size	100
2.	Hidden Units	32
3.	Output Dim	1
4.	Loss function	MSELoss
5.	Optimizer	Adam
6.	Learning Rate	0.0001
7.	Epochs	10
8.	Num. of Layers	2

Table 1: Model Hyper-parameters

As stated in our expected outcome, our code produces six total models, one for each rubric category. Together these models are capable of producing a grid of prediction scores, providing a score for each subsection of each rubric category for a given essay. We were also able to achieve reasonable values for loss and accuracy across our models. The training and validation loss decrease substantially within the first two epochs and then the rate of decrease of the loss decreases. Accuracy was measured by rounding predictions to the nearest integer and then comparing these values to the true score values of the labeled dataset. The training and validation accuracy show a similar behaviour to loss, whereupon they increase substantially in the first two epochs and then stagnate further.

The model performed best with traits such as word choice and voice, obtaining a validation accuracy of 65%. The model was able to score the essays correctly with an accuracy of nearly 55% on all other traits except conventions. Since the score on conventions depend on the punctuation, spellings and capitalization which are all removed from the data while preprocessing the accuracy score is below 50%, which is expected as a lot of information is lost during pre-processing.

5 Limitations and Future Work

Although we achieved much of what we planned to with this project, our work is limited in some ways. LSTM models requires a huge amount of training data to perform their best. However, we had a limited amount of data of around 700 essays. This means fewer essays for training which resulted in slight overfitting. If we could have obtained more training samples, we may have achieved higher accuracy on the validation set. Moreover, the dataset is slightly imbalanced towards middle score values which makes the models slightly biased. If we could have used a more diverse and balanced dataset, our model would have learned better parameters.

Future studies might build upon this by further subdividing essays into smaller chunks, possibly allowing for even more specific feedback scores so that students might better understand how to improve their essays. Additionaly, it would be interesting to try different types of models beyond LSTM and different preprocessing techniques to better understand if some perform better for different types of rubric scores. For example, our techniques did not work well for conventions, likely because we removed all stopwords, but removing these stopwords is necessary to perform well on other categories.

References

[1] Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. Automatic text scoring using neural networks. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2016.

[2] Noah Arthurs and Sawyer Birnbaum. Automated essay feedback.

[3] Ming Liu, Yi Li, Weiwei Xu, and Li Liu. Automated essay feedback generation and its impact on revision. *IEEE Trans. Learn. Technol.*, 10(4):502–513, October 2017.

[4] Abdulaziz Shehab, Mohamed Elhoseny, and Aboul Ella Hassanien. A hybrid scheme for automated essay grading based on lvq and nlp techniques. pages 65–70, 12 2016.

[5] Kaveh Taghipour and Hwee Tou Ng. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1882–1891, Austin, Texas, November 2016. Association for Computational Linguistics.