WHAT DOES A NETWORK-BASED AUTOMATED ESSAY SCORING SYSTEM KNOW ABOUT GOOD WRITING?

Nicole Hessler, Yihong Hu, and Evey Huang

Northwestern University
Computer Science Department
2233 Tech Drive
Evanston, IL

1. LEARNING PROBLEM

Essay writing has been a key part of the student assessment process in standardized exams such as the SAT and GRE. Millions of students across the world take such exams each year, resulting in a huge burden for human graders to grade such a high volume of writing efficiently and consistently. With recent advancement in neural networks and natural language processing, there is a possibility to scale up human graders' ability and reduce the amount of time needed to grade large numbers of essays in standardized tests and eliminate human bias caused by raters' expertise and inconsistency.

Previous research has built Automated Essay Scoring (AES) systems to automate the grading of essays. Most of the systems are based on regression models and rely heavily on a set of carefully designed and extracted features such as clarity and coherence (Kumar & Boulanger, 2020). But feature engineering is often challenging and time-consuming. Following advancements in neural nets, researchers have applied neural models to the problem of AES to learn features and relations between essays and scores automatically (Taghipour & Ng, 2016). Beyond the prediction of a holistic score, some researchers have explored ways to provide feedback on the rhetorical structures of students' writing (Fiacco et al., 2019).

In our project, we chose to replicate a neural model for automated essay scoring (Taghipour & Ng, 2016) with the Automated Student Assessment Prize (ASAP) dataset released on Kaggle[1], and test how the model performs on a professionally written essay and on computer-generated texts to see whether the model has learned to distinguish good writing.

2. INTELLECTUAL INTEREST

Although replicating the work of Taghipour and Ng is not new research, it provides us an opportunity to learn about lookup tables, convolution, LSTMs, and deep nets in general. These are all important topics in working with natural language and deep learning more broadly.

After replicating the aforementioned model, we tested its generalizability by running it on a professionally written essay and computer-generated texts. We then looked for explanations for the outputs to gain a better understanding of our model and deep nets overall. For future work, we plan to go beyond just predicting holistic scores but to generate actionable feedback that could help students learn how to improve their writing.

3. DATASET

In order to replicate the model built in the paper (Taghipour & Ng, 2016), we used the same training dataset the authors used: ASAP. This dataset contains eight sub-datasets of essays for eight different prompts. Due to time constraints, we only used prompt 1, which asked eighth grade students to write a letter to a local newspaper about their opinion on the effects computers have on people. The prompt 1 sub-dataset has 1,783 essays with an average length of 350. The longest essay is around 950 words. The essays are scored on a scale of 2–12.

After building the model and tested its performance on the original ASAP dataset, we decided to go one step further and see whether this model has learned to distinguish good and bad writing beyond the ASAP dataset. We first selected a professionally written essay to run on our model. We chose an essay from *The New Yorker* titled ”Are Computers Making Society More Unequal?” because of its relevance to prompt 1. As it is professionally written, it should receive a perfect score compared to eighth grade essays.

Additionally, we generated a list of random words using a random CSV generator[2]. This list of words deserves a low grade, as it has no meaning and is simply random words. Finally, we used the BABEL Essay Generator[3] to generate an on-topic essay (with keywords ”computer,” ”society,” and ”impact”) and an off-topic essay (with keywords ”pizza”,

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

[2] https://onlinerandomtools.com/generate-random-csv

[3] https://babel-generator.herokuapp.com/

”magic”, and ”trees”). While, the generated essays are of unclear quality compared to eighth graders, they certainly deserve grades greater than that of a list of random words and less than that of a professionally written essay, with the on-topic essay deserving a higher grade than the off-topic essay.

4. EXPERIMENT

Our system learns to predict a score given an essay. The input is a list of tuples, each of which, among other things, includes an `essay_id`, an `essay_set` (corresponding to an essay prompt), the essay itself, up to three raters' scores, and an aggregated score. Our system predicts the aggregated score given an essay. In the ASAP dataset, this ground truth label is called `domain1_score`, and the essay is called `essay`.

To evaluate the models generalized performance, we use the model output score, which is a value normalized from 0 to 1. We compare these results to draw conclusions about the model's ability to find good writing.

4.1. Architecture

This project revolves around reproducing the most successful architecture proposed by Taghipour and Ng: a network that contains a lookup table, an optional convolution layer, a recurrent layer with LSTM units, and finally a fully connected layer with a sigmoid activation. The lookup table takes a sequence of words represented by one-hot encoding. A convolutional layer can be applied to the output of the lookup table to extract local features before the recurrent layer. Its objective is to improve the system with n-gram level information and contextual dependencies. The recurrent layer takes the embedded output of either the lookup table or the convolution layer and creates a representation of the essay, which is passed through a mean-over-time layer. Finally, the linear layer maps the representation to a scalar score value.

In order to make meaningful comparisons, our system uses the same hyper-parameters as the Taghipour model. We use the RMSProp optimizer to minimize the mean squared error (MSE) with decay rate (ρ) of 0.9 and learning rate of 0.001. We also use dropout regularization and gradient clipping to avoid over-fitting and exploding gradient respectively.

We used 5-fold cross validation to evaluate our systems. In order to make our results comparable to the experiment of Taghipour and Ng, we divided the dataset in the same fashion, where 60% of the data is used for the training set, 20% for the development set, 20% for the testing set. A model is trained for each fold for 50 epochs and evaluated on the corresponding development set. The best performing model is selected to perform the generalization task.

4.2. Evaluation Metric

To replicate the experiment, we used the same evaluation metric used by Taghipour and Ng, the quadratic weighted kappa (QWK), which was adopted by Kaggle for the ASAP competition as the official metric. QWK is used as a measure of agreement between two graders. A QWK of 1 corresponds to complete agreement, around 0 indicating total chance, and -1 to indicate complete disagreement. For our experiments, we calculated the QWK between human graders and the scores predicted by the model. The closer to 1 QWK indicates a higher accuracy for the model.

The QWK is calculated as follows. First we generate a weight matrix $\mathbf{W}_0$, where i and j are scores given by human graders and our system respectively, and N is the number of possible ratings.

$$\mathbf{W}_{i,j} = \frac{(i-j)^2}{(N-1)^2} \tag{1}$$

Another matrix $\mathbf{O}$ is generated where $\mathbf{O}_{i,j}$ represents the number of essays given score i by a human grader and j by our model. We also need an expected matrix $\mathbf{E}$ that is the outer product of histogram vectors of the two scores. We need to normalize $\mathbf{E}$ so that the sum of elements in $\mathbf{E}$ is the same as the sum of the elements in $\mathbf{O}$. Finally, the QWK score is calculated with $\mathbf{W}$, $\mathbf{E}$ and $\mathbf{O}$ as follows.

$$\kappa = 1 - \frac{\sum_{i,j} \mathbf{W}_{i,j}\mathbf{O}_{i,j}}{\sum_{i,j} \mathbf{W}_{i,j}\mathbf{E}_{i,j}} \tag{2}$$

For the generalization part of this project, since we do not have a ground-truth score, we are unable to calculate a useful QWK. Instead, we use the model output score (value normalized from 0 to 1) and conduct pairwise student-t significance test to see if the model is able to discern the difference in quality among the selected texts.

4.3. Baseline

For the experiment replication part of the project, we use the performance of the Enhanced AI Scoring Engine (EASE) as our baseline approach. This model is the best open-source system that participated in the ASAP competition. It relies on manually crafted features and regression methods to assign a score to an essay. We use the EASE results reported by Taghipour and Ng as our baseline metric.

Our generalization task does not involve a comparison to prior work, so we do not have a baseline approach for this part of the project. Instead we perform pairwise comparisons among the selected texts.

5. RESULTS

In this section, we present the results and our findings for our replicated study as well as the generalization study.

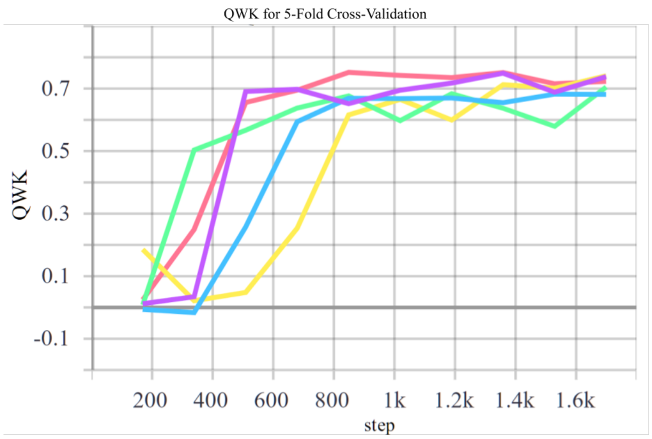

Fig. 1. Change in QWK during training for 5-fold validation

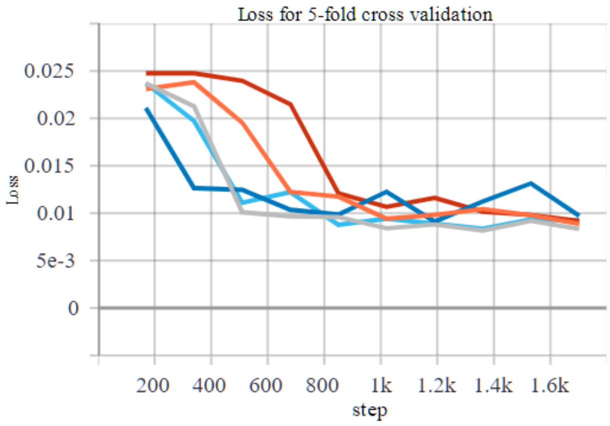

Fig. 2. Change in loss during training for 5-fold validation

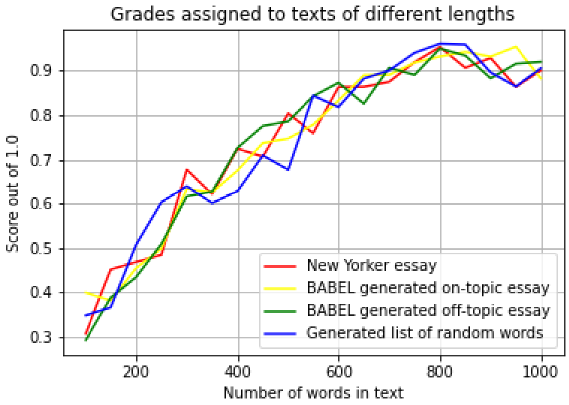

Fig. 3. Score variations by number of words in essay. Scores normalized to range [0,1]

	T statistic	p-value
List of words, BABEL on-topic	-0.2714	0.3946
List of words, BABEL off-topic	-0.1883	0.4264
List of words, *New Yorker*	-0.1074	0.4578
BABEL on-topic, BABEL off-topic	0.0535	0.4789
BABEL on-topic, *New Yorker*	0.1468	0.4425
BABEL off-topic, *New Yorker*	0.0987	0.4612

Table 1. T statistics and p-values for all pairwise combinations of externally sourced essays.

Figure 1 shows the change in validation QWK during training for each of the five folds. Our best performing model obtained a QWK of 0.770. This is 0.049 lower than the 0.821 prompt 1 QWK obtained from Taghipour and Ng's LSTM with CNN approach, but our system is able to outperform the baseline approach by 0.009 on prompt 1.

While our system achieves similar results to those presented by Taghipour and Ng, we fail to conclude that it has learned what makes a good English essay. We graded increasingly long subsections of externally sourced essays, and the scores are shown in Figure 3. We should expect the professionally written *New Yorker* essay to perform better than the on-topic generated essay, which should perform better than the off-topic generated essay, which should perform better than the generated list of random words. However, the essays' scores are more clearly determined by their length than their substance. We conducted a one-sided t-test on each pair of essays. The results are shown in Table 1. None of the differences between essays is significant, indicating that the system is unable to discern the difference in quality between a collection of unrelated words, a syntactically correct essay, and an overall coherent essay.

6. CONCLUSIONS AND FUTURE WORK

Though we were able to replicate the model, train it with the same tuning parameters and number of steps, and reach a similar QWK score to the one obtained by the authors, our model did not necessarily learn about what constitutes a good essay. Compared to other AES approaches with heavy feature engineering such as coherence, clarity, or grammar (Kumar & Boulanger, 2020), our approach of directly feeding in the tokenized text to the model resulted in a model that evaluates essays as merely lists of words. The model did not learn about what makes good writing, but only memorized the differences in vocabularies between the good and bad essays within the prompt 1 essays from ASAP dataset.

We have a few important future work directions to investigate: (1) A more thorough evaluation of our model: besides the length of an essay, what else is contributing to the score? How would it perform on a larger set of unseen data? (2) How might we preserve the meaning, coherence, and overall structure of an essay when training a neural net model? How might a model really learn to recognize good writing? (3) How might we make our model and its predictions more transparent and explainable? A main challenge working with neural models is that they are ”black boxes.” We could not understand or analyze what contributed to the model’s prediction easily. (4) Beyond generating a single score, an AES system would be much more beneficial if it could provide learners actionable feedback on how they could improve their writing.

7. RELATED PAPERS

[1] Fiacco, J., Cotos, E., & Rosé, C. (2019). Towards enabling feedback on rhetorical structure with neural sequence models. In ACM International Conference Proceeding Series. https://doi.org/10.1145/3303772.3303808

[2] Taghipour, K., & Ng, H. T. (2016). A neural approach to automated essay scoring. In EMNLP 2016 - Conference on Empirical Methods in Natural Language Processing, Proceedings. https://doi.org/10.18653/v1/d16-1193

[3] Kumar, V.S., Boulanger, D. Automated Essay Scoring and the Deep Learning Black Box: How Are Rubric Scores Determined?. Int J Artif Intell Educ (2020). https://doi.org/10.1007/s40593-020-00211-5