Automated Essay Grading Using Deep Learning

Yao Huang, Yuhui Tang, Xuechun Wang, Xinyu Zhang
Georgetown University

Abstract

Automated Essay Grading contributes to give writers a fairer grade and more efficient feedback. In this project, we use Deep Learning to solve the problem and provide advanced solutions based on previous work. The modeling procedure is divided into four stages. In the final stage, we apply LSTM with Glove embedding to grade hand-written essays and achieved a final accuracy (evaluated by average Kappa score) of 0.9765.

Credits

This paper is a written version of the final project for the class "Deep Learning and Neural Network" at Georgetown University in fall 2019. The results received help from Professor Keegan Hines along the way.

1. Introduction

Essay is an analytic or interpretative literary composition which is used to convey a limited or personal point of view with regard to a domain specific topic. It is a common and important form of communication for the purpose of education to evaluate the perspectives and learning outcomes of students. However, it is commonly regarded that essay grading is not only time-consuming but also subjective. This leads to long time waiting for feedbacks for students and potential unfair assessment based on graders' preference of word choice, perspective and values.

For the difficulty of being objective in grading an essay, Automated Essay Scoring has been an active area of research in recent years. The application of the research is very likely to be extended beyond education and apply to more diverse areas such as search engines and question answering systems.

As the prosperous development of machine learning and neural networks in natural language processing, there is diverse ways to approach this problem. In this project, we will apply Natural Language Processing tools on the essay data and build Neural Network models to achieve auto essay grading. The goal is to build an automatic scoring engine which gave each essay a grade. The accuracy is assessed based on the closeness of the score generated by the model and by the human expert graders.

We start form the baseline model and develop add-ons on the baseline mode to improve the model performance. In addition, we apply basic deep Neural Network, Recurrent Neural Network (LSTM), Convolutional Neural Network, along with feature engineering method, Neural Bag of Word, Word2vec, Glove word embedding. We successfully achieved a final Kappa score of 0.9765 compared to our baseline model with a Kappa score of 0.7833.

2. Related Works

Before applying deep model in the text scoring system, previous work treated text scoring as a supervised text classification task. Predictive features of the system need to be manually engineered by human experts. There is substantial manual effort involved in reaching these results on different domains and genres, as the linguistic features are hand-selected and tuned for specific domains. In order to reach out good performance on different kind of text, separate models with distinct features should be tuned.

Project Essay Grade is one of the earliest auto-essay grading system which uses linear regression on

vectors of textual features. Intelligent essay assessor uses latent semantic analysis to calculate the semantic similarity between text on a specific score point on test text, and give the test text a score based on the score of the most similar text on training set. Lonsdale and Strong-Krause (2003) use the Link Grammer Parser to analyze and calculate the score of text based on parser's cost vector on average sentence level.

In 2012, Kaggle is sponsored by Hewlett Foundation hosted Automated Student Assessment Prize (ASAP) contest which aims to find fast, effective and affordable solutions for automated grading of student-written essays. In this contest, Dimitrios Alikaniotis, Helen Yannakoudakis and Marek Rei from University of Cambridge proposed the use of recurrent neural networks which can automatically learn useful features from data without the need of manual tuning. They produced model which is constructed by score-specific word embedding (SSWE) plus two-layer BLSTM, reached best result among all the baseline models with 0.96 of Cohen's kappa. Another group of competitors Kaveh Taghipour and Hwee Tou Ng from National University of Singapore used LSTM to get the best system which outperforms the baseline model by 5.6% in terms of quadratic weighted kappa. Apart from automatic essay grading, recurrent neural networks have also been used for opinion mining (Irsoy and Cardie, 2014), sequence labeling (Ma and Hovy, 2016), language modeling (Kim et al., 2016; Sundermeyer et al., 2015), etc.

3. Data Source

The graded essays are from Kaggle and are selected according to the specific data characteristics. On average, each essay is approximately 150 to 550 words in length. There are eight essay sets in total, all essays were written by students ranging in grade levels from Grade 7 to Grade 10. All essays were hand graded and were double scored. Each of the eight data sets has its own unique characteristics, distinct marking criteria and score range. The variability is intended to test the limits of scoring engine's capabilities. Each data point has essay ID, essay set ID, ascii text of essay, score of different raters on different domains.

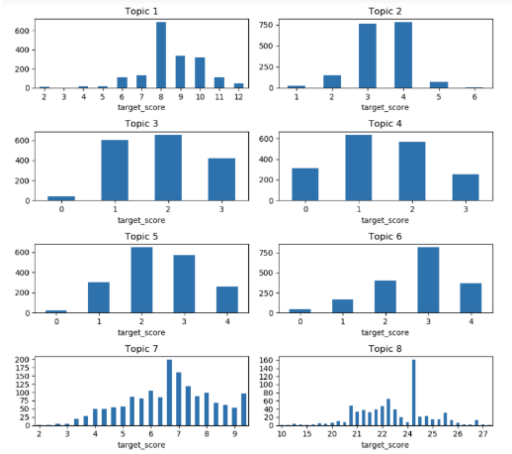

Figure 1: Histogram of target score and word count

4. Experiment & Result

The experiments are conducted in four stages. For stage 1 and 2 (our baseline model), we apply Neural Network model on eight topics separately and one combined topics. To build the model for stage 1 and 2, we first generate vectorized features from processed essays and try to measure similarity using a similarity metric available from package 'SpaCy'.

Stage 3 and 4 are our advanced models, where we apply Neural Bag of Words and LSTM with Glove embedding. It is expected that we will achieve better model performance in the stage 3 and 4.

4.1 Stage 1 (Baseline): Neural Network on Separate Topic

For the baseline model, we first train a neural network model on separate topic with feature engineering by adding the new features to the dataset

Layer (type)	Output Shape	Param #
dense_19 (Dense)	(None, 14)	4634
dropout_10 (Dropout)	(None, 14)	0
dense_20 (Dense)	(None, 1)	15

Total params: 4,649
Trainable params: 4,649
Non-trainable params: 0

Table 1: Stage 1 model summary

.

The kappa score on test data, with added new features to the model, achieved a combined Kappa

score 0.573 and a weighted mean Kappa score of 0.7833

4.2 Stage 2: Neural Network on Combined Topic

In addition to train a Neural Network within each topic group, we also build a model which combine all topics into a single model. In this case, as each topic has a different range of scores, we need to scale the essay scores to a common min-max range.

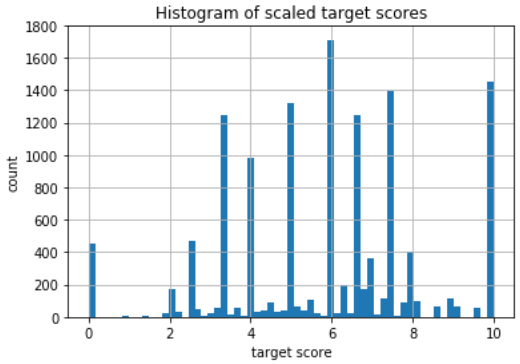

Figure 2: Scaled Target Score

A new data frame with the original scores is created and passed through the train, test, split step. This step is crucial to ensure proper re-scaling of target scores in cases where either the highest or lowest scores were eliminated in the split step.

Some previous studies apply the Kappa metric directly to the complete set of essays. However, due to the differences in scale, essay topics (sets) that have a narrower scoring range will end up with significantly smaller weighted distances and thus artificially higher Kappa scores. The Kappa metrics shows a scores of up to 94.5%

Layer (type)	Output Shape	Param #
dense_37 (Dense)	(None, 14)	4634
dropout_19 (Dropout)	(None, 14)	0
dense_38 (Dense)	(None, 1)	15

Total params: 4,649
Trainable params: 4,649
Non-trainable params: 0

Table 2: Stage 2 Model Summary

The combined essay topics kappa score is much higher than that obtained from individual topics, however, as noted above, this is deceptive. After we did some recalculation, we have a combined essay kappa score of 0.9774 and a Weighted by topic Kappa score: 68.60%.

The following plots show our model results.

Figure 3: Difference of predicted score and actual score

4.3 Stage 3: Neural Bag of Words

As demonstrated in Figure 3, Neural Bag of Words (NBOW) has the bag-of-words assumption which is similar as linear models, which makes predictions using the sums the weights for each input word. Each word in the essay is treated as an input and altogether predict the final score of the essay.

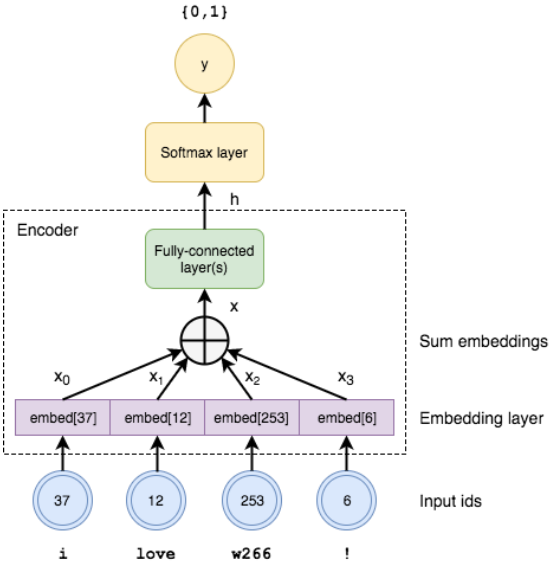

Figure 4: NBOW explanation

In NBOW model, we tokenized the essays and prompts either padded or truncated the list of tokens based on a decided maximum essay and prompt length. The maximum length of the essay was set at 650, as more than 95% of the essay data was shorter than this length. We chose to set the maximum prompt length at 130, as all the longer prompts which are at more than 800 words have very long reference passages before reaching the actual essay question at the end of the prompt. We also chose to truncate the essays and prompts differently, as we expect the most relevant content in an essay located

up front whereas the most important content in a prompt is at the end with the essay question. Hence, essays which exceed the maximum essay length are truncated from the end and prompts which exceed the maximum prompt lengths are truncated at the beginning.

The results of the NBOW show that Quadratic Weighted Kappa (QWK) score for dev and test are 0.7111 and 0.7265 respectively.

4.4 Stage 4: LSTM with Glove embedding

By adding glove embedding layer to LSTM, the performance is very impressive. In this stage, we are using un-weighted kappa score to measure the performance. The average Kappa score after 5-fold cross validation is 0.9765.

It's also interesting that, the loss was high at the beginning, however, it goes down from the second epoch and keeps dropping at a stable rate. That shows our model is learning at a stable rate.

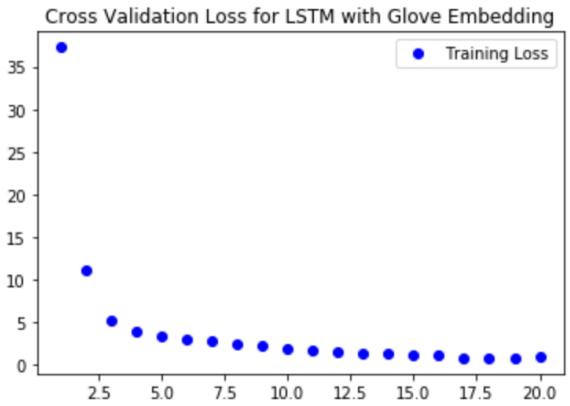

Figure 5: Loss for LSTM with Glove Embedding

5 Conclusion

In our stage 2, 3 and 4, our advanced model outperforms the baseline model of 0.573. So our general goal is achieved. However, several further steps could be implemented.

In terms of the Natural Language Processing, we can utilize pre-trained models such as ELMo and BERT word embedding, which can generate different word embedding for a word that captures the context of a word - that is its position in a sentence. We can also try to implement Grammar and Spelling correction by checking and correcting the grammar and spelling errors in the essay. Then the number of errors could be used as a new feature in the model and would provide better input for the NLP processing (word embedding specifically). Some of the tools to use: python package and language tools. Drawbacks: processing speed. In terms of Neural Network, we could search for a more intensive hyper-parameter on deep LSTM to further improve the model.

References

ASAP AES dataset https://www.kaggle.com/c/asap-aes/data

Deryle Lonsdale and D. Strong-Krause. 2003. Automated rating of ESL essays. In Proceedings of the HLT-NAACL 2003 Workshop: Building Educational Applications Using Natural Language Processing.

Ellis B.. Grading essays by computer: progress report. Page. 1967 In Proceedings of the Invitational Conference on Testing Problems

Farag, Y., Yannakoudakis, H., Briscoe, T. (2018). Neural Automated Essay Scoring and Coherence Modeling for Adversarially Crafted Input. NAACL 2018: Human Language Technologies, Volume 1.

Kaveh Taghipour and Hwee Tou Ng 2016. A Neural Approach to Automated Essay Scoring

Kim, Seon-Wu, and Sung-Pil Choi. "Research on Joint Models for Korean Word Spacing and POS (Part-Of-Speech) Tagging Based on Bidirectional LSTM-CRF." Journal of KIISE, vol. 45, no. 8, 2018, pp. 792–800

Kuangwei Huang, Martin Jung; Automated Essay Scoring with Attention and BERT; UC Berkeley, School of Information

Peilu Wang,Yao Qian, Frank K. Soong, Lei He and Hai Zhao 2015. Automatic Text Scoring Using Neural Networks.