Automatic Essay Grading Using Neural Network

Author: Yanchen Wang, Xintong Zhao, Zhen Zhong

Instructor: Joshuah Touyz

1 Abstract

Automatic Essay Scoring (AES) is an effective way to reduce cost and it provides more consistent grading results than human grader. However, designing an auto-scoring method that is close enough to human grader requires a solid background in mathematics and statistics.

In this project, we introduce two auto-scoring models. One model applies natural language processing (NLP) method to select features and uses neural network to predict the essay score. The other model uses each essay as an input and uses recurrent and convolutional neural network to predict scores.

2 Introduction and Problem Statement

The Automatic Essay Scoring (AES) integrates the knowledge of Natural Language Processing and Statistical Analysis and automatically returns a score for the input text.

To achieve a similar performance to human grader, one major challenge is feature selection. Good features tend to increase the performance of the model. To select features, In the first model, we apply Latent Dirichlet Allocation (LDA) topic modeling, N-gram analysis to extract the main content of input text, then we feed our features to both recurrent and artificial neural network. In the second model, we use essays as input and use neural networks to automatically select features from essays and predict scores.

3 Related Work

Automated essay scoring (AES) has a long history. Educational Testing Service (ETS) offered e-rater starting in 1999 (Burstein, 2003). In 2012, the Hewlett Foundation sponsored a competition on Kaggle called the Automated Student Assessment Prize. During the competition, there were many models invented to accurately grade essays. Many models use hand-crafted features in their machine learning model. However, those hand-crafted features promotes teaching to test that students learn to write essays to meet features in AES. In our model, we use two approaches. First, we use essay responses as direct inputs and use recurrent and convolutional neural network to automatically learn features from essays to predict score of the essay. Second, we create some features from essays such as length, number of unique words and relative between essay and its prompt.

4 Description of Data Source

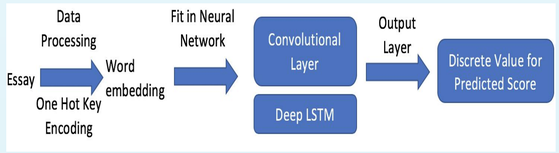
We use dataset from Automated Student Assessment Prize sponsored by the Hewlett Foundation from Kaggle. In this dataset, there are six essay sets. Here is some statistics of the dataset:

Essay Set	Type	Score Range	Average Length	Number of Essays in the set
1	Persuasive/Narrative/Expository	2-12	350 Words	1,785
2	Persuasive/Narrative/Expository	2-12	350 Words	1,800
3	Source Dependent Responses	0-6	150 Words	1,726
4	Source Dependent Responses	0-6	150 Words	1,772
5	Source Dependent Responses	0-8	150 Words	1,805
6	Source Dependent Responses	0-8	150 Words	1,800

5 Description of Models

We develop two models using neural network. The first model uses recurrent neural network and convolutional neural network using the whole essay to predict its score. The second model selects features from each model and use those features as predictors to predict its score. In the second model, we use artificial neural network and recurrent neural network to do the prediction.

5.1 Recurrent and Convolutional Neural Network Architecture



Data Processing: First, we remove all punctuations and stop words such as “a”, “the”, “I”, etc. from each essay. Then we create an index number for each unique word among all essays and a vector to represent each essay. In each vector, each word in that essay would be represented as a number. We generate a matrix by putting all essays’ vectors together that each row represents an essay. We also do one hot key encoding on essay scores to transform categorical features into a one hot numeric array.

Word Embedding: In this part, we create a word embedding to project each word into a 64 dimensional space.

Neural Network Fitting: After generating word embedding, the convolutional layer starts to process the input into a representation. Then this representation will be fed into recurrent layer. In the recurrent layer, we choose long short-term memory (LSTM) units.

Output: In this layer, we use softmax function as activation function. In this model, we take the problem as a classification problem and we choose categorical cross entropy as the loss function. At the end, this model outputs discrete values for classes.

5.2 Feature Selection

In one word, the keypoint of feature selection in our case is how to accurately extract the main content and writing pattern of input text data. After removing stop words and stemming process, we firstly apply Latent Dirichlet Allocation method to extract the keywords related to the prompt, then use unigram, bigram and trigram analysis to search potential writing patterns that could exist in given document, and compute their frequencies.

Other than N-gram analysis and LDA topic modeling, we also select length of the essay, number of sentences, average sentence length, word length, unique word length and number of unique words, all predictors include values before and after removing stopwords. Most importantly, we create a feature that measures the “relativeness” to the prompt, shown as “number of synonyms/antonyms”.

For the main content of the essay prompt, we use corpus from NLTK to find all words that are related to the main content. Then, while processing the input text, we check if each word is related to the essay prompt and count the frequency of synonyms and antonyms.

After generating all features above, we implement artificial neural network and recurrent neural network using all features extracted from each essay. We created a vector in 71 dimensional space that contains all features from each essay.

We build a regression model using artificial and recurrent neural network with mean squared error as loss function. Outputs from the two neural networks are continuous values and we round the result into integers and use them as score for each essay.

5.3 Regularization and Tuning

In all neural networks we build, we use early stopping as our regularization method.

Early stopping helps us prevent overfitting in neural networks. In Tuning process, we choose different learning rates and picked the optimal one.

6 Analysis of Results

Models with essay as direct input:

Model Type	Essay Set/Exact Accuracy						Essay Set/± 1 point Accuracy					
	1	2	3	4	5	6	1	2	3	4	5	6
RNN	0.63	0.59	0.58	0.56	0.49	0.49	**0.91**	**0.93**	**0.88**	**0.85**	**0.86**	**0.88**
CNN	0.52	0.51	0.52	0.54	0.52	0.47	0.87	0.86	0.87	0.83	0.84	0.84
RNN + CNN	0.56	0.54	0.56	0.58	0.51	0.50	0.87	0.87	0.85	0.82	0.85	0.84

Models with features

Model Type	Essay Set/± 1 point Accuracy					
	1	2	3	4	5	6
ANN	0.92	0.89	0.88	0.86	0.88	0.86
RNN	0.93	0.92	0.89	0.88	0.92	0.87

7 Reference

[1] Burstein "The E-rater(R) Scoring Engine: Automated Essay Scoring with Natural Language Processing", p. 113.

[2] “The Hewlett Foundation: Automated Essay Scoring.” RSNA Pneumonia Detection Challenge Kaggle, www.kaggle.com/c/asap-aes#description. fyf3BuL4/edit?ts=5c048094#slide=id.g462a66394e_2_79

[3] Bird, Steven, Edward Loper and Ewan Klein (2009), Natural Language Processing with Python. O’Reilly Media Inc.