Automated Essay Grading Using Deep Learning

Yao Huang, Yuhui Tang, Xuechun Wang, Xinyu Zhang

Georgetown University

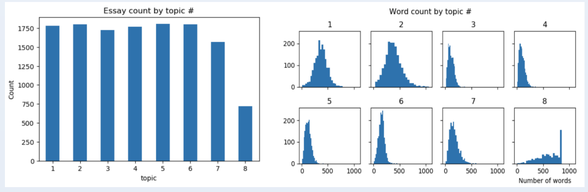
INTRODUCTION

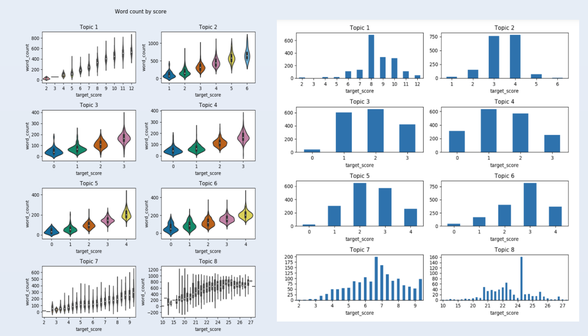
Essay is an analytic literary composition which is used to convey a limited or personal point of view on a domain specific topic. It is a common and important form of communication for the purpose of education to evaluate the perspectives and learning outcomes of students. However, it is commonly regarded that essay grading is not only time-consuming but also subjective. This leads to long time wait 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.

In this project, we apply Natural Language Processing method on the essay data and build Neural Network models to achieve auto essay grading. We start from the baseline model and develop add-ons on the baseline model to improve the model performance. 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 embeddings.

DATA

The graded essays dataset comes from Kaggle and is 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 limit 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.



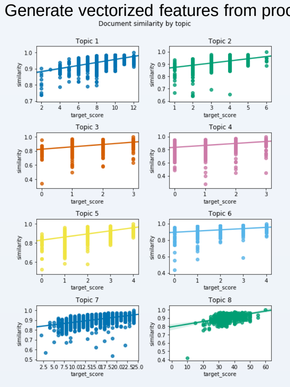


Reasonable correlation between word count and score for all but topic 8 where the word count apparently reaches a maximum at the upper third of the scores.

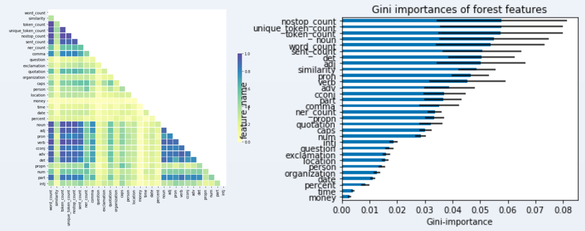
Many scores are underrepresented. Classification could be difficult without rebalancing.

METHOD

Method 1: Feature engineering (with add-on features)
Generate vectorized features from processed essays and measure



Length-based features: word_count, token count, unique_token_count, nostop_count, sent_count
Part-of-speech fearues: comma, question, exclamation, quotation, organization, caps, person, location, money, time, data, percent, noun, adj, pron, verb, cconj, adv, det, propn, num, intj
Other features: similarity, ner_count



Method 2: Neural Bag of Words

Using the following notation:

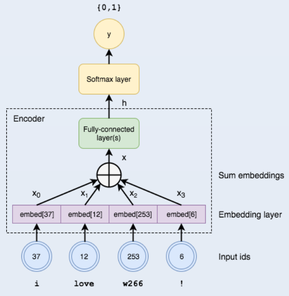
- $w^{(i)} \in \mathbb{Z}$ for the i^{th} word of the sequence (as an integer index)
- $x^{(i)} \in \mathbb{R}^d$ for the vector representation (embedding) of $w^{(i)}$
- $x \in \mathbb{R}^d$ for the fixed-length vector given by summing all the $x^{(i)}$ for an example
- $h^{(j)}$ for the hidden state after the j^{th} fully-connected layer
- y for the target label ($\in 1, \ldots,$ `num-classes`)

Our model is defined as:

- **Embedding layer:** $x^{(i)} = W_{embed}[w^{(i)}]$
- **Summing vectors:** $x = \sum_{i=1}^{n} x^{(i)}$
- **Hidden layer(s):** $h^{(j)} = f(h^{(j-1)}W^{(j)} + b^{(j)})$ where $h^{(-1)} = x$ and $j = 0, 1, \ldots, J-1$
- **Output layer:** $\hat{y} = \hat{P}(y) = \text{softmax}(h^{(final)}W_{out} + b_{out})$ where $h^{(final)} = h^{(J-1)}$ is the output of the last hidden layer.

Logits for the softmax is defined as:

$$\text{logits} = h^{(final)}W_{out} + b_{out}$$



For our 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.

Method 3: LSTM with Glove embedding
GloVe is an unsupervised learning algorithm for obtaining vector representations for words. The network we built here includes:

- embedding layer with pretrained GloVe.
- a LSTM layer with 64 nodes
- a dense output layer

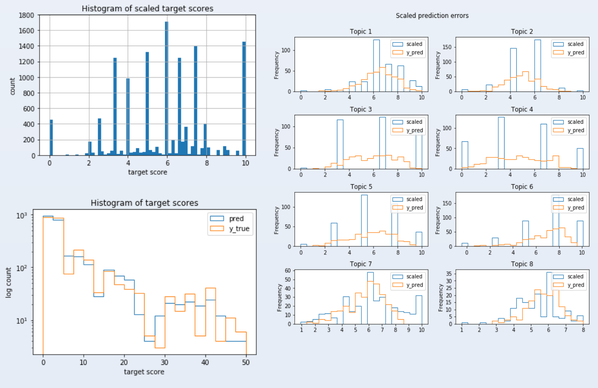
The key difference between LSTM using GloVe embedding with simple LSTM network is that the embedding layer can be seeded with the GloVe word embedding weights. We chose the 100-dimensional version, therefore the Embedding layer must be defined with output_dim set to 100. Using pre-trained GloVe will benefit in the following ways:

- The goal of Glove is very straightforward, i.e., to enforce the word vectors to capture sub-linear relationships in the vector space. Thus, it proves to perform better than Word2vec in the word analogy tasks.
- Glove adds some more practical meaning into word vectors by considering the relationships between word pair and word pair rather than word and word.
- Glove gives lower weight for highly frequent word pairs so as to prevent the meaningless stop words like "the", "an" will not dominate the training progress.

RESULTS

Stage 1: Train neural network on separate topic
Trained on essay body with add on features (with feature engineering)
The kappa score (on test data) improved dramatically after adding on new features to the model. The combined Kappa score is 0.7833

Stage 2: Train neural network on combined topic
Fit each topic into a single model. As each topic has different range of scores, essay scores should be scaled to a common min-max range.
We train on the new scaled data with the original scores is created and passed through the train, test, split step. It is necessary to ensure proper re-scaling of target scores in cases where either the highest or lowest scores were eliminated in the split step.



The combined essay topics kappa score is much higher than that obtained from individual topics, however, as noted above, this is deceptive, as shown in a recalculation below: **Combined essay kappa score: 0.9774**
Weighted by topic Kappa score: 68.60%

Stage 3: Neural Bag of Words
The results of the NBOW show that Quadratic Weighted Kappa (QWK) score for dev and test are 0.7111 and 0.7265 respectively.

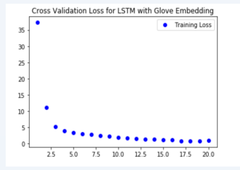
Based on running on 7 Apr:

```
Neural BOW baseline for MSE on dev set: 3.83%
Neural BOW baseline for QWK on dev set: 0.7111

Neural BOW baseline for MSE on test set: 3.53%
Neural BOW baseline for QWK on test set: 0.7265
```

Stage 4: LSTM with Glove embedding

In this stage, we are using unweighted Kappa score to measure the performance. **The Average Kappa Score after a 5-fold cross validation is 0.9765.** By adding glove embedding layer to LSTM, the performance is very impressive. It's noticed that the loss was high at the beginning but goes down from the second epoch and keeps dropping at a stable rate. That shows our model is learning at a stable rate.



APPLICATION

Spell Checking
Researchers developed a new system for detecting misspelled words, in which some actual corrections performed by the typist provides the basis for error detection. These corrections are used to train a feed-forward neural network so that if the same error is remade, the network can flag the offending word as a possible error.
Named Entity Recognition(NER)
NER classify named entities (e.g. Microsoft, London) into predefined categories like persons, organizations etc. Many NER systems were already created, and the best of them use neural networks.
Part-of-Speech Tagging
It is the process of marking up a word in a text to a particular part of speech. The model in *Part-of-Speech Tagging with Bidirectional LSTM* was tested on the Wall Street Journal data from Penn Treebank III data set and achieved a performance of 97.40% tagging accuracy.
Semantic Parsing and Question Answering
Question Answering systems automatically answer different types of questions asked in natural languages. Neural networks usage makes it possible to develop higher performance. The paper Semantic Parsing via Staged Query Graph Generation Question Answering with Knowledge Base applies an advanced entity linking system and a deep convolutional neural network model which was tested on Web Questions dataset outperforms previous methods substantially.
Machine Translation
Machine translation software Is another aspect of application. A neural network requires fewer resources for training and maintenance, and also performs better than traditional models.

CONCLUSION

In our stage 2, 3 and 4, our advanced model significantly outperforms the baseline model (Kappa of 0.573) . So we achieved the goal to improve the Auto Essay Grading model. 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 embeddings, which can generate different word embeddings 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). But the drawback of this method is the slow processing speed. For further investigation on Neural Network techniques, we can search for a more intensive hyperparameter on deep LSTM to improve the model.

REFERENCES

[Ellis B. Page. 1967]. Grading essays by computer: progress report. In Proceedings of the Invitational Conference on Testing Problems
[Lonsdale and Strong-Krause 2003] 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.
[Wang et al.2015] Peilu Wang,Yao Qian, Frank K. Soong, Lei He and Hai Zhao 2015. Automatic Text Scoring Using Neural Networks.
[Taghipour and Ng.2016] Kaveh Taghipour and Hwee Tou Ng 2016. A Neural Approach to Automated Essay Scoring