Automated Essay Scoring: My Way or the Highway!

Alexander Hurtado, Vamsi Saladi
Department of Computer Science
Stanford University
Stanford, CA 94305
hurtado@stanford.edu, vamsi99@stanford.edu
https://github.com/alexanderjhurtado/nlp_aes

Abstract

The following research attempts to approach the problem of automated essay scoring, a long-standing goal in the world of natural language processing. We approached this problem using deep learning techniques, rather than more commonly used machine learning techniques like bag-of-words logistic regression or support vector machines. We implemented and trained our own single-layer unidirectional LSTM network, multi-layer unidirectional LSTM network, word-level recurrent highway network, and word-to-sentence-level recurrent highway network. We used a dataset provided by the platform Kaggle, which hosted a competition sponsored by the Hewlett Foundation, the creator of the dataset. We allocated essay scores into one of four buckets (0,1,2,3) to account for different grading schemes. After training our models on virtual machines, we found that the multi-layer unidirectional LSTM outperformed the rest of the models, producing an accuracy of approximately 0.63. However, the recurrent highway network and the single layer unidirectional LSTM both did relatively well as well, with accuracies around 0.54 and 0.55 respectively.

1 Introduction

Automating the process of essay scoring has been a long-standing wish in the world of NLP. As a natural venue of research in the world of natural language processing, automated essay scoring became a hot topic for research as the popularity of sentiment analysis increased. Research began on automated essay scoring as early as 1999, with the development of the CRASE automated constructed response grader developed by Howard Mitzel and Sue Lottridge as a part of Pacific Metrics. However, the research did not really take off in academia until 2012, when Kaggle released a dataset provided by the Hewlett Foundation with over 13,000 transcribed essays and teacher criticism and ratings.

Our goal was to use deep learning methods to address the problem, and build a model by training on approximately 13,000 essays with their respective scores. There are 8 essays prompts, and take a respective proportion of each prompt to train, validate and test on. We wanted to compare our results to the baselines established before us using machine learning techniques like regression, generalized linear models, and Support Vector Machines. However, we wanted to use these baselines but improve upon them by pursuing deep learning techniques. Using techniques like LSTMs, RNNs, and highway networks, we wanted to see if we could improve upon the performance of non-network based models on automated essay scoring.

Our proposed contribution to this area of research is to infuse deep learning techniques, which are criminally underused as of today. We hope to report the accuracy scores that result from using the methods described above.

32nd Conference on Neural Information Processing Systems (NIPS 2018), Montréal, Canada.

2 Related Works

There has been a significant amount of research done into the area of automated essay scoring, varying from the use of machine learning techniques that do not involve neural networks to those that depend entirely on them. One of the earliest papers written on this topic deals with using logistic regression and SVMs on the essay representations to get a decision boundary, effectively treating as multi-class classification problems.

Next, we see research developed as more machine learning techniques began to be applied to this area. We see that paper by Taghipour and Ng [4] which was one of the earlier papers written dealing with automated essay scoring to consider using the idea of convolutions. This idea spread to other researchers as by Farag, Yannakoudakis, and Briscoe [2] shows, which used an convolution layer of windows to get convolutions to pass into the main RNN.

We also drew inspiration from highway networks and their increased use in NLP tasks. Specifically, we drew from the work of Zilly et. al. (2017) [1] on the idea of using recurrent highway networks, a model structured to combine the ease-of-training of highway networks and infuse it into the classic vanilla recurrent neural network architecture. The intuition behind this comes from the idea that recurrent highway networks can learn complex structures in the data because, like multi-layer LSTMs, they are structurally deep along the vertical axis. Unlike multi-layer LSTMs, however, recurrent highway networks are computationally easier to train (as seen by our results) because of their use of highway networks. By replacing LSTM cells with highway networks and restructuring the step-to-step transitions, a recurrent highway network is less prone to the issue of vanishing and exploding gradients than LSTMs and other deep recurrent models.

3 Dataset

The dataset we used was provided by the Hewlett Foundation as part of the Automated Student Assessment Prize (ASAP) contest, hosted by the computer science platform Kaggle. The essays are responses from students between grades 7 to grade 10. All essays were hand-written and double-scored, and are later transcribed onto a word document for our purposes.

We were given 8 different datasets of essays, each of which were essays in response to a different prompt. However, the dataset provided was structured in a way such that for proper nouns like names of people or organizations, the names themselves are replaced with placeholders like "@ORGANIZATION1" or "@NAME3." Thus, in order to deal with this, we decided to replace these tokens with the corresponding word vector for the noun it represents. For example, "@NAME2" would be replaced with the word vector for "name," and "@ORGANIZATION3" would replaced with the word vector for "organization." We felt this was the easiest way to not completely lose the word and maintain some semblance of the meaning. Additionally, there are some tags that cannot just be converted directly to a lowercase word, like "@CAPS" or "@NUM", which we handle manually. So, for @NUM, we changed the word to "number". For @CAPS and @DR, we changed it to "name," because they are used to replace the names of proper nouns and doctors respectively. Additionally, we changed the suffix "Dr." to "doctor" so we could also represent this properly.

Furthermore, we had to account for the different rubrics and different grading scales used for each essay. Each dataset had its own scale range, so we consolidated all the essays into a one large dataset, and used histograms to assign 4 possible scores: $\{0, 1, 2, 3\}$. Thus based on the scoring scale, and some frequency analysis, we were able to model the overall score distributions among essays using this method.

4 Methods

In order to fairly compare the effectiveness of our deep neural models, we need to observe how well a simpler machine learning model performs on the task.

We first obtained a baseline score using a single-layer multinomial logistic regression model. In order to input our data into the logistic regression model, we constructed a vector representation for each essay using a bag-of-words approach. This approach views essay representation a vector whose dimension is equivalent to the size of the vocabulary; each entry in the vector corresponds to

the frequency of a certain word from the vocabulary in the given essay. These bag-of-words vectors are then passed through our model, which outputs the classification probabilities for essay. These probabilities are then used to predict the score class of the essay. We then use cross entropy loss as the loss function to backpropagate prediction error.

We then moved onto deep recurrent neural models. In particular, we approached the task through two primary architectures: long short-term memory (LSTM) models and recurrent highway networks (RHN). In each recurrent model, we first convert each word in a given essay to their corresponding Global Vector (GloVe) representation. These word embeddings will then serve as input to our models.

The first recurrent model that we utilized was a vanilla single-layer, unidirectional LSTM. For this LSTM model, we convert each essay into a sequence of GloVe embeddings corresponding to the sequence of words in the essay. We then pass each word embedding into the LSTM until the entire sequence is processed. We then determine a vector representation for the essay by taking the sum of the hidden states output by each LSTM cell. The resulting essay vector is then passed into a fully connected layer. This fully connected layer determines a mapping from the essay vector to output classification probabilities for the essay. The score class of the essay is predicted using these probabilities and the prediction error is back propagated using cross entropy loss.

The second recurrent model we constructed was a multi-layer, unidirectional LSTM. This model works nearly identically to the vanilla single-layer LSTM described above. However, a multi-layer LSTM is structurally different in that it is made deep in the vertical axis by applying stacking multiple LSTMs on top of each other. By stacking LSTMs, our network can compute more complex representations, with the idea that the lower-level LSTMs compute lower-level features while the higher-level LSTMs compute complex, higher-level features. This multi-layer architecture is advantageous to task of essay score as it allows for greater model complexity, allowing our model to better understand the complex reasons that dictate the score of an essay.

The third recurrent model implemented was a recurrent highway network, as described in the work of Zilly et. al. (2017). A recurrent highway network is similar in structure to a multi-layer LSTM; both are recurrent neural models that are deep in both the time dimension and in the vertical dimension. However, RHN models are fundamentally different from multi-layer LSTMs, architecturally. Whereas a multi-layer LSTM has a step-to-step transition where an input is processed through a single LSTM cell before being passed off to the next layer and the next cell, the step-to-step transition of a recurrent highway network is defined by processing the input through L stacked highway layers before being passed off to the next input. Similarly to how LSTMs can be described as a recurrent sequence of LSTM cells, an RHN model can be best described as a sequence of recurrent highway stacks, where each stack is constructed by L stacked highway layers.

Let's define m as the dimension of the GloVe word embeddings, n as the dimension of the hidden layers, L as the depth of the step-to-step transition (highway stack), $\mathbf{x} \in \mathbb{R}^m$ as the input to the highway stack, and $\mathbf{y} \in \mathbb{R}^n$ as the output of the highway stack. We can then also define $\mathbf{W_{H,T,C}} \in \mathbb{R}^{n \times m}$ as the input weight matrix (with bias) and $\mathbf{R_{H_\ell,T_\ell,C_\ell}} \in \mathbb{R}^{n \times n}$ as the recurrent weight matrix (with bias). Let $\mathbf{s}_\ell$ denote the output at depth ℓ with $\mathbf{s_0}^{[t]} = \mathbf{y}^{[t-1]}$. Then, we can formally describe the computation done by a single highway layer in a stack at depth $\ell \in \{1, 2, \ldots, L\}$ as follows:

$$\mathbf{s}_\ell^{[\mathbf{t}]} = \mathbf{h}_\ell^{[\mathbf{t}]} \cdot \mathbf{t}_\ell^{[\mathbf{t}]} + \mathbf{s}_{\ell-\mathbf{1}}^{[\mathbf{t}]} \cdot \mathbf{c}_\ell^{[\mathbf{t}]},$$

where

$$\begin{aligned}
\mathbf{h}_\ell^{[\mathbf{t}]} &= \tanh(\mathbf{W_H x}^{[\mathbf{t}]} \mathbb{I}_{\{\ell=\mathbf{1}\}} + \mathbf{R_{H_\ell}} \mathbf{s}_{\ell-\mathbf{1}}^{[\mathbf{t}]}) \\
\mathbf{t}_\ell^{[\mathbf{t}]} &= \quad \sigma(\mathbf{W_T x}^{[\mathbf{t}]} \mathbb{I}_{\{\ell=\mathbf{1}\}} + \mathbf{R_{T_\ell}} \mathbf{s}_{\ell-\mathbf{1}}^{[\mathbf{t}]}) \\
\mathbf{c}_\ell^{[\mathbf{t}]} &= \quad \sigma(\mathbf{W_C x}^{[\mathbf{t}]} \mathbb{I}_{\{\ell=\mathbf{1}\}} + \mathbf{R_{C_\ell}} \mathbf{s}_{\ell-\mathbf{1}}^{[\mathbf{t}]})
\end{aligned}$$

and $\mathbb{I}_{\{\}}$ is the indicator function.

Similarly to how the output for an LSTM was handled, we take the sum of the outputs $\mathbf{y}^{[\mathbf{t}]} = \mathbf{s}_\mathbf{L}^{[\mathbf{t}]}$ of each highway stack and pass the resulting vector into a fully connected layer. This fully connected layer then produces the classification probabilities for the essay, which is then used to predict the essay's score. The loss function used to backpropagate error is cross entropy loss.

The final recurrent model implemented comprised of passing in the output of a word-level RHN into a sentence-level RHN. In particular, an essay would first be broken up into its constituent sentences. Then, the word embeddings for each word in a sentence would be passed into an RHN; the output vector of this RHN is used as a form of sentence representation. These sentence representation vectors are then passed into another RHN that processes the sentence vectors to output a final essay representation in an identical manner to the RHN model described above. This final representation is similarly passed into a fully connected layer to get classification probabilities that are then used to predict the essay's score/class. The overall structure of this word-to-sentence-level RHN is similar to how character-level and word-level models interact in a hybrid NMT model.

5 Experiments / Results / Discussion

5.1 Experimental Details

The model parameters for our Adam-optimized gradient descent we used were as follows:

- Evaluation method: Accuracy
- Learning rate: 0.0001
- Max epoch: 15
- Dimension size of embeddings: 300
- Layer/stack depth: 3

The optimal learning rate hyperparameter was found by iteratively running the RHN model on different learning rates until convergence or a maximum of 3 epochs. The learning rate hyperparameter search is summarized below.

Table 1: Performance of RHN model vs Learning Rates

Learning Rate	Accuracy (%)
1×10^{-2}	0.2158
1×10^{-3}	0.2991
1×10^{-4}	0.4003
1×10^{-5}	0.3170
1×10^{-6}	0.3092

5.2 Results

The following table highlights the results of our work.

Table 2: Performance of the Models

Baseline Models		
Model	Training Time (hrs)	Accuracy (%)
Logistic Regression	0.56	0.452
Single-Layer LSTM	7.43	0.540
Deep Models		
Model	Training Time (hrs)	Accuracy (%)
Multi-Layered LSTM	20.39	0.631
Recurrent Highway Network (Word-level)	16.39	0.543
Recurrent Highway Network (Word-to-sentence)	16.68	0.548

The multi-layered LSTM model and the single-layered LSTM models were the ones that performed the best on our dataset, which relatively high accuracy scores ($\sim 60\%$) scores compared to random

guessing (25% chance). However, the recurrent highway network models trains much faster than the LSTM models. Overall, it was surprising the model that best balanced training time and accuracy was the single-layered LSTM.

5.3 Error Analysis / Discussion

Given the scores above, there is definitely room for improvement in our model accuracy. There were certain things about the grading scheme that prevented the model from being as effective as it could be. For example, consider the following essay:

> Dear local newspaper I raed ur argument on the computers and I think they are a positive effect on people. The first reson I think they are a good effect is because you can do so much with them like if you live in mane and ur cuzin lives in califan you and him could have a wed chat. The second thing you could do is look up news any were in the world you could be stuck on a plane and it would be vary boring when you can take but ur computer and go on ur computer at work and start doing work. When you said it takes away from exirsis well some people use the computer for that too to chart how fast they run or how meny miles they want and sometimes what they eat. The thrid reson is some peolpe jobs are on the computers or making computers for exmple when you made this artical you didnt use a type writer you used a computer and printed it out if we didnt have computers it would make ur @CAPS1 a lot harder. Thank you for reading and whe you are thinking adout it agen pleas consiter my thrie resons.

The correct score for this essay was 2 on our scale from 0 to 3. However, the predicted score was 0. We can see by reading the essay itself that the arguments made are not actually terrible for the age of the students writing them; the likely reason they lost a point was due to misspelling. Words that are misspelled do not contribute useful information to our model since they are typically represented by the unknown word token. As a result, misspellings contribute negatively towards the essay's score; thus, our model is not very effective at dealing with well-written essays riddled with misspellings.

However, the model did handle length differences relatively well. For an example essay on the shorter end of the range of 150-550 words that scored a 2 on the 0-3 scale, all three models predicted a 2 as well. This demonstrates that essay length, though important in determining quality, was handled appropriately by the sequential nature of our models.

6 Conclusion

Overall, the models did relatively well in getting close to the correct score, even if it didn't exactly match the score. In this sense, accuracy was not an ideal evaluation metric. Ultimately, the multi-layered LSTM performed best on this task while both RHN models and the single-layer LSTM performed relatively similarly. Our RHN model was successful in that it achieved around the same accuracy as a single-layer LSTM while being around 5% faster.

We could improve upon these models by implementing bidirectionality. This would increase the complexity of our model and could improve its accuracy. Additionally, it would be interesting to consider using character embeddings to create representations for words that are replaced by the unknown token. This could dramatically affect the performance of our model by addressing the issue of misspelled words. There was also a lot of research done towards the potential use of convolutions and using convolutional layers in tandem with highway networks. The idea of using convolutional layers to capture the information of 2-grams or 3-grams in the essay would be worth exploring in the future

Additionally, our use of the recurrent model where the outputs of word-level RHN are passed into a sentence level RHN was perhaps a bit off the mark. It is possible that the highway cells in the second part of this model did not actually learn anything new out of what passed in, because there is not much more to learn in going from words to sentences and then to essays, rather than words to essays. This might be what caused the negligible difference in accuracy between the two (a difference of 0.005%). However, the models did much better than random guessing and improved noticeably on our baseline models.

Contributions

Both team members were responsible for different aspects of the project. Alex mostly focused on fully implementing the four deep learning models that were compared in the report and was tasked with debugging them to ensure that they ran locally without trouble. Vamsi focused on handling the data, building the proper pre-processing methods, and building the non-deep baseline models. He also focused on debugging the virtual environment so the models could be run on a GPU. Both members split the work on the final report and on the project poster.

Acknowledgments

We would like to thank our professors Andrew Ng and Kian Katanforoosh for helping us acquire the tools necessary to conduct this research. Additionally, we would like to thank our mentor Weini Yu for her help and advice in developing our models and working through the obstacles we faced while conducting this research.

References

[1]J. Zilly, R. Srivastva, J. Koutník and J. Schmidhuber, "Recurrent Highway Networks", Arxiv.org, 2019.

[2]Y. Farag, H. Yannakoudakis and T. Briscoe, "Neural Automated Essay Scoring and Coherence Modeling for Adversarially Crafted Input", 2017. [Online]. Available: https://arxiv.org/pdf/1804.06898.pdf. [Accessed: 02-Mar- 2019].

[3] Pennington, Jeffrey, Richard Socher, and Christopher D. Manning. "Glove: Global Vectors for Word Representation." EMNLP. Vol. 14. 2014.

[4]K. Taghipour and H. Ng, "A Neural Approach to Automated Essay Scoring", 2016.

[5]Valenti, S., Neri, F., Cucchiarelli, A. (2003). An Overview of Current Research on Automated Essay Grading