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.

However, 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 Approach

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.

First, we realize that this a classification task, and thus we have to use algorithms that are classiying in nature. We first obtained a baseline score using a single-layer multinomial logistic regression model; from our research, we observed that others that have tackled automated essay scoring in the past have employed multinomial logistic regression as the typical baseline 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 as a simple combination of the words that constitute the essay. In essence, an essay can be represented as 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, as depicted below.

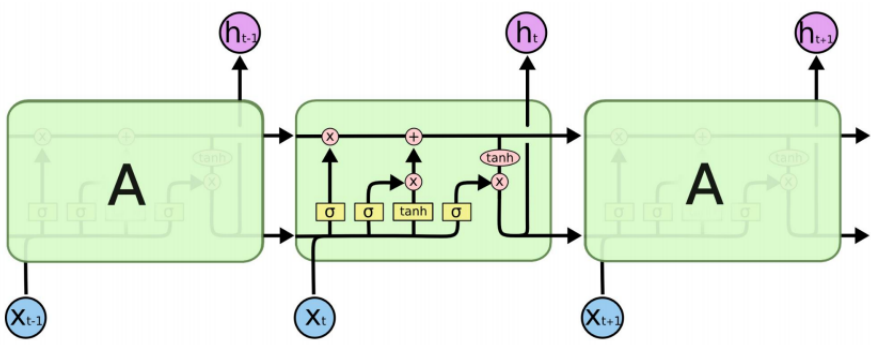


Figure 1: Uni-directional LSTM with a single cell detailed

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 backpropagated 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, as can be seen in Figure 2.

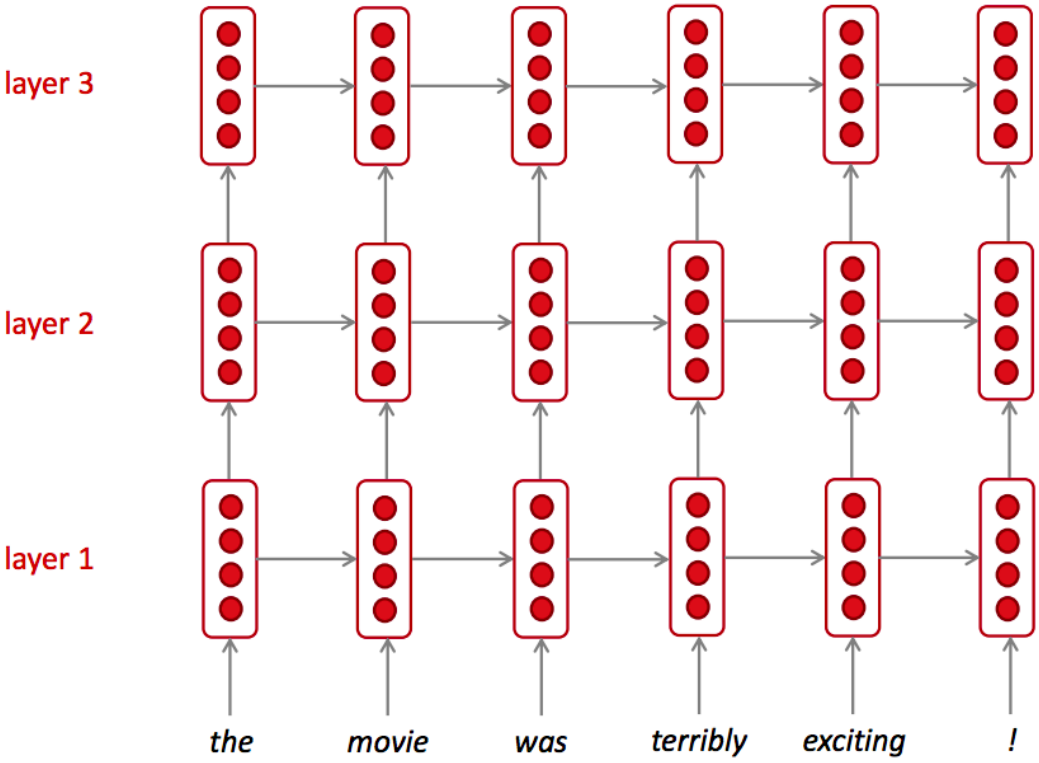


Figure 2: Uni-directional, Multi-Layer LSTM Network

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). We see the general structure below:

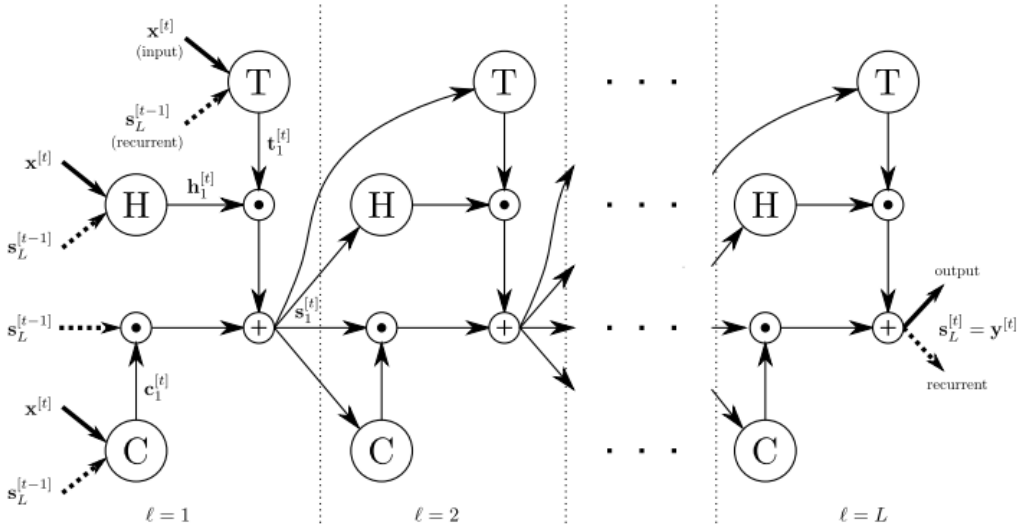


Figure 3: A single computation within the RHN layer inside the recurrent loop.

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 $\mathbf{W}_{\mathbf{H},\mathbf{T},\mathbf{C}} \in \mathbb{R}^{n \times m}$ as the input weight matrix and $\mathbf{R}_{\mathbf{H_l},\mathbf{T_l},\mathbf{C_l}} \in \mathbb{R}^{n \times n}$ as the recurrent weight matrix. Let $\mathbf{s_l}$ 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 $l \in \{1, 2, \ldots, L\}$ as follows:

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

where

$$\begin{aligned}
\mathbf{h}_\ell^{[t]} &= tanh(\mathbf{W}_H \mathbf{x}^{[t]} \mathbb{I}_{\{\ell=1\}} + \mathbf{R}_{H_\ell} \mathbf{s}_{\ell-1}^{[t]} + \mathbf{b}_{H_\ell}), \\
\mathbf{t}_\ell^{[t]} &= \quad \sigma(\mathbf{W}_T \mathbf{x}^{[t]} \mathbb{I}_{\{\ell=1\}} + \mathbf{R}_{T_\ell} \mathbf{s}_{\ell-1}^{[t]} + \mathbf{b}_{T_\ell}), \\
\mathbf{c}_\ell^{[t]} &= \quad \sigma(\mathbf{W}_C \mathbf{x}^{[t]} \mathbb{I}_{\{\ell=1\}} + \mathbf{R}_{C_\ell} \mathbf{s}_{\ell-1}^{[t]} + \mathbf{b}_{C_\ell}),
\end{aligned}$$

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

This process is shown in the Figure (2), where we can see the H, C, and T parameters corresponding to the variables $\mathbf{s}_l^{[t]}, \mathbf{t}_l^{[t]}, \mathbf{h}_l^{[t]}$ in the equations shown above.

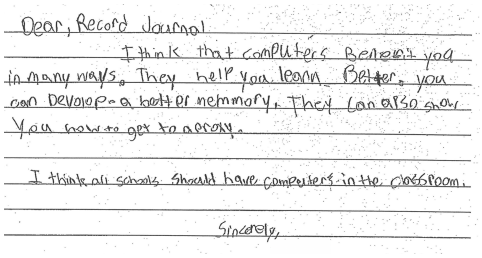
Similarly to how the output for an LSTM was handled, we take the sum of the outputs $\mathbf{s_L}$ of each highway stack and pass the resulting vector into a fully connected layer. This fully connected layer then produces the classification probabilities for the essay, which is then used to predict the 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.

4 Experiments

4.1 Data

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. Thus, whenever a handwritten word is illegible, it is transcribed as "illegible" or "???". We believe that in general, it makes sense that poorly written essays would score relatively worse, we decided to leave those words as they are. Two examples of these handwritten essays are shown below.

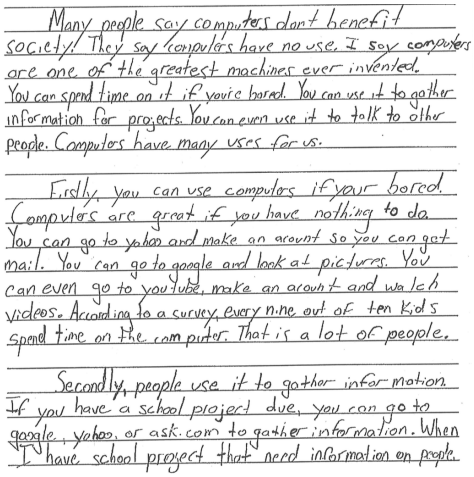

Dear, Record Journal

I think that computers Benefit you in many ways. They help you learn. Better, you can Develop a better memory. They can also show you how to get to a essay.

I think all schools should have computers in the classroom.

Sincerely,

Figure 4: A bad essay


Many people say computers don't benefit society! They say computers have no use. I say computers are one of the greatest machines ever invented. You can spend time on it if you're bored. You can use it to gather information for projects. You can even use it to talk to other people. Computers have many uses for us.

Firstly, you can use computers if your bored. Computers are great if you have nothing to do. You can go to yahoo and make an acount so you can get mail. You can go to google and look at picturns. You can even go to youtube, make an acount and watch videos. According to a survey, every nine out of ten kids spend time on the computer. That is a lot of people.

Secondly, people use it to gather information. If you have a school project due, you can go to google, yahoo, or ask.com to gather information. When I have school project that need information on people.

Figure 5: A good essay

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 meaning of the word while maintaining 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.

Finally, we want to find the proper GLoVe embeddings for each of the words in our essay, before inputting them into our model.

4.2 Evaluation Method

To evaluate our model, we have standard metrics for accuracy and precision. First, we can calculate strict error, where we iterate through each of the m essays and calculate the accuracy A as follows:

$$A = \frac{\sum_{i=1}^{m} 1\{p_j = s_j\}}{m}$$

where p_j is the predicted score for essay j and s_j is the true score.

4.3 Experimental Details

The model parameters for our stochastic gradient descent we used were as follows:

- Learning rate: 0.001

- Max epoch: 15
- Dimension size of embeddings: 300
- Frequency of validation: every 1000 iterations
- Layer/stack depth: 3

Additionally, we ran our models on Microsoft Azure services using the NV6 configuration, which consists of 6vcpus, consisting of 56 GB of memory. The training took approximately 20 hours for each of the models on these configurations.

4.4 Results

The following table highlights the results of our work.

Table 1: Performance of the Models

Baseline Models		
Model	Training Time (hrs)	Accuracy (%)
Logistic Regression	0.56	0.452
Single-Layer LSTM	7.43	0.540
Neural Network 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 Analysis

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 (although just barely) 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. However, the reason they lost a point was entirely because of the spelling. However, misspelled words in our model do not just contribute to one missed point because all misspelled words that aren't in the vocabulary are automatically embedded as the unknown word token. Thus, they contribute negatively to the essay far more than just one missed point, and thus the model is not very effective at dealing with this scenario.

However, the model did handle length differences relatively well. For example, there were essays that scored remarkably well, despite being on the shorter end. For example, the essay below:

> Dear Newspaper People, I think that computers do benefit society for a few reasons. Computers make work easier they can do things people can't like solve difficult problems, and kids like playing games on them. Computers make work easier and neater. Typing is often faster than writing, and is always easier to read. Studies have also shown that people who use computers to do work finish faster and have up to @PERCENT1 more free time to do whatever they want. Computers also have e-mail, which allows you to send work that you have done to your boss without printing it and wasting paper. Computers can solve difficult problems that people either can't do, or don't want to waste their time doing. If you need to solve a math problem you use a calculated or a computer. So you don't have to figure it out yourself. I have to do a lot of math problems for homework and I think it is much easier to use a calculator. My last reason why computers are good is that kids like playing games on them. A lot of people say that kids shouldn't play computer games but some of them are educational. Even non-educational games help kids to have fun and have something to look forward to after their work is done. @PERCENT2 of kids say that they are happier when they have something fun to look forward to, and happier kids do better work. I hope that after reading this you will understand how much computers contribute to society.

scored a 2 on the 0-3 scale, despite being on the shorter end of the range of 150-550 words at 250 words. However, all three models prediction was a 2 as well, showing that length, though important in determining quality, was handled appropriately.

6 Conclusion

Overall, the model did relatively well in getting close to the correct score, even if it didn't exactly match the score. However, it was not ideal measurements regardless. Perhaps, we could improve upon these models by making the LSTM models bidirectional. This might help improve the complexity of our model and predictions. Additionally, it might be interesting to consider using character embeddings rather word embeddings and seeing how that would affect the accuracy of the model, and how it would handle misspelled words differently than the current model, which dramatically affects the score by automatically throwing it an unknown token. Additionally, there was a lot of research done into the potential use of convolutions, and convolutional layers working in tandem with the highway network, something that 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.

Acknowledgments

We would like to thank our professor Christopher Manning for helping us acquire the tools necessary to conduct this research. Additionally, we would like to thank our mentor Amita Kamath 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