Automated Essay Scoring

Aman Agrawal
IIT Delhi
cs1150210@iitd.ac.in

Suyash Agrawal
IIT Delhi
cs1150262@iitd.ac.in

Abstract

Automated essay scoring systems(AES) are used in evaluating and scoring student essays written based on a given prompt. Recent advances in Deep Learning and Natural Language Processing have produced state of art results in this task. These systems are approaching human level performance according to the evaluation metrics, but their qualitative performance in actual scenarios is not yet explored. In this paper we explore and analyze how these systems perform in real life scenarios as compared to other non-neural models. We also explore some ways to overcome these shortcomings.

1 Introduction

Essay writing is usually a part of the student assessment process. Several organizations, such as Educational Testing Service (ETS)(ets), evaluate the writing skills of students in their examinations. Because of the large number of students participating in these exams, grading all essays is very time-consuming. Thus, automated essay scoring systems(AES) are used in evaluating and scoring student essays written based on a given prompt. The performance of these systems is assessed by comparing their scores assigned to a set of essays to human-assigned gold-standard scores. Since the output of AES systems is usually a real valued number, the task is often addressed as a supervised machine learning task. Recently, there is a lot of development of deep neural models for the task of automated essay scoring. These models have shown to out perform the traditional models based on hand crafted features and currently have the state of art performance in this task. The current state of art model that we presently know of is from Taghipour et al. (Taghipour and Ng, 2016a) and the state of art non-neural model is Enhanced AI Scoring Engine (EASE) (eas). In this paper we explore the performance of deep-learning based systems on real life scenarios. We try to see if their performance is actually better than traditional non-neural based models in a qualitative sense and analyze the issues that these systems can cause.

2 Related Work

There exist many automated essay scoring systems (Shermis and Burstein, 2013) and some of them are being used in high-stakes assessments. E-rater (Attali and Burstein, 2004) and Intelligent Essay Assessor (Foltz et al., 1999) are two notable examples of AES systems. In 2012, a competition on automated essay scoring called *Automated Student Assessment Prize (ASAP)* was organized by Kaggle and sponsored by the Hewlett Foundation. A comprehensive comparison of AES systems was made in the ASAP competition. Many AES systems have been developed to date, and most of them have been built with hand-crafted features and supervised machine learning algorithms. Researchers have devoted a substantial amount of effort to design effective features for automated essay scoring. These features can be as simple as essay length (Chen et al., 2013) or more complicated such as lexical complexity, grammaticality of a text (Attali and Burstein, 2004), or syntactic features (Chen et al., 2013). Readability features (Zesch et al., 2015) have also been proposed in the literature as another source of information. Moreover, text coherence has also been exploited to assess the flow of information and argumentation of an essay (Chen et al., 2013). A detailed overview of the features used in AES systems can be found in (Zesch et al., 2015).

Recently due to a lot of surge in deep-learning, a lot of neural models have been proposed which claim to beat the human level performance on evaluation metrics. Unlike traditional systems these systems, accepts an essay text as input directly and learns the features automatically from the data. A lot of models with varying architectures have been developed in the recent years. Some of them use a long short-term memory network (LSTM) (Taghipour and Ng, 2016b), while others use a convolutional neural networks (CNN) (Dong and Zhang, 2016) for the effect of automatically learning features. Some of them also use an augmented method to learn word embeddings (Alikaniotis et al., 2016).

We have tried to analyze these models over various qualitative metrics on which a human scorer would have graded the essays. We consider the length of the essay, grammatical and spelling mistakes, and the flow of ideas and organization of essay. Along with this, we also try to test if any of these models recognize deviation from the specifed topic. We finally suggest some ways to improve the performance of model on these qualitative metrics.

3 Models

3.1 Neural Model

The present state of the art neural model is from (Taghipour and Ng, 2016b). They use a simple a LSTM based model, where the output of each LSTM unit is averaged over time, to get a representation of the essay. Finally, they use a linear layer with sigmoid activation which maps its input vector generated by the mean-over-time layer to a scalar value and limits the possible scores to the range of $(0, 1)$.

3.2 Non-Neural Model

The state of the art, open-source non neural model is Enhanced AI Scoring Engine (EASE) (eas). Unlike, the neural model, this model uses a lot of hand-crafted features to learn a vector representation of the input essay. These hand-crafted features encode the essay length, spelling mistakes, grammatical correctness and some of content-based features. Finally, a Gradient Boosting Regressor is used to predict the score of the essay.

Prompt	**Essays Tested**	**% Increase**	
		Neural	**Non-Neural**
1	37	**72.785**	**37.240**
2	45	53.428	**33.843**
3	93	**137.284**	**56.061**
4	85	**235.539**	**44.515**
5	70	**145.933**	78.571
6	72	**119.254**	161.905
7	53	40.181	**14.659**
8	23	0.279	**4.141**

Table 1: Effect of Length

4 Experiments

4.1 Setup

The dataset that we have used in our experiments is the same dataset used in the ASAP competition run by Kaggle. We use quadratic weighted Kappa as a quantitative evaluation metric. We trained both the state of the art models described in section 3 using the code released by the authors. Same hyper-parameters were used as reported in respective papers in order to reproduce the results stated in the papers.

The dataset contains essay from eight different prompts. Since the test data is not publicly available, the model was trained on 64% of the data and tested on the remaining 20%. The remaining 16% was used to tune hyper-parameters and select the best model. All of our analysis stated below has been done using the test dataset.

4.2 Effect of Length

We tried to analyze the effect of the length of the essay on the prediction of the models. Typically, low scoring essays have low scores and this can be seen from fig. 1. Let, μ_l denote the mean length of the essays in the test set and σ_l denote the standard deviation of the lengths. We selected essays from each prompt that had length less than $\mu_l - 2\sigma_l$ and appended the same essay till the point where the length of the essay exceeded $\mu_l - 2\sigma_l$. The mean score of these essays before and after increasing length were calculated and the percentage increase in the score was calculated. 1 shows that the average increase in the predicted score of the essay is 100.58% while for some test sets the increase in the score was close to 235.53%.

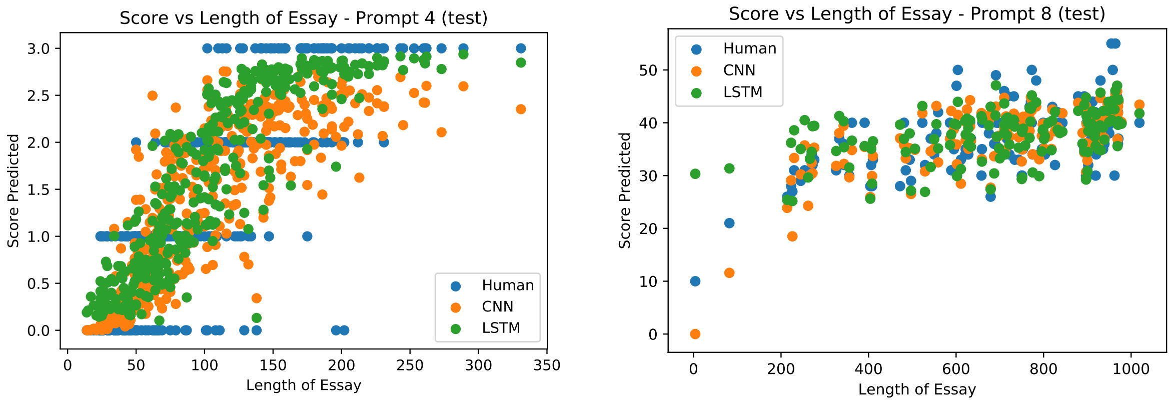

Figure 1: Distribution of Length of essay and associated score in the test dataset

4.3 Effect of Spelling Mistakes

We tried to analyze the effect of the spelling mistakes on the prediction of the models. We randomly chose 20% of the words of the essay and then randomly shuffled the characters of the words to generate words with wrong spellings. We had a set of predicted scores for the essays with correct spellings and a new set of predicted scores for essays with wrong spellings. We calculated QWK between these two sets of scores. table 2 shows that for the neural model, the QWK is close to 1 for most prompts whereas it is very close to 0 for non-neural model. Thus, for the neural model there is no disagreement between the scores predicted for essays with correct spellings and essays with 20% wrong spellings.

4.4 Effect of Word Shuffle

We tried to analyze the effect of the bad grammar on the prediction of the models. We randomly shuffled the words of each sentence in the essay. This resulted in an essay with terrible grammar, and thus should be scored low by any good AES system. We had a set of predicted scores for the essays with good grammar and a new set of predicted scores for essays with bad grammar and a lot of grammatical mistakes. We calculated QWK between these two sets of scores. table 2 shows that for the neural model, the QWK is close to 1 for most prompts whereas it is significantly less than 1 for the non-neural model. Thus, for the neural model there is no disagreement between the scores predicted for essays with correct grammar and essays with bad grammar.

4.5 Effect of Sentence Shuffle

We tried to analyze the effect of the bad organization and flow of ideas on the prediction of the models. We randomly shuffled the sentences of the essay. This resulted in an essay with terrible organization and flow of ideas, and thus should be scored low by any good AES system. We had a set of predicted scores for the essays with good flow of ideas and a new set of predicted scores for essays with a poor organization. We calculated QWK between these two sets of scores. table 2 shows that for both the the models, the QWK is close to 1 for most prompts. Thus, there is no disagreement between the scores predicted for essays with good organization and flow of ideas and essays without it for both the models.

5 Methods Proposed

5.1 Data Bootstrapping

In the current dataset, most of the essays are reasonably good and hence the neural models are not able to learn that longer essays does not necessarily mean better scores. Also, it is not able to learn that spelling mistakes should reduce the overall score. To overcome this problem we propose to augment the dataset with negative examples like adding longer essays with lower scores (by concatenating low score and small essays multiple times).

5.2 New Tokens

Presently, the neural model used a special symbol of $<UNK>$ for out-of-vocabulary words. Thus words which have wrong spelling, were replaced with the embedding of $<UNK>$. We added a new symbol $<MISSPELL>$. We checked if word has

Prompt	Essays Tested	Spelling Mistakes		Word Shuffle		Sentence Shuffle	
		Neural	Non-Neural	Neural	Non-Neural	Neural	Non-Neural
1	357	**0.944**	**0.027**	**0.940**	0.991	**0.919**	**1.000**
2	360	0.408	**0.013**	0.739	**0.515**	**0.847**	**0.997**
3	345	**0.901**	**0.163**	**0.893**	0.709	**0.906**	**0.995**
4	355	**0.936**	**0.019**	**0.930**	**0.495**	**0.944**	**0.986**
5	361	**0.930**	**0.027**	**0.913**	**0.251**	**0.903**	**0.998**
6	360	0.591	**0.017**	**0.914**	**0.306**	**0.907**	**0.984**
7	314	**0.871**	**0.081**	**0.944**	0.962	**0.977**	**0.998**
8	145	**0.822**	**0.086**	**0.934**	0.931	**0.991**	**0.999**

Table 2: Effect of spelling mistakes, grammatical errors and poor organization in terms of QWK between old predicted scores and new predicted scores

Prompt	Previous Score	New Score	% Decrease
1	0.647	0.588	**26.16**
2	0.469	0.268	**10.07**
3	0.590	0.606	**43.72**
4	0.458	0.376	**36.21**
5	0.603	0.567	**33.80**
6	0.667	0.618	**25.98**
7	0.532	0.418	**25.39**
8	0.614	0.449	**11.68**

Table 3: Scores of new model (with <MISSPELL>token) and old model on essays with 20% misspelled words.

Prompt	LSTM Score	H-LSTM Score
1	0.633	0.633
2	0.408	0.401
3	0.603	0.604
4	0.498	0.495
5	0.626	0.618
6	0.719	0.701
7	0.543	0.543
8	0.531	0.532

Table 4: Effect of sentence shuffle on LSTM and Heirarchical LSTM models

correct spelling using a corpus of 200,000+ words and if it had a wrong spelling, it was replaced with the symbol <*MISSPELL* >.

5.3 Hierarchial LSTM

The current SoTA neural model, used basic LSTM cell to encode the whole essay. Typically, an essay has a length of about 500 tokens, and it is very hard for a LSTM to remember this long sequence. We proposed using hierarchical LSTM network to mitigate the long history length and to model grammar and coherence better. We effectively run a LSTM over the words of each sentence and generate a vector representation of the sentence and then run a LSTM over the representations of the sentences learned in the previous layer to generate a representation of the essay.
We were not able to get satisfactory results with this approach (shown in table 4). One of the possible explanations for this is that the dataset size is too small for hierarchical LSTM to learn the grammar.

6 Conclusions

[Aman: Write this up]

References

???? Enhanced ai scoring engine. https://github.com/edx/ease. Accessed: 2018-04-25.

???? Ets - measuring the power of learning. https://www.ets.org/. Accessed: 2018-04-25.

Dimitrios Alikaniotis, Helen Yannakoudakis, and Marek Rei. 2016. Automatic text scoring using neural networks. *arXiv preprint arXiv:1606.04289* .

Yigal Attali and Jill Burstein. 2004. Automated essay scoring with e-rater® v. 2.0. *ETS Research Report Series* 2004(2).

Hongbo Chen, Jungang Xu, and Ben He. 2013. Automated essay scoring by capturing relative writing quality. *The Computer Journal* 57(9):1318–1330.

Fei Dong and Yue Zhang. 2016. Automatic features for essay scoring–an empirical study. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 1072–1077.

Peter W Foltz, Darrell Laham, and Thomas K Landauer. 1999. The intelligent essay assessor: Applications to educational technology. *Interactive Multimedia Electronic Journal of Computer-Enhanced Learning* 1(2):939–944.

Mark D Shermis and Jill Burstein. 2013. *Handbook of automated essay evaluation: Current applications and new directions*. Routledge.

Kaveh Taghipour and Hwee Tou Ng. 2016a. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 1882–1891.

Kaveh Taghipour and Hwee Tou Ng. 2016b. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. pages 1882–1891.

Torsten Zesch, Michael Wojatzki, and Dirk Scholten-Akoun. 2015. Task-independent features for automated essay grading. In *Proceedings of the Tenth Workshop on Innovative Use of NLP for Building Educational Applications*. pages 224–232.