Towards Building FoolProof Automated Essay Scoring System Robust to Adversarial Attacks

Dheeraj Kumar Kondaparthi
Computer Science department
University of Illinois at Chicago
dkonda2@uic.edu

Abstract — Automatic essays scoring(AES) systems appear in increasingly important standardized testing applications. Admission to graduate school, teacher evaluations and possibly student grade promotion depends on fair and reliable functioning of automated essay scoring systems, but often bad actors are attempting to manipulate scores thereby resulting in higher scores for low quality essays. So, in this paper, I set to answer following questions: can adversarial attacks on neural automatic essays scoring systems be generated? To what extend neural automated essay scoring systems are susceptible to adversarial attacks? Is it possible to build foolproof neural automated essay scoring system susceptible to adversarial attacks? I attempt to answer these questions by building a dual score CNN based neural AES system and testing its robustness by generating anchor adversarial attacks. I found out that system is robust to adversarial to most of the essays, while a small percentage of essays can be still be manipulated by adversarial attacks.

1. INTRODUCTION

Automated Essay Scoring (AES) is a method which uses both statistical and natural language techniques to assign a numerical grade to an essay on a given marking scale. Specificaly, in simple terms, AES is art of giving students automatic, iterative and correct scores on their essays without involvement of human grader. Essay writing is essential part of student assessment process to access their writing skills. Due to exponential increase in number of students taking writing assessment tests, it usually takes lot of time and effort for human graders to grade the essays manually. It is very challenging for human graders to consider all the features while grading the essays thereby increasing the demand for AES systems. Therefore, organizations are using AES systems to grade student essays automatically.

AES systems take input as essay and assign grade based on relevance of essay to the given topic, grammar, organization and content. These systems depend of heavily carefully designed features called as feature engineering which is most difficult part of AES systems. This is important because human grader considers grammar, spelling mistakes, words organization while grading the essays but to generate such features by AES system is complex task. For decades standard approaches to AES tasks has been feature engineering models, e-Rater from ETS(Attali et al., 2004). Individual features can be quantities like vocabulary size, essay length, average length of word, or other quantities derived from the student-supplied text. These features could then be fed into any number of predictive models, including linear regressions, ridge regressions, and feed-forward neural networks. These systems often boast higher than human reliability(Page et al., 1995) and instant scoring once the model has been trained.

In their early work on AES systems, Page et al., 1995 anticipated three main objections to use of AES systems: Humanist objections, defensive objections and construct objections. The humanist objection is, simply, that machines do not understand or appreciate essays the way humans do, and computer judgments should be outright rejected. Construct objections center around what the computer is measuring and how (or if) they directly relate to the constructs of interest. The general response is simple: if human-machine reliability is high, then AES systems do a good job of predicting human scores, therefore they are capturing the essence of the construct of interest(Attali et al., 2006).

Defensive objections, however, consider the possibility of an attacker (or bad actor) interacting with the AES system by submitting intentionally deceptive essays with the intent of receiving an “undeserved” grade. Page et al., 1995 response first assume that “motivated students eager to receive good scores” would not behave this way and ultimately suggests the retention of a human reader to flag off-topic or bizarre essays.

Classic feature-based models are susceptible to nonsense attacks as well as simple word-replacement strategies(Bejar et al., 2014). Knowing or guessing the input features can allow for the development of model specific attacks. Neural AES systems, on the other hand, do not take input features, but instead rely on RNN and/or CNN architectures to learn important features. This adds a layer of up-front security, but in computer vision, where the CNN is leader, adversarial examples can be developed without access to systems and are known as black-box attacks(GoodFellow et al., 2017) and can be hard to detect(Carlini et al., 2017).

2. RELATED WORK

Efforts to develop AES system were made as early as 1966 by Ellis Page. He developed a software called Project Essay Grade (PEG). This system uses linear regression over vectors of textual features. Burstein, J et al., in 2004 developed a system called e-rater and Foltz et al., in 1999 developed Intelligent Essay Assessor which are notable AES systems.

A lot of research has been devoted to design effective features which can be as simple as essay length by Chen and He, 2013 to more complex features such as syntactic features by Chen and He, 2013. Text Coherence was also exploited by Chen and He, 2013 to assess flow of information and argument in the essay. Bayesian scoring system was proposed by Rudner and Liang in 2002 that uses multinomial and Bernoulli Naïve

Bayes models to classify texts bases on content and style features.

Recent approaches have tried incorporating more diverse set of features into models, Klebanov and Flor in 2013 have developed a set of features consisting of count of highly associated, mildly associate and dis associated words thereby showing that model performance can be improved.

The work presented in this paper is at the intersection of two specific strands of neural NLP research, adversarial example generation and interpretability. There are mainly two types of adversarial examples from NLP literature. The first is the insertion of ungrammatical text. Prior work has shown that this strategy has proven surprisingly effective in reducing model accuracy in reading comprehension models(Jha et al., 2017). However, this type of attack is not appropriate for application that might retain human reader. The other popular attack in NLP is random-perturbation approach. This approach goes through candidate words and replaces them either with a random token or a word that is similar in embedding space. This approach was successful in fooling sentiment analysis models and natural language interference tasks. One consideration is that while generating adversarial examples for tasks that generally take shorter input strings has been done(Alzantot et al., 2018) using a random perturbation approach for essay-length input is impractical, because the search space for essay modification is orders of magnitude larger than that of a single sentence.

In order to narrow down search space, efficient developing of adversarial attacks may benefit from understanding how AES model is making its scoring decisions. One particularly promising model for neural network interpretability is anchors(Ribeiro et., 2018). Anchors represent the minimum input required to guarantee a model delivers a particular classification. The authors highlight this approach as model-agnostic and high precision. By attempting to identify anchors for machine scored essays, we can gain insights into the text being used to make classification results. We can then propose, that by identifying, studying, and manipulating anchors, we may gain insight into how an AES system makes scoring decisions and what vulnerabilities may exist in the logic of those decisions. Qualitative anchor analysis(Ribeiro et., 2018) has suggested that users who relied on anchor-based models to make predictions performed more quickly and accurately than individuals who were provided with more quantitative measures, like LIME scores(Ribeiro et., 2016). Because of this, anchors are viewed as a promising strategy for developing generalizable adversarial attacks. In this paper, I have used essay topic to generate anchors and to experiment with different adversarial attacks.

3. METHODOLOGY

Most AES systems are generally proprietary software, therefore adversarial examples are developed through black box attacks. To study adversarial examples in AES system, following approach is followed: First, construct an AES system. The model employs Convolution Neural Network(CNN). CNN is opted as apposed to Recurrent Neural Network(RNN) because of the efficient computation of the CNN and potential of CNN to model relationships between non-adjacent words. After developing a functional CNN based AES system, anchors are found for held out test examples. In this project, anchors are constructed based on essay topic. After anchor search, strategies are developed for targeted adversarial example construction. These strategies used anchors to guide text injection into essays, an attempt to induce change in score assigned by the model. If a strategy can induce score variation without knowledge of anchors in an essay, we can classify the strategies as successful.

3.1 DATSET

The dataset used in this project is from the Hewlett Foundation's Automated Student Assessment Prize (ASAP) that was released on Kaggle seven years ago. The data consists of essays written by students from grades 7-10 pulled from eight different datasets; each of the eight datasets was generated from written student responses to different writing tasks (prompts). Each essay was graded by at least two different graders on at least two different rubrics. The data includes each of the prompts, grading criteria, basic descriptive statistics, and other pertinent information. In total, there are 12,978 essays in the training set and 4,254 essays in the test set (17,232 total essays). Each essay is on one of eight possible topics. These essays range from approximately 150 words in length to 550 words. A subset of the data, specifically 1,800 essays from the same 10th grade writing prompt was used. This was done for a two main reasons. First, each prompt is graded on a different rubric and different scale. Because the project is not focused on developing a universal AES system, score equating across different prompts is beyond the scope of what was needed. Second, developing own model on a restricted dataset approximates the black box attack development process and will lend external validity to future work on the transferability of our adversarial examples. The specific essay set used for evaluation discusses the issue of censorship in libraries. Students are asked to write a persuasive essay articulating their view on the topic and scores range from 1-6.

3.2 MODEL DESIGN

To generate, a closed "black box" host for adversarial testing, an AES CNN based system is created by using Lang et al., 2019 implementation as basis of design. This model is also based on initial sentence classification model created by Kim et al., 2014. Lang model is used only for the half of essay evaluation i.e. single domain scoring system whereas in this project, Lang model is expanded to consider language construction and grammar in addition to meaning, thereby predicting domain1 and domain2 scores.

After experimenting with Kim et al., 2014 based model initially, model performance was observed to be poor, which was expected given massive difference in input length. Because essays are being modeled not sentences, using kernel sizes 3, 4 and 5 does not seem to be capturing relevant information and structure across entire input. Therefore, more convolutions with larger kernel seizes are added capture information only in small group of words but also in entire sentences, groups of sentences, paragraphs and groups of paragraphs. So in addition to original kernel size, convolutions layers of $k \in \{10, 15, 20, 50, 100, 150, 200, 350, 500\}$ are added based on the idea that single sentence is around 10 words and a paragraph is around 100 words. To capture language and grammar information another CNN model was built with kernel size $k \in \{5, 10, 15, 20\}$ to identify language features of short sentences, long sentences and paragraphs.

The final model is built as follows:

1. Each complete essay is tokenized into word tokens, and the words are assembled into a dictionary (including both words and punctuation), with a maximum of 20,000 words.
2. Using Stanford's GloVe 300dimension word embeddings, the words for each essay are embedded into vectors, and those embeddings are concatenated into a single vector.
3. The essay vector is then passed into two parallel CNNs (see Figure), one to score writing style (content and voice), and the other to score language conventions (grammar and punctuation).
4. In the "Domain 1" (Writing Style) Deep Convolutional Neural Network:
    a. The essay is passed through an embedding layer, with GloVe embeddings.
    b. Eleven parallel Convolutional layer sets of 3 feature maps each convolve over the essay, using ReLU activation, and each with a different kernel size from the range of 3,4,5,10,20,50,100,150,200,350 and 500.
    c. Learning representation of each Conv1D layer is then reduced by 50% via a Maxpooling layer.
    d. Dropout of 50% is applied to address overfitting.
    e. Essay representation is then flattened and is passed into a fully connected dense layer of 6 neurons with softmax activation to generate output score between 1 to 6 for writing style.
5. In the "Domain 2"(Grammar and Language) Deep Convolution Neural Network:
    a. The essay is passed through an embedding layer, with GloVe embeddings.
    b. Four parallel Convolutional layer sets (see Figure 4) of 3 feature maps each convolve over the essay, each with a different kernel size from the range of 5,10,20 and 50
    c. Learning representation of each Conv1D layer is then reduced by 50% via a Maxpooling layer.
    d. Dropout of 50% is applied to address overfitting.
    e. Essay representation is then flattened and is passed into a fully connected dense layer of 4 neurons with softmax activation to generate output score between 1 to 4 for language conventions.
6. Both Domain 1 and Domain 2 scoring outputs are fed into a scoring engine. First performance is evaluated for both different scoring methods. Then all subsets of essay scoring accuracies are computed, and their intersection recorded.
7. A confusion matrix is generated for final score, for essays scored correctly by both domains, either domain, or neither.
8. A Final Score is evaluated and produced. Only for essays that received a correct score on both domains are considered accurately scored.

One important consideration for hyperparameter selection was that while the investigation needed an AES model, but the purpose was not to develop a perfect AES model. The model was only required have an acceptable performance. E-rater v.2.0, an AES system that has been deployed by ETS, has exact agreement between human and machine scores between 0:50 - 0:58, so this was considered as lowest acceptable accuracy(Attali et al., 2014).

3.3 MODEL EVALUATION

Primary evaluation criteria used for AES model is accuracy. Because AES is trained to produce agreement with a human scorer and exact agreement is valued in human scoring applications, this seemed most appropriate. As such model is trained as a classification task using cross – entropy loss. Model was trained in ranges of 2 to 200 epochs and epoch count 8 has highest accuracy for domain1 and domain2 scores. These two models, each for a domain is packaged into two h5 models and isolated from model training as foreign entity.

To benchmark AES performance, three references were selected – high score of ASAP student competition from 2013, the most recent AES manual published by ETS(for GRE) and chances of randomly guessing a correct set from a most likely set of scores(3 or 4).

- ASAP student competition high score – 81.40 %
- **Dual CNN model(model built for the project) – 56.83%**
- ETS GRE scoring system – 52.00%
- Random guess(set of 3 or 4) – 25.00%

It is also worth noting that ASAP student competition high score was for single domain(domain1) score evaluation. So "black box" model accuracy is 76.83% vs 81.40% in competition. So, we can consider this model acceptable for testing its robustness for adversarial attacks.

3.4 ADVERSARIAL ATTACK STRATEGIES

To evaluate susceptibility of model to adversarial examples, this project set out to systematically examine the stability of model, search for potential attack strategies by identifying anchors in essays, evaluate efficacy of these anchor-based attack strategies. Experiments detailing these approaches are presented below.

4. EXPERIMENTS

To begin building a picture of adversarial examples in AES, experiments are broken into three categories.

4.1 IDENTIFYING CANDIDATE ANCHORS

Kanopka and Lang have used UNK token distribution and then beam search to identify anchors. However, problem with this approach is that it is slow, because each word in essay can be replaced with UNK token and essay length of W produces search space of 2^W possible essays. So, to increase efficiency of anchor search, we used "Censorship" and "Library" as anchor words because essay set title, model is trained on "Censorship in Libraries". Apart from these anchor words, "the" is used as control word.

4.2. MODEL STABILITY

Experiments, investigating model stability were designed to see extent to which scores produced by AES model were stable under input perturbation. Lang and Kanopka have suggested four types of attacks to test model stability, but in this project five types of attacks are proposed to investigate model stability. These attacks are:

1. **Shuffling**: Words in essay are randomly shuffled. One benefit of this type of attack is that models that solely relied on bag of words should be invariant to these types of attacks.
2. **Appending**: A target word is appended to the end of the essay with no more modifications.
3. **Single Substitution**: A word in essay is replaced with the target anchor word. The objective of single substitution attack is to test the model sensitivity to the location of a word perturbation.
4. **Insertion**: A target word is randomly inserted into an essay with no other modifications. This is distinct from substitution attack because, it fundamentally changes the distance between the words on either side of the insertion, potentially making predictions based heavily on convolutions of larger kernel sizes more unstable across perturbation.
5. **Progressive Overload**: Words in essay are cumulatively replaced with target anchor word. The objective of progressive overload is to determine how the model is affected by increasing and repetitive use of an anchor word.

4.3 EVALUATING ANCHOR BASED ATTACK STRATEGIES

In order to carry out above attacks described in section 4.2 following procedure was followed: First a set of essays was identified as base for perturbation; these essays are usually from test set. Next, a single shuffling perturbation is generated for each essay. Next target words are identified for appending, insertion, progressive overload and single substitution. Anchors selected for these attacks are "censorship" and "library". Perturbed essays are generated by applying each perturbation to base set of essays for each anchor word. After implementing these adversarial attacks on a sample of essays on test data, AES score predictions are observed to check if they are different than the original, unmodified essays and determine the influence of adversarial attack strategies on AES score prediction.

5. RESULTS AND DISCUSSION

In Table1 and Table2, evaluation of change in predicted writing content score and predicted language convention score is tabulated due to adversarial attacks per section 4.2. This includes adversarial attacks on both anchor and control words.

Attack	**Score Decreased**	**Score Increased**	**No Change**
Shuffling	4.0%	8.0%	88.0%
Appending	2.0%	2.0%	96.0%
Insertion	2.0%	0.7%	97.3%
Progressive Overload	2.0%	0%	98%
Single Substitution	2.7%	0.7%	96.7%

Table1: Change in Writing Content Score due to adversarial attacks

Attack	**Score Decreased**	**Score Increased**	**No Change**
Shuffling	14.0%	14.0%	72.0%
Appending	10.0%	4.0%	86.0%
Insertion	7.3%	6.0%	86.7%
Progressive Overload	10.0%	3.3%	86.7%
Single Substitution	8.0%	2.0%	90%

Table2: Change in Language Convention Score due to adversarial attacks

These results suggest that in most of the examples, the adversarial attacks do not affect the predicted writing content and language convention scores. However, in some cases, the adversarial attack results in a score prediction higher than the original score prediction and in other cases, the adversarial attack results in a score prediction lower than the original score prediction. In each case, the change in predicted score is one point. Table2 suggests that the adversarial attacks have a larger impact on the language convention score than the writing content score.

In Tables3 and 4, the mean score changes for the predicted writing content score (Table3) and predicted language convention score (Table 4) are tabulated and compared the results for both the anchor words ("library", "censorship") and the non-anchor control word ("the").

Based on these results, we can conclude that in general adversarial attacks resulted in lower essay score prediction. While this is evident for both writing content and language convention scores, the magnitude of score decrease is higher for the language convention score. The only adversarial attack where we see net score increase in writing content score prediction is shuffling attack.

Based on the progressive overload and single substitution attacks, we observed that the words which were chosen as suitable anchor candidates had a lower language convention score than the control word. For the insertion attack, we see the opposite trend: the control word had a lower mean language convention score as compared to the anchor candidates.

Attack	Anchor	Non- Anchor
Shuffling	4%	4%
Appending	0%	0%
Insertion	-2%	0%
Progressive Overload	-2%	-2%
Single Substitution	-2%	-2%

Table3: Mean change in Writing Content Score prediction due to adversarial attack

Attack	Anchor	Non- Anchor
Shuffling	0%	0%
Appending	-6%	-6%
Insertion	0%	-4%
Progressive Overload	-8%	-4%
Single Substitution	-7%	-4%

Table4: Mean change in Language Score prediction due to adversarial attack

Unsurprisingly, the attack that had the strongest impact on output scores from model was the shuffling attack. There are several finding to note. First, this type of perturbation seems to eliminate the possibility of any essay achieving the highest score on the rubric. Realistically, however, this type of attack is impractical for two reasons. First, it is hard to write an essay of low quality, randomly rearrange the words, and then submit it under time pressure. Second, and more importantly, a human reader would flag an essay manipulated in such a way as anomalous with a high probability.

From the results of appending, insertion, progressive overload and substitution attacks, there is either no change in score predictions or decrease on score prediction. From this, we can conclude that, scoring model is generally stable from these types of perturbations.

6. CONCLUSION AND FUTURE WORK

In this project, a dual score CNN model is built for AES. While this model performed better on predicting the writing quality than on grammatical accuracy, the combined score is lower than current state of the art AES systems which requires more development work to increase accuracy. In the project a framework was presented for adversarial search and development. Using anchors, suitable candidate injection texts were identified that had potential to induce some variance into results. Several perturbation strategies are experimented, and their associated effects were explored with both identified anchors and control words. It is found that the success of adversarial perturbation strategy is sensitive not only to specific essay being perturbed, but the location in which target text is introduced into the essay. Overall, the results demonstrate that the use of adversarial attacks is not effective in their intended purpose of achieving higher AES score predictions.

Four main areas are identified for future work. First, improving the model accuracy using GRU based attention model preferably with BERT embeddings. Second, improving efficiency of anchor searches over essay length inputs will be key for devising sophisticated adversarial attacks. Third, exploring more complex perturbation strategies over wide variety of inputs to determine the extent to which the anchors have already identified are promising candidates for adversarial attacks. Finally, generalizing beyond single word perturbations and experimenting with multiple word or sentence level perturbations to further test and improve model robustness of AES systems.

ACKNOWLEDGEMENTS

I would like to thank my professor Dr. Natalie Parde for giving me opportunity to work on this project and for all the useful discussions.

REFERENCES

Moustafa Alzantot, Yash Sharma, Ahmed Elgohary, Bo-Jhang Ho, Mani Srivastava, and Kai-Wei Chang. Generating natural language adversarial examples. arXiv preprint arXiv:1804.07998, 2018.

Yigal Attali and Jill Burstein. Automated essay scoring with e-rater R v. 2.0. ETS Research Report Series, 2004(2):i–21, 2004.

Yigal Attali and Jill Burstein. Automated essay scoring with e-rater R v. 2. The Journal of Technology, Learning and Assessment, 4(3), 2006.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, pages 3–14. ACM, 2017.

Robin Jia and Percy Liang. Adversarial examples for evaluating reading comprehension systems. CoRR, abs/1707.07328, 2017.

Yoon Kim. Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882, 2014.

Ellis B Page and Nancy S Petersen. The computer moves into essay grading: Updating the ancient test. Phi delta kappan, 76(7):561, 1995.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, pages 506–519. ACM, 2017.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model agnostic explanations. In AAAI Conference on Artificial Intelligence, 2018.

Briscoe, Y. F. a. Y. a., Neural Automated Essay Scoring and Coherence Modeling for AdversariallyCrafted Input. CoRR abs/1804.06898, (2018).

Dery, N.H. a. D. a., Neural Networks for Automated Essay Grading. Stanford University cs224d/huyenn, (2016).

Jain, Anant. "Breaking Neural Networks with Adversarial Attacks." Towards Data Science, Medium, 9 Feb. 2019, towardsdatascience.com/breaking-neural-networks-with-adversarialattacks-f4290a9a45aa.

Kaggle. "Develop an automated scoring algorithm for student-written essays." (2012). https://www.kaggle.com/c/asap-aes

Lang, K. K. a., Adversarial Examples for Neural Automatic Essay Scoring Systems. Stanford University cs224n/15720509, (2019).

Zhao, J. L. a. X. a., Automated Essay Scoring based on Two-Stage Learning. CoRR abs/1901.07744, (2019).

Landauer et al. 2003. Automated scoring and annotation of essays with the Intelligent Essay Assessor. In M.D. Shermis and J.C. Burstein, editors, Automated essay scoring: A cross-disciplinary perspective, pages 87–112

Hongbo Chen and Ben He. 2013. Automated essay scoring by maximizing human-machine agreement. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing.

Klebanov and Flor. 2013. Word association profiles and their use for automated scoring of essays. In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, pages 1148–1158.

Tagipour and Ng. 2016. A Neural Approach to Automated Essay Scoring

Lee, Michael and Nickerson, Micah. Adversarial techniques for automatic essay scoring systems. Link