A Comparative Study of Pretrained Language Models for Automated Essay Scoring with Adversarial Inputs

1st Phakawat Wangkriangkri
Department of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
phakawat.w@alumni.chula.ac.th

2nd Chanissara Viboonlarp
Department of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
chanissara.v@alumni.chula.ac.th

3rd Attapol T. Rutherford
Department of Linguistics
Chulalongkorn University
Bangkok, Thailand
attapol.t@chula.ac.th

4th Ekapol Chuangsuwanich
Department of Computer Engineering
Chulalongkorn University
Bangkok, Thailand
ekapolc@cp.eng.chula.ac.th

Abstract—**Automated Essay Scoring (AES) is a task that deals with grading written essays automatically without human intervention. This study compares the performance of three AES models which utilize different text embedding methods, namely Global Vectors for Word Representation (GloVe), Embeddings from Language Models (ELMo), and Bidirectional Encoder Representations from Transformers (BERT). We used two evaluation metrics: Quadratic Weighted Kappa (QWK) and a novel "robustness", which quantifies the models' ability to detect adversarial essays created by modifying normal essays to cause them to be less coherent. We found that: (1) the BERT-based model achieved the greatest robustness, followed by the GloVe-based and ELMo-based models, respectively, and (2) fine-tuning the embeddings improves QWK but lowers robustness. These findings could be informative on how to choose, and whether to fine-tune, an appropriate model based on how much the AES program places emphasis on proper grading of adversarial essays.**

Keywords—**Automated Essay Scoring, Embedding, Language Modeling, Adversarial Input**

I. Introduction

The ability to write coherently and fluently is imperative for students. Automated Essay Scoring (AES) has gained increasing attention as it allows language learner's writing skills to be assessed at scale. This type of assessment presents a challenge in natural language processing (NLP) because the grading criteria involves all levels of linguistic analyses. Systems can detect word-level and sentence-level errors, such as typos and grammatical errors, at a reasonable accuracy, but discourse-level errors, such as lack of coherence, focus, or structure, require a more sophisticated model, and remain the crux of AES. Although recent neural-network approaches are proven to be effective for grading naturally written essays [1]–[3], they are prone to exploitation by adversarial input which tricks the systems to yield good scores without respecting the actual grading criteria. Several past studies explored AES systems that detect coherency [4], [5] and prompt-relevancy [5] in order to rectify this problem.

In this study, we employed models that encode paragraphs of text and capture discourse-level context required for scoring essays. We experimented three popular transfer-learning text encoders: Global Vectors for Word Representation (GloVe) [6], Embeddings from Language Models (ELMo) [7], and Bidirectional Encoder Representations from Transformers (BERT) [8]. An ideal AES system should both correlate with manually-graded essay scores and be resistant to adversarial hacks; thus, we evaluated the models against both criteria.

Our contributions can be summarized as follows:

- We developed three AES-specific models utilizing different transfer-learning text encoders, and compared their performances using Quadratic Weighted Kappa (QWK) [9], which measures how well the models' predictions agree with human ratings.
- We also introduced a new evaluation metric based on robustness, which measures the models' resistance to adversarial essays. A model's QWK and robustness, considered together, are indicative of the model's suitability for the AES task. We found that the BERT-based model is the most suitable, followed by the GloVe-based and ELMo-based models, respectively. Nevertheless, we noticed that these models did not display sufficient robustness according to the evaluation metric used.
- We discovered that fine-tuning the models' pre-trained embedding weights on the dataset used in this work improves QWK but decreases the three models' robustness as a result of the models being more accustomed to poor writing.

II. Related Works

AES system has been improved with new techniques over the decades. The Intelligent Essay Assessor system, developed in 1999 [10], employed Latent Semantic Analysis, while the e-rater program used by Educational Testing Service [11] utilized linear regression with feature engineering to grade essays. In a program developed in 1998 [12], a Naive Bayes model classified texts based on their content

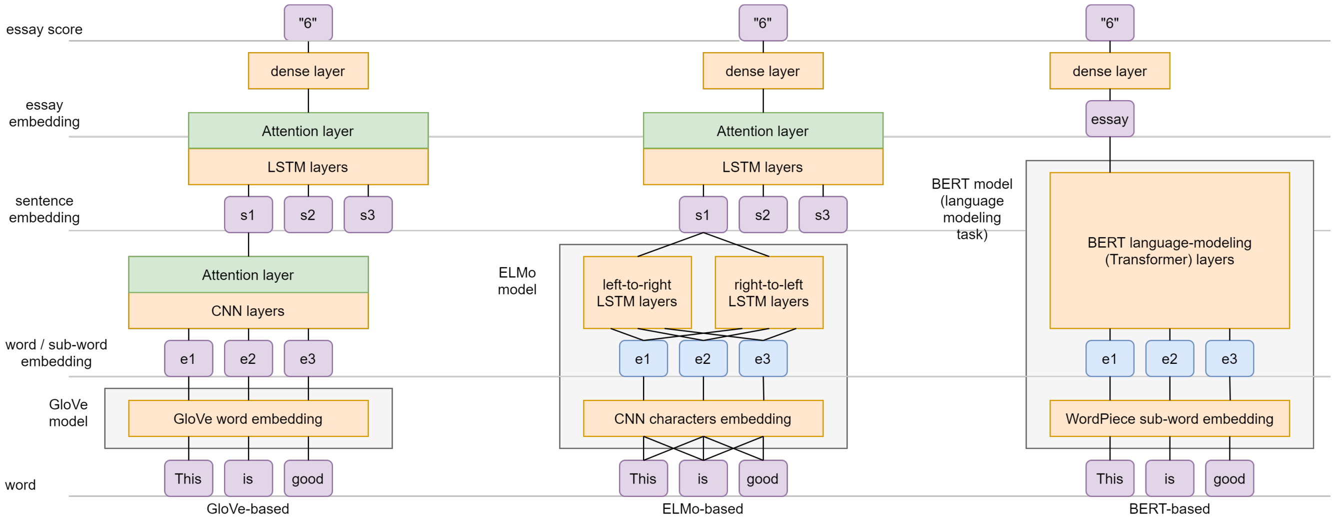

Fig. 1. Each of the three models (GloVe-based, ELMo-based, and BERT-based) are hierarchical in nature, moving up from word-level to essay-level.

and writing styles. AES has also been posed as a ranking problem [13].

More recent AES systems often incorporate neural networks. Various architectures have been used; for example, an LSTM model [14] with embedding layer and average pooling [1], and a CNN model [15] with sentence embedding [2]. Reference [3] introduced attention mechanism [16] to the AES task and found that CNNs are better-suited for detecting sentence structures, while LSTMs are preferable for evaluating document-level coherence.

Many AES systems utilize word or context embedding. Word embeddings encode knowledge of the relationships between words. The technique dates back to the early 2000's [17] and gained popularity in 2013 [18]. One of the prominent word embeddings is GloVe [6], which encodes the ratio of co-occurrence probabilities between pairs of words. Word embeddings' limitation is that a word's representation is the same in all contexts. Context embedding addresses this limitation by encoding text on a case-by-case basis and giving a unique representation to each text. Examples of context embeddings are ELMo [7], which encodes texts using two unidirectional LSTM models, and BERT [8], which encodes texts using a bidirectional transformer model [19]. In each model, embeddings are initialized with pre-trained weights, then are either fixed while training the main system or updated during training.

Due to the large number of parameters involved in AES systems, the exact criteria for grading essays is often obscure. System fragilities have been exposed by tricking them with incoherent essays that serve as adversarial inputs. Adversarial examples include well-written paragraphs repeated many times [20] or a set of well-written sentences randomly permuted [4]. Systems have been developed to detect adversarial essays, and each system usually employs a single text embedding. Reference [4] used Senna word embedding [21], [22] in their model of local coherence, while [5] preferred BERT context embedding for their system of two-stage learning framework.

To the best of our knowledge, our work is one of the earliest works to compare different text embedding methods on their suitability for detecting adversarial inputs, by creating and testing models that are structurally similar but based on three different text embeddings: BERT, ELMo and GloVe.

III. Model

In this section we describe our AES models. We developed three AES models based on different embedding techniques, namely GloVe-based, ELMo-based, and BERT-based models. Our models' illustration is shown in Fig. 1. Our models all have a hierarchical nature: each computes word-level representations, then sentence-level, then essay-level successively. The following paragraphs describe each sub-level in details.

Note that for BERT-based model, the whole transformer architecture is used as an end-to-end AES model as suggested by [8]. We pass a sequence of word tokens to the pre-trained BERT model [23]. We use the model named BERT-Base, which, before connected to an output layer, has 12 layers (transformer blocks) of hidden size of 768 and has 12 self-attention heads. The implementation of our models is available on Github[1].

A. Word-level

For the GloVe-based model, word tokens $[w_1, w_2, w_3, ..., w_n]$ of each sentence s are passed into a GloVe pre-trained lookup layer to acquire word vectors $[\mathbf{w}_1, \mathbf{w}_2, \mathbf{w}_3, ..., \mathbf{w}_n]$. These word vectors are fed into a convolutional layer followed by attention-pooling to acquire the sentence representation $\mathbf{s}$.

For the ELMo-based model, as described by [7], word tokens w_i are passed into convolutional layers followed by highway layers to acquire the intermediate word representations $\mathbf{x}_i$. These representations are passed into the main two-layer language model in both directions known as Bidirectional Language Model (BiLM) to generate a hidden state $\mathbf{h}_i^{forw}$ for forward direction and $\mathbf{h}_i^{back}$ for backward direction. The $\mathbf{x}_i$ are then weight-summed with the two hidden states to acquire the context-sensitive word embedding $\mathbf{w}_i$.

[1] https://github.com/sunnypwang/aes

Fixed mean-pooling across all words determines the sentence representation $\mathbf{s}$. We used the pre-trained BiLM module from Tensorflow Hub, which is trained on the 1 Billion Word Benchmark [2].

B. Sentence-level

Sequences of sentence vectors $[\mathbf{s}_1,\mathbf{s}_2,\mathbf{s}_3,...,\mathbf{s}_n]$ are passed into a single-layer forward LSTM with attention-pooling to acquire the final representation. Each sentence vector $\mathbf{s}_i$ is passed through an LSTM unit to generate a hidden state $\mathbf{h}_i$. The hidden states from every time step are then passed through the self-attention layer in the method described by [3]. The attention mechanism helps to indicate how each sentence contributes to the final score. A sentence with high attention weight has more responsibility in determining the final score than a sentence with low attention weight.

The attention weights are calculated as follows: Let $\mathbf{H}$ be a matrix in which rows are composed of hidden state vectors, i.e. $\mathbf{H} = [\mathbf{h}_1; \mathbf{h}_2; \mathbf{h}_3; ...; \mathbf{h}_n]$. Let $\mathbf{W}_a$ and $\mathbf{w}_a$ be the trainable weight matrix and the vector, respectively, and $\mathbf{b}_a$ be a bias vector. The attention weight vector $\mathbf{a} = [a_1, a_2, ..., a_n]$, which is composed of attention weights for each sentence, is calculated accordingly:

$$\hat{\mathbf{A}} = tanh(\mathbf{H}\mathbf{W}_a + \mathbf{b}_a) \tag{1}$$

$$\mathbf{a} = softmax(\hat{\mathbf{A}}\mathbf{w}_a) \tag{2}$$

$\hat{\mathbf{A}}$ is an intermediate matrix. Note that each attention weight is always positive and summed to 1, due to the softmax function. Each hidden state is then multiplied with the corresponding attention weight and summed to produce the final essay representation $\mathbf{e}$. This is written as:

$$\mathbf{e} = \mathbf{a}^T\mathbf{H} \tag{3}$$

C. Essay-level

The final representation $\mathbf{e}$ is passed through a dense layer with a sigmoid activation function to obtain the score $\hat{y}$. In the case of the BERT-based model, the first transformer block (corresponding to the first input token) of the last hidden layer is connected to a dense layer. This score is converted back into an integer score $\hat{y}_{int}$ whose range is defined by each prompt as shown in Table I.

D. Freezing weights

We also report the results of each model when freezing the weights that correspond to the embedding mechanism. For the GloVe-based model, the GloVe pre-trained lookup layer is fixed. For the ELMo-based model, the biLM weights are always fixed, so there is no fine-tuned variant. For the BERT-based model, all transformer layers are fixed, which means that all of the model is fixed except the last regression layer.

IV. Experiment Setup

In this section we discuss the dataset, methodology, and the training parameters used in the experiment.

TABLE I
ASAP Dataset Description

prompt	#essays	avg sentences	score	mode
1	1783	23	2-12	8
2	1800	20	1-6	4
3	1726	6	0-3	2
4	1772	5	0-3	1
5	1805	6	0-4	2
6	1800	7	0-4	3
7	1569	11	0-30	16
8	723	34	0-60	40

A. Dataset

We used the Automated Student Assessment Prize (ASAP) dataset from the Hewlett Foundation, available on Kaggle[3]. This dataset consists of essays written by students in grades seven to ten in 8 different prompts, as described in Table I.

This dataset does not contain a labeled test set; therefore, the labeled data was split into training, validation, and test set with a ratio of 80/10/10, respectively

We used Quadratic Weighted Kappa (QWK)[4] as an evaluation metric to measure agreement between raters. The possible range of QWK is from -1 (complete disagreement) to 1 (complete agreement). QWK of 0 represents a random agreement.

The calculation of QWK requires a weight matrix (or penalty matrix) W, which specifies how much penalty value is given for each pair of score label i by the human rater and j by the AES. The score label ranges from 0 to $N-1$, where N is the number of all possible score labels. The calculation of W is defined in Equation 4. Note that the diagonal values has zero penalty, while the further in score difference yields the higher penalty value.

$$W_{i,j} = \frac{(i-j)^2}{(N-1)^2} \tag{4}$$

The QWK, κ, is then defined as shown in Equation 5.

$$\kappa = 1 - \frac{\sum W \circ O}{\sum W \circ E} \tag{5}$$

O is a confusion matrix of scores between two raters with $O_{i,j}$ being the number of essays that receive a score label i by the human raters and j by the AES. E is an expected value matrix of scores between two raters with $E_{i,j}$ being the product of the number of essays that receive a score label i by the human raters and score j by the AES. Both O and E are normalized in such that the sum of all elements in each matrix equals 1. These matrices are then multiplied element-wise (denoted by ∘ symbol) with W and sum over all elements to yield the weighted kappa value κ.

B. Training

We pre-process the text by converting all characters to lower case, by removing hyperlink texts, and by removing special characters. Each essay is sentence-tokenized via regular expression and word-tokenized using NLTK's TreebankWordTokenizer module[5]. In the case of the BERT-based

[2] https://tfhub.dev/google/elmo/2

[3] https://www.kaggle.com/c/asap-aes

[4] We used scikit-learn's implementation

[5] https://www.nltk.org/_modules/nltk/tokenize/treebank.html

TABLE II
HYPERPARAMETERS USED IN THE TRAINING PROCESS

	GloVe	ELMo	BERT
LSTM Unit	100	100	-
dropout	0.5	0.3	0.1
learning rate	1e-3	1e-3	5e-5
optimizer	rmsprop	rmsprop	adam
batch size	10	10	4
epoch	50	10	3

model, each essay is tokenized into a sequence of words (or, in some case, sub-words) using WordPiece tokenizer [24] as described in the BERT paper [8].

Mean Square Error (MSE) is used as a loss function. The best model is chosen by the lowest MSE on the validation set. The hyperparameters for each model are listed in Table II.

We also compare our models' results to the Neural Essay Assessor (NEA), a public AES system developed by K. Taghipour and H. T. Ng [1]. NEA uses pre-trained word embeddings released by W. Y. Zou, R. Socher, D. Cer, and C. D. Manning [25]. These embeddings were fed to a convolutional layer and LSTM units with mean-over-time layer to produce a score. We used this system as our baseline model.

V. ROBUSTNESS

We used robustness as our main evaluation metric. Corrupting an original essay to a certain degree produces a less legible essay, which should result in a lower score. The model is considered to have good robustness if an altered essay is given a lower score than the original essay.

In our experiment, we corrupted every essay in the test set, creating an altered test set. We created a total of 8 altered sets, each having a different alteration method. Each alteration method corrupts a different aspect of an essay. A list of all altered sets is described in Table III. For clarity, the unaltered test set is called the "original set" from here on.

We define robustness as follows: for any altered set m of prompt p, the model is tasked to score essays from both the original set and the altered set. If an altered essay receives the lower score than an original essay, we count the altered essay as a *worse*-type essay. If an altered essay receives the higher score, then it counts as a *better*-type essay. Finally, we define robustness r_m^p as the fractional difference between the total number of *worse* and *better* essays:

$$r_m^p = \frac{1}{n^p}(w_m^p - b_m^p). \tag{6}$$

Here, w_m^p denotes the total number of worse essays from set m of prompt p, b_m^p is the total number of better essays, and n^p is the number of essays in prompt p. Thus, r_m^p can range from -1 (anti-robust) to 1 (highly robust). Note that w_m^p and b_m^p do not include the case that there is no score difference between the original and altered essays, so $w_m^p + b_m^p \leq n^p$.

The model with the highest average robustness of all prompts and altered sets is considered to be the most preferred model for our task.

TABLE III
ALTERED SETS

Alteration	Description
no_art	remove articles (a, an, the)
no_conj	remove conjunctions (and, or, but)
add_and	add "and" randomly to the start of a sentence (10%)
swap_word	swap words in a same sentence randomly (5%)
no_first_sent	remove the first sentence
no_last_sent	remove the last sentence
no_longest_sent	remove the longest sentence
reverse_sent	reverse sentence order

TABLE IV
QUADRATIC WEIGHTED KAPPA

prompt	GloVe	GloVe-fw	ELMo	BERT	BERT-fw	NEA
1	0.850	0.825	0.816	0.834	0.825	0.612
2	0.717	0.707	0.675	0.725	0.679	0.473
3	0.695	0.617	0.658	0.731	0.695	0.691
4	0.828	0.788	0.815	0.830	0.720	0.827
5	0.806	0.777	0.810	0.811	0.759	0.777
6	0.800	0.806	0.793	0.757	0.731	0.733
7	0.792	0.774	0.823	0.842	0.809	0.776
8	0.716	0.654	0.681	0.671	0.709	0.614
avg	0.775	0.744	0.759	0.775	0.741	0.688

The *robustness for each prompt* r^p is defined as an average of robustness of all altered sets, while the *robustness for each altered set* r_m is an average of robustness of all prompts.

Note that our definition of robustness is different than that of most other tasks, where robustness describes an ability to map noisy input to the same output class.

VI. RESULTS

For each different embedding model, with frozen and unfrozen weights in their embedding layers, we compared the robustness and QWK to the NEA baseline model. The QWK is shown in Table IV. The robustness of each prompt is shown in Table V. The robustness of each altered set is shown in Table VI. Models with the "-fw" suffix indicate frozen weights. As mentioned in Section III-D, the ELMo-based model's embedding weights are always frozen.

a) Robustness: The BERT-based model achieved higher robustness than the GloVe-based and ELMo-based models, with the fixed-weight variant attaining the highest robustness. The ELMo-based model, despite using bidirectional language models, achieved lower robustness than the GloVe-based model.

b) Quadratic Weighted Kappa: All models achieved state-of-the-art QWK, which are at the level of human performance. Our models' QWK values are in the range of 0.741 to 0.775, while the QWK between the two human raters is 0.754 [1].

A. Analysis

We observed that the models have higher robustness on sentence-level alterations than word-level alterations, as r_m for the *no_first_sent*, *no_last_sent*, *no_longest_sent*, and *reverse_sent* are significantly higher in Table VI. Among these sets, removing the longest sentence yields the highest robustness, suggesting that the longest sentence contributes the most to the essay. We support this claim by looking into the

TABLE V
ROBUSTNESS FOR EACH PROMPT (r^p)

prompt	GloVe	GloVe-fw	ELMo	BERT	BERT-fw	NEA
1	0.171	0.124	0.105	0.115	0.115	-0.006
2	0.049	0.069	0.022	0.085	0.113	-0.003
3	0.122	0.129	0.046	0.116	0.127	0.087
4	0.127	0.129	0.111	0.131	0.182	0.073
5	0.110	0.129	0.086	0.144	0.144	0.104
6	0.116	0.122	0.117	0.151	0.251	0.018
7	0.244	0.313	0.283	0.369	0.408	0.154
8	0.361	0.344	0.284	0.399	0.394	-0.140
avg	0.163	0.170	0.132	0.189	0.217	0.036

TABLE VI
ROBUSTNESS FOR EACH ALTERED SET (r_m)

alteration	GloVe	GloVe-fw	ELMo	BERT	BERT-fw	NEA
no_art	0.236	0.232	0.110	0.320	0.412	0.008
no_conj	0.167	0.179	0.067	0.185	0.190	-0.096
add_and	-0.007	-0.012	0.005	0.012	0.012	0.019
swap_word	-0.007	0.002	0.074	0.259	0.274	-0.020
no_first_sent	0.255	0.263	0.195	0.196	0.242	0.033
no_last_sent	0.235	0.245	0.215	0.183	0.191	0.172
no_longest_sent	0.403	0.415	0.325	0.339	0.374	0.219
reverse_sent	0.020	0.035	0.064	0.017	0.038	-0.049
avg	0.163	0.170	0.132	0.189	0.217	0.036

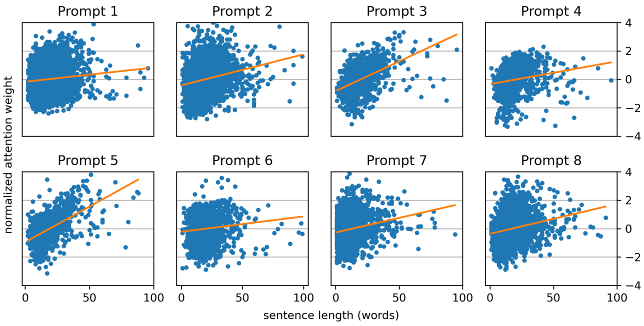

Fig. 2. Attention weights are generally higher for longer sentences.

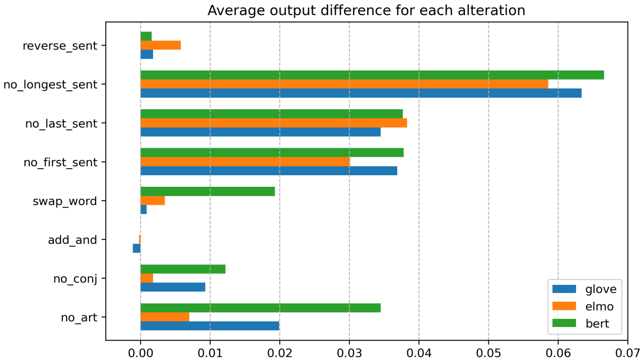

Fig. 3. Average output difference for each alteration

attention weights given to each sentence in respective models and finding that longer sentences receive more attention than shorter ones. The relationship between sentence length and attention weight is shown in Fig. 2.

The test results revealed that, on average, freezing weights in the embedding layer responsible for text embedding in the GloVe-based and BERT-based models result in a lower QWK but higher robustness. We ascribe this tradeoff to the fact that models' embedding layers are mostly trained on well-written passages, and thus are more sensitive to errors presented in the dataset. We found that after fine-tuning, the models became less sensitive to errors. In particular, they often yielded higher scores, closer to human ratings, resulting in higher QWK. However, at the same time, the fine-tuned models are less able to differentiate altered essays from the non-altered ones, often giving both essays the same high scores. As a result, fine-tuned models (GloVe, BERT) have lower robustness than their counterparts (GloVe-fw, BERT-fw).

The ELMo-based model has the lowest robustness among all three embedding techniques. Fig. 3 shows the difference between an original essay score $\hat{y}_{org}$ and its altered counterpart score $\hat{y}_{aug}$, averaged across all prompts, split by each embedding technique. The ELMo-based model (orange) has the lowest output difference in the majority of alteration types, suggesting a particular insensitivity to changes in the essays.

We hypothesize that the ELMo-based model's predictions change less because the changes in lower-layer features are too subtle. We looked into the lowest-layer feature sentence vectors s_i of the GloVe-based and ELMo-based models[6] and found that in word-level altered sets (*no_art*, *no_conj*, *add_and*, *swap_word*), the cosine distances between the original and altered sentence vectors are generally higher in the GloVe-based model than in the ELMo-based model, most notably in shorter sentences, suggesting that our alterations affected GloVe's embedding more than ELMo's. We show selected sentences along with their cosine distances from each model in Table VII.

B. Discussion

Regarding attention-based AES models, we believed that, despite promising performance, there are vectors of potential abuse that can undermine the integrity of AES systems.

First, we found that the models pay more attention to longer sentences. A submitter could thus trick the AES into giving a better score by writing the main point in a long sentence while disregarding short supporting sentences. AES developers should be cognizant of this particular type of adversarial essay.

Second, we observed that adding non-important but common words to an essay unexpectedly improves its score on average. As shown in Fig. 3, *add-and* is the only altered set with negative average output difference, meaning that there are more *better* than *worse* altered essays. Since adding "and" to the beginning of a sentence is often grammatically and semantically correct, this alteration could confuse the model into assigning more credence.

VII. Conclusion

We compared the performance of models using three different text embedding techniques on the task of Automated Essay Scoring (AES). Our models achieved QWK on par with human performance. We also introduced a

[6] We omit the BERT-based model because it is not explicitly subdivided into a hierarchical structure

TABLE VII
Cosine distances between the original and altered sentence vectors by the GloVe-based and ELMo-based models. The original text is shown to the right. Underlined words are removed or changed after alteration.

	Cosine distance		
alteration	GloVe	ELMo	sentence
no_conj	0.431	0.071	stop reading this article and go out and meet new friends
no_art	0.173	0.065	many things in the setting affect the cyclist
no_art	0.679	0.017	it really was a good experience we were all laughing telling jokes all the time and since then we tell jokes in the family table

new evaluation metric, robustness, which tests the models' performance against adversarial inputs, and ranked the three text embedding techniques by their models' robustness as follows: BERT > GloVe > ELMo. This suggests that BERT is the most suitable choice, among the three models, for an AES system in which adversarial exploits are a matter of concern.

References

[1] K. Taghipour and H. T. Ng, "A neural approach to automated essay scoring," in *Proceedings of the 2016 conference on empirical methods in natural language processing*, 2016, pp. 1882–1891.

[2] F. Dong and Y. Zhang, "Automatic features for essay scoring–an empirical study," in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016, pp. 1072–1077.

[3] F. Dong, Y. Zhang, and J. Yang, "Attention-based recurrent convolutional neural network for automatic essay scoring," in *Proceedings of the 21st Conference on Computational Natural Language Learning*, 2017, pp. 153–162.

[4] Y. Farag, H. Yannakoudakis, and T. Briscoe, "Neural automated essay scoring and coherence modeling for adversarially crafted input," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 263–271.

[5] J. Liu, Y. Xu, and Y. Zhu, "Automated essay scoring based on two-stage learning," *arXiv*, pp. arXiv–1901, 2019.

[6] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing*, 2014, pp. 1532–1543.

[7] M. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 2227–2237.

[8] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 4171–4186.

[9] J. Cohen, "A coefficient of agreement for nominal scales," *Educational and psychological measurement*, vol. 20, no. 1, pp. 37–46, 1960.

[10] P. W. Foltz, D. Laham, and T. K. Landauer, "Automated essay scoring: Applications to educational technology," in *EdMedia+ innovate learning*. Association for the Advancement of Computing in Education, 1999, pp. 939–944.

[11] Y. Attali and J. Burstein, "Automated essay scoring with e-rater® v. 2," *The Journal of Technology, Learning and Assessment*, vol. 4, no. 3, 2006.

[12] L. S. Larkey, "Automatic essay grading using text categorization techniques," in *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, 1998, pp. 90–95.

[13] H. Yannakoudakis, T. Briscoe, and B. Medlock, "A new dataset and method for automatically grading esol texts," in *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 2011, pp. 180–189.

[14] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] Y. LeCun, P. Haffner, L. Bottou, and Y. Bengio, "Object recognition with gradient-based learning," in *Shape, Contour and Grouping in Computer Vision*. Springer, 1999, pp. 319–345.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations*, 2015.

[17] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, "A neural probabilistic language model," *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003.

[18] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *Computing Research Repository*, vol. abs/1301.3781, 2013.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.*, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[20] D. E. Powers, J. C. Burstein, M. Chodorow, M. E. Fowles, and K. Kukich, "Stumping e-rater: challenging the validity of automated essay scoring," *Computers in Human Behavior*, vol. 18, no. 2, pp. 103–134, 2002.

[21] R. Collobert, "Deep learning for efficient discriminative parsing," in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 2011, pp. 224–232.

[22] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu *et al.*, "Natural language processing (almost) from scratch," *Journal of machine learning research*, vol. 12, pp. 2493–2537, 2011.

[23] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue *et al.*, "Huggingface's transformers: State-of-the-art natural language processing," *arXiv*, vol. abs/1910.03771, 2019.

[24] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi *et al.*, "Google's neural machine translation system: Bridging the gap between human and machine translation," *arXiv*, pp. arXiv–1609, 2016.

[25] W. Y. Zou, R. Socher, D. Cer, and C. D. Manning, "Bilingual word embeddings for phrase-based machine translation," in *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013, pp. 1393–1398.