Integrating Deep Learning into An Automated Feedback Generation System for Automated Essay Scoring

Chang Lu, Maria Cutumisu
Department of Educational Psychology
Faculty of Education, University of Alberta
{clu4, cutumisu}@ualberta.ca

ABSTRACT

Digitalization and automation of test administration, score reporting, and feedback provision have the potential to benefit large-scale and formative assessments. Many studies on automated essay scoring (AES) and feedback generation systems were published in the last decade, but few connected AES and feedback generation within a unified framework. Recent advancements in machine learning algorithms enable researchers to develop more models that explore the potential of automated assessments in education. This study makes the following contributions. First, it implements, compares, and contrasts three AES algorithms with word-embedding and deep learning models (CNN, LSTM, and Bi-LSTM). Second, it proposes a novel automated feedback generation algorithm based on the Constrained Metropolis-Hastings Sampling (CGMH). Third, it builds a classifier to integrate AES and feedback generation into a systematic framework. Results show that (1) the scoring accuracy of the AES algorithm outperforms that of state-of-the-art models; and (2) the CGMH method generates semantically-related feedback sentences. The findings support the feasibility of an automated system that combines essay scoring with feedback generation. Implications may lead to the development of models that reveal linguistic features, while achieving high scoring accuracy, as well as to the creation of feedback corpora to generate more semantically-related and sentiment-appropriate feedback.

Keywords

Automated essay scoring, deep learning, feedback generation, assessment, machine learning, natural language processing

1. INTRODUCTION

Automatic essay scoring (AES), the task of machine-grading essays or constructed-response items, has been gaining attention due to technology-powered advances in educational assessment [16]. The goal of AES is to produce reliable and valid scores using machine scoring rather than human scoring [43]. Previous research has made advances in automatic grading essays with handcrafted features [16, 30, 40]. Currently, with the availability of large volumes of

trainable corpora extracted online and the development of models for word representation in the Natural Language Processing (NLP), deep learning approaches have produced highly reliable scores using text classification methods [10, 12, 22]. However, few studies have approached automated essay scoring and automated feedback generation to achieve a fully automated computer-based testing system (CBT).

Earlier attempts at implementing feedback have been made using real-time online tutoring by humans [18, 31]. Findings show that human tutoring is effective at improving students' performance, but it is time consuming and labor intensive. Also, human tutoring is not applicable to large-scale practice and open-ended platforms with large numbers of students. Research on automated feedback generation emerged in the last decade to fill this gap by developing tools to scaffold students within computer-based testing environments [24, 36]. Previous studies have focused on generating formative feedback using rule-based approaches [3, 38]. Although rule-based feedback generation is relatively easy to achieve and the generated sentences can be considered to be appropriate feedback, this approach is usually restricted to pre-designed templates. Recent efforts have been made to engage students in more communicative and adaptive environments and to propose feedback-generation frameworks using sentence generation with constraints, where the constraints are often defined by domain-specific terms [8, 11]. Nevertheless, few studies have empirically examined automated language generation in CBTs. This study proposes a framework that introduces an algorithm based on deep learning models with an unsupervised sentence-generation approach to automatically grade essays and to generate feedback.

2. RELATED WORK

2.1 Automated Essay Scoring

Automated essay scoring constitutes the task of automatically assigning scores to written essays based on features or characteristics in the text. Several systems for Automated Essay Scoring (AES) have already been developed and used in large-scale high-stakes assessments for several decades. Page [33] designed the first intelligent scoring system, Project Essay Grade (PEG), using simple linear regression with hand-crafted features such as essay length and proposition counts to perform text classification tasks based on these features. Since then, other systems for automated essay scoring emerged such as Intelligent Essay Assessor [25], e-rater [6], IntelliMetric [41], and My Access! [41]. Several AES methods have been later adopted to make predictions on student writing scores. Yannakoudakis et al. [44] approached AES as a preference-ranking problem and evaluated essays based on pairwise comparisons of features, such as POS n-grams features and complex grammatical features. Gierl et al. [16] demonstrated the application of AES in medical exams with Support Vector Machine (SVM). Phandi et al. [34] approaches AES with Bayesian Linear Ridge Regression. Taghipour and Ng [40] designed an 'Enhanced AI Scoring Engine' (EASE) based on four genres of

features: length-based features, Parts-of-Speech (POS), word-prompt overlap, and bag of n-grams. These features were fed into several model architectures such as Convolutional Neural Network (CNN) and the Recurrent Neural Network (RNN)-variants, namely Long Short-Term Memory (LSTM) and Bidirectional LSTM (Bi-LSTM), to compare the prediction performance of the models. Phandi et al.'s [34] attempt to employ deep learning models to predict essay scores was later used as a baseline for related studies. Previous studies on automated essay scoring rely heavily on hand-crafted feature engineering and knowledge on linguistic discourse. Inspired by recent advances of deep learning models and word embedding techniques, a substantial body of literature has emerged, which has contributed to applying deep learning methods in automated essay scoring tasks. Alikaniotis et al. [1] implemented a two-layer bidirectional LSTM with score-specific word embeddings to learn essay representations and conduct AES tasks. The proposed model outperformed the baseline SVM model. Later, Dong and Zhang [9] established a three-layer model architecture combining CNN for character representation and LSTM for sentence representation with an extra attention-pooling layer, which performed better than Taghipour and Ng's [40] model and their two-layer CNN model. Taghipour and Ng's attempt of combining feature engineering and deep learning models inspired later trials of applying word embeddings and deep learning methods on AES.

2.2 Automated Feedback Generation

Providing feedback is a key ingredient in performance improvement. In education, feedback is defined as the information provided by an agent regarding aspects of one's performance or understanding [17]. High-quality personalized and timely feedback can improve learners' performance [17], but feedback provision is often reported as the long-standing weakness of ITSs and computer-based assessment systems [27]. On the one hand, students complain that they receive too little quality feedback in the process of learning [5, 13]. On the other hand, students are reported to misuse and abuse the feedback or hints provided by the ITSs [37]. Thus, knowing how and when to provide real-time personalized feedback that guides and motivates students' learning remains a challenge.

Williams and Dreher [42] advocated the potential of fully-automated systems that perform both scoring and feedback provision with machines in tasks such as essay grading. Previous efforts have been made to produce feedback in intelligent tutoring or assessment systems [7, 19, 21] for various disciplines, such as computer science [14, 23], information and communication technology (ICT [7]), and English as a Second Language (ESL [26]). However, most automated assessment systems adopt a template-based method to generate feedback [4, 26, 42], which usually produces feedback that is limited to fixed expressions.

Recent advances on constrained sentence generation shed some light onto flexible feedback generation. For example, Su et al. [39] proposes a Gibbs sampling method to meet the constraints of sentiment control. However, Gibbs sampling is not able to vary the sentence length or handle keywords when generating the sentence. Miao et al. [32] extends the Gibbs sampling to a novel unsupervised sampling approach, named Constrained Generation by Metropolis-Hastings sampling (CGMH). The CGMH is a subtype of the Markov Chain Monte Carlo (MCMC [15]) methods. The CGMH allows for more flexible operations on word tokens in a sentence space, thus it is easier to generate content with constraints and varying sentence lengths. Miao et al. [32] tested the CGMH on three tasks including key-to-sentence generation with hard constraints, paraphrase, and error correction with soft constraints. The CGMH method outperformed state-of-art sentence-generation algorithms. Yet, one of the reasons that the research on automated feedback generation with NLP is lagging behind may be that there is no publicly-available feedback corpus.

2.3 Present Study

We propose an AES and an automated feedback generation framework to support students' performance. Specifically, the current study implements three deep learning models for automated essay scoring: (1) CNN; (2) CNN and LSTM; and (3) CNN and Bi-LSTM. In addition, a novel unsupervised sentence-generation approach uses CGMH to automatically provide feedback for test takers based on their predicted essay scores. The remaining sections are guided by the following research questions:

1. To what extent can the AES algorithms generate accurate performance on essay scoring?

2. To what extent can the CGMH algorithms generate fluent and semantically-related feedback?

The contributions of the present study are three-fold. First, the study advances computer-based testing by incorporating automated feedback generation into the assessment framework, especially for unstructured text (e.g., essays). Second, the flexible unsupervised learning approach creates a corpus of semantically-related and sentiment-appropriate feedback for scaffolding. Third, the scalable automatic assessment and feedback provision system is automated and performs accurately, which paves the way for future implementations of feedback generation for various domains within intelligent tutoring systems.

3. METHOD

3.1 Datasets and Corpus

The dataset for the AES task was retrieved from a Kaggle challenge named Automated Student Assessment Prize (ASAP) sponsored by the Hewlett Foundation in 2012 and detailed in Table 1.

Table 1. Summary of ASAP dataset

Prompt	Genre	Grade Level	Training set size	Score Range	Ave Length
1	persuasive /narrative / expository	8	1783	2-12	350
2	persuasive /narrative / expository	10	1800	1-6	350
3	source dependent	10	1726	0-3	350
4	source dependent	10	1772	0-3	150
5	source dependent	8	1805	0-4	150
6	source dependent	10	1800	0-4	150
7	persuasive /narrative/ expository	7	1569	0-30	250
8	persuasive /narrative / expository	10	723	0-60	650

The ASAP is the benchmark dataset for piloting AES studies. It consists of 8 prompts and 4 genres, including persuasive, narrative,

expository, and source-dependent responses. In total, 12,979 essays were released. Since the ASAP has not made the official test sets publicly available, we used 60% of the training set for training, 20% for validation, and 20% for testing. We first performed text cleaning, tokenization, and padding. Then, we used Stanford's publicly-available GloVe 300-dimensional model to conduct word embeddings [35]. The GloVe 300-dimensional embeddings were trained on 6 billion words scraped from Wikipedia and other web texts. The writing prompts have different score ranges as shown in Table 1. To address the issue of inconsistent score ranges, we followed Phandi et al.'s [34], Taghipour and Ng's [40], and Dong and Zhang's [9] method by approaching the AES as a regression task, rescaling the essay score to [0, 1] in the training, validating, and test stages, and projecting the scores back to their original scales in the evaluation stage.

The corpus used to train language models for sentence generation consisted of the publicly-available IMDB dataset, which contains 25,000 positive reviews and 25,000 negative reviews. The dataset was split into three parts: the training set consisted of 20,000 negative reviews and 20,000 positive reviews, the validation test consisted of 1,250 negative and 1,250 positive reviews, and the test set consisted of 1,250 negative and 1,250 positive reviews. A third-party corpus, the Reuters corpus from NLTK, was used for evaluation of the quality of the generated sentences.

3.1.1 AES Step

The present study was conducted in two steps. Step 1 addressed automated essay scoring, whereas Step 2 addressed automated feedback generation. An essay performance classifier was added to synthesize the two steps into a unified framework.

In the AES task, three deep-learning algorithms were implemented and compared regarding their performance and efficiency to select the optimal algorithm as the foundation of the feedback generation step: CNN, CNN + LSTM, and CNN + Bi-LSTM. The convolutional layer is seen as a function that could learn features from n-grams, and can be represented as:

$$Z_i = f\left(W_z\left[x_i^j : x_i^{j+h_w-1}\right] + b_z\right),$$

where x_i is the ith embedded word, W_z is the weight matrix, b_z is the bias vector, h_w is the window size of the convolutional layer, f is a non-linear activation function (i.e., sigmoid, tanh, or ReLu), and Z_i is the output of feature representation.

LSTM is an RNN model for processing sequence data [20]. The unit or memory cell of LSTM consists of an input gate, a forget gate, and an output gate to control information flow. The gates decide preserving, forgetting, and passing information as a vector sequence at each time step.

More specifically, assuming there are *T* sentences in an essay in total, the composite functions at sentence *t* can be written as:

$$i_t = \sigma(W_i s_t + U_i h_{t-1} + b_i),$$
$$f_t = \sigma(W_f s_t + U_f h_{t-1} + b_f),$$
$$g_t = tanh(W_g s_t + U_g h_{t-1} + b_g),$$
$$c_t = i_t \odot g_t + f_t c_{t-1},$$
$$O_t = \sigma(W_o s_t + U_o h_{t-1} + b_o),$$
$$h_t = O_t \odot \tanh(c_t),$$

in which s_t is the input vector, h_t is the output vector, W_i, W_f, W_g, W_o, U_i, U_f, U_g, U_o are the estimated weight matrices, and b_i, b_f, b_g, b_o are the bias vectors.

Bi-LSTM is an extension of unidirectional-LSTM for deeper representations. Compared with unidirectional-LSTM that can only preserve and pass information from history, Bi-LSTM can also make use of information from future. In AES tasks, Bi-LSTM could process the words in the input vector in both a forward and a backward manner. The composite function for Bi-LSTM is similar with LSTM:

$$y_t = W_{yh}\begin{pmatrix}\overrightarrow{h_t} \\ \overleftarrow{h_t}\end{pmatrix} + b_y.$$

The summary of the model architectures for the three models is shown in Table 2.

Table 2. Model architecture summary

Layer	Hyperparameter	Value
CNN		
Embeddings	dimension	300
Convolutional	filters, kernel size	100, 5
CNN + LSTM		
Embeddings	dimension	300
Convolutional	filters, kernel size	100, 5
LSTM	units	32
CNN + Bi-LSTM		
Embeddings	dimension	300
Convolutional	filters, kernel size	100, 5
Bi-LSTM	layers	16

3.1.2 Feedback Generation Step

The feedback generation phase included two steps. In Step 1 (Corpus Development), we will develop a corpus of feedback using CGMH based on the expert-derived essay descriptors. In Step 2 (Feedback Generation & Provision), we will develop feedback based on the essay scores provided by the AES algorithms.

Table 3 shows the part of the essay-scoring rubrics (the score ranged from 1 to 6) and descriptors developed by experts.

Table 3. Sample descriptor for Essay Prompt 1

Score	Descriptors
1	An undeveloped response that may take a position but offers no more than very minimal support.
Element	Contains few or vague details.
	Is awkward and fragmented.
	May be difficult to read and understand.
	May show no awareness of audience.
2	An under-developed response that may or may not take a position.
Element	Contains only general reasons with unelaborated and/or list-like details.
	Shows little or no evidence of organization.

	May be awkward and confused or simplistic.
	May show little awareness of audience.

To expand the corpus, we will adopt the *Constrained Sentence Generation by Metropolis-Hastings Sampling* method (CGMH [32]) to perform unsupervised paraphrase generation. The CGMH facilitates the generation of content with constraints and varying sentence lengths. Miao et al. [32] tested the CGMH on three tasks, including keywords-to-sentence generation with hard constraints, paraphrase, and error correction with soft constraints. In the present research, we will implement unsupervised paraphrasing to augment the feedback corpus. Specifically, we will first train a language model based on the IMDB review corpus [29]. The IMDB dataset consists of 25,000 positive and 25,000 negative movie reviews. It was selected for the feedback-generation task for the following reasons. First, to date, there is no database of academic feedback, the IMDB was the closest commentary corpus available. More importantly, this corpus is split into positive and negative phrases, which makes it domain-independent. Thus, it can transfer more easily to other domains. Then, we will perform the paraphrase generation.

A Markov model is used to train the language model on the selected corpus. The Markov Chain is commonly used to model natural language as a function of the probability that a word appearing in position *n* is only dependent on the previous *z* ϵ [1, n-1] such that:

$$p(w_1, w_2, \dots, w_n) = p(w_1)\, p(w_2|w_1), \dots\dots, p(w_n|w_{n-z}, \dots, w_{n-1}),$$

where $p(w_1, w_2, \dots, w_n)$ refers to the probability of a specific sentence based on the trained corpus, that is, the joint probability of all words within the sentence. In the present research, we used forward-backward dynamic programming to train the language model.

In Step 2 (feedback paraphrase), we performed the CGMH task of unsupervised sentence paraphrasing. The CGMH is concerned with a goal of stationary distribution that defines the sentence distribution sampled from the corpus and three actions, namely, replacement, insertion, and deletion. Specifically, $\pi(x)$ was set as the distribution from which we plan to sample sentences, where *x* denotes a particular sentence and x_0 refers to the feedback template that is fed to the algorithm at time step 0. The MH sampler either accepts or rejects a word from the given distributions $\pi(x)$ to finally form a desired joint distribution of all words based on a predefined stationary distribution. The process is intuitive, as it mainly involves two actions: accepting or rejecting a word monitored by the acceptance rate α:

$$\alpha = \min\{1, \frac{\pi(x')g(x_{t-1}|x')}{\pi(x_{t-1})g(x'|x_{t-1})}\}$$

At time step *t*, the word sampling is conducted to update the previous state *x* to a candidate distribution x' from a proposed distribution $g(x'|x_{t-1})$, where x_{t-1} refers to the distribution from previous step (*t-1*), thus $x' = x_t$. Therefore, α determines the acceptance or rejection of a sample. In our paraphrase generation, the desired distribution denotes the most likely and logical sentence related to the original sentence.

At each step, a selected word in the sentence will be updated by the actions such as insertion, deletion, and replacement, randomly, where the respective probabilities are $[p_{insert}, p_{delete}, p_{replace}]$. At the first time step, these probabilities are set as being equal. At the following step, if *Replacement* is applied on a selected word w_m in a sentence $x = [w_1, w_2, \dots, w_{m-1}, w_m, w_{m+1}, \dots, w_n]$, then the conditional probability of choosing w_m^{new} to replace w_m to form candidate sentence x' from *x* can be computed as:

$$g_{replace}(x'|x) = \pi(w_m^{new}|x_{-m}) = \frac{\pi(w_1, w_2, \dots, w_{m-1}, w_m^{new}, w_{m+1}, \dots, w_n)}{\sum_{w \in V}(w_1, w_2, \dots, w_{m-1}, w, w_{m+1}, \dots, w_n)},$$

where *V* refers to the vocabulary, and w_m is the selected word. If, on the other hand, *Insertion* is applied, an additional step of inserting a placeholder will be conducted before taking the action *Replacement*, and then a real word will be sampled to replace the placeholder token with the *Replacement* token. Finally, if *Deletion* is applied, the w_m word selected will be deleted, and $g_{deletion}(x'|x)$ =1 if $x' = [w_1, w_2, \dots, w_{m-1}, w_{m+1}, \dots, w_n]$, and 0 otherwise. The detailed settings of the sentence-generation phase, including the hyperparameter values determined in the tuning process are included in Table 4.

Table 4. MCMC hyperparameter

Hyperparameters	**Value**
Dictionary size	50,000
Hidden nodes per LSTM layer	300
Number of steps	50
Maximum sentence length	50
Max epoch	30
Minimum of Sentence Length	7
Initial action probability	[0.3, 0.3, 0.3, 0.1]

3.1.3 Synthesis

One important purpose of the present study is to develop a framework linking automated essay scoring and automated feedback generation. Thus, the study can be decomposed in two parts: a supervised text classification task using CNN and RNN models and an unsupervised learning paraphrase generation task using MCMC sampling method with constraints. In the synthesis, a performance classifier was applied to extract feedback that corresponds to the score that is assigned by the AES algorithms.

3.2 Evaluation Metrics

The objective of the AES training stage is to minimize the mean squared error (MSE) between the scores provided by human raters and the prediction scores generated by the models.

In the automated essay scoring tasks, several measures including the quadratic weighted kappa (QWK [9, 10]), exact agreement, and alternate-form reliabilities [2] have been used to evaluate the performance of AES models in previous studies. In the current study, we present the results of QWK, which measures the degree of agreement between human raters and the machine on one essay and can be calculated by:

$$\text{QWK} = 1 - \frac{\sum_{i,j} W_{i,j}\, O_{i,j}}{\sum_{i,j} W_{i,j}\, E_{i,j}}.$$

where $W_{i,j}$ is calculated by $W_{i,j} = \frac{(i-j)^2}{(N-1)^2}$ (i: represents the human-rated score; j: represents machine-rated score; N: represents the score range), $O_{i,j}$ represents the number of essays that receive a rating i by the human and a rating j by the machine, and E is the outer product of the histogram vectors of the two scores. According to Williamson, Xi, and Breyer [43], QWK scores higher than 0.7 indicate high accuracy.

For the feedback generation task, we used several measures to evaluate the generated sentences. The first step is concerned with language model training, whereas the second step is concerned with generating sentences with the MCMC sampling method. More specifically, we first reported the training process of the language model over epochs. The objective of the first training process is to minimize the perplexity of the language model, which can be calculated by:

$$PPL = 2^{-\frac{1}{N}\sum_{i=1}^{N} \log\, p(w_i)},$$

where N equals the number of words in the corpus and $p(w_i)$ indicates the probability of a word appearing in the position. The lower the PPL is, the more precisely the corpus is modeled.

For the generated sentences, we evaluated the model performance using two measures. First, we computed the Negative Likelihood (NLL) of the sentences to evaluate their fluency using the Reuters corpus released by NLTK modules. The lower the NLL is, the more fluent the sentences are. Second, we invited two volunteers to rate the quality of 50 pieces of feedback in terms of the sentence fluency and relatedness at a scale of 0-1, and the higher the scores are, the more fluent and related the generated feedback sentences are.

4. RESULTS

4.1 Rating Accuracy of AES Algorithms

The results show that, for Prompt 1, the most accurate algorithm is CNN + Bi-LSTM, whereas for Prompts 2 to 8, the most accurate algorithm is CNN + LSTM. The average QWK of CNN + LSTM reaches 0.734, as shown in Table 5. In general, the models that integrate LSTM/Bi-LSTM perform better than CNN. Compared with the baseline [34], the CNN+LSTM model in the present study performs better on Writing Prompt 1-7, but poorer on Prompt 8. In addition, the average QWK of CNN+LSTM also outperforms the baseline model [34].

Table 5. Comparisons of QWK of the implemented models

Prompt	CNN	CNN + LSTM	CNN + Bi-LSTM	Phandi et al., 2015
1	0.81	0.87	0.88	0.76
2.1	0.62	0.64	0.52	0.61
2.2	0.51	0.61	0.51	-*
3	0.73	0.63	0.62	0.62
4	0.83	0.83	0.72	0.74
5	0.77	0.86	0.76	0.78
6	0.77	0.85	0.8	0.78
7	0.72	0.79	0.75	0.73
8	0.35	0.53	0.54	0.62
Ave	0.68	0.73	0.68	0.71

Note: *indicates that prompts 2.1 and 2.2 were combined into a single score.

Results of the average QWK across genres (e.g., persuasive, narrative, and expository) and source-independent writing can be found in Table 6, which shows that CNN+LSTM outperformed CNN and CNN+Bi-LSTM on both genres. However, the three models all performed poorly on the persuasive, narrative, and expository criteria. The results are consistent with previous studies [34], as the models generally have better predictions on the prompts with smaller score ranges. The wide-score range may cause more complexities for the training process of deep learning models. In addition, previous studies on applying deep neural networks in AES yielded similar results showing that models generally performed poorly on Prompts 2 and 8 [9, 10, 34, 40]. The present study also found that the three deep learning algorithms showed higher efficiency on scoring certain types of genres of writing, but less accuracy on Prompts 2, 3, and 8. One possible explanation is that, for Prompt 2, two domain scores instead of one single global score are provided. The inherent inconsistency or low reliability of a single human rater's scoring makes it difficult for machines to learn the scoring pattern. While for Prompt 8, the score range is 0 to 60, as shown in Table 1. Compared with other prompts whose score ranges are narrow (0 to 3 or 0 to 4), this extremely wide range (i.e., the categories of the outcome variable) may hinder the learning process of deep learning models.

Table 6. Average QWK across genres

QWK	persuasive /narrative/ expository (Prompt 1,2,7,8)	source independent (Prompt 3,4,5,6)
CNN	0.601	0.775
CNN+LSTM	0.688	0.793
CNN+Bi-LSTM	0.640	0.726

4.2 Runtime of AES

Prediction accuracy is of utmost priority in machine learning. However, in a fully-automated scoring and reporting system, scoring efficiency represented as the time it took to run one epoch (i.e., the runtime) also plays an important role. Table 7 shows the average runtime for one epoch of the three models: CNN was the fastest of the three on average. Therefore, it can be concluded that CNN+LSTM has the highest performance, but also has relatively high efficiency (i.e., it is the second fastest algorithm of the three). Thus, it was chosen as the AES algorithm for the feedback-generation step.

Table 7. Average runtime and memory

Model	Runtime for one epoch	N of Parameters
CNN	51s	5117233
CNN + LSTM	53s	4988977
CNN + Bi-LSTM	55s	4986929

4.3 NLL of Generated Feedback and Human Ratings

For the sentence generation process, the generated sentences were the ones with the lowest NLL after 50 steps of running. The feedback phrases were generated using a sentence paraphrasing CGMH approach before being passed on to the performance classifier. The feedback templates were sampled from the ASAP rating descriptors and feedback phrases were generated based on the language model trained on the IMDB dataset.

Figure 1 presents the training process of the language model, and Table 8 shows the NLL and human-rater evaluations of the generated sentences regarding *Fluency* and *Relatedness* on a scale of 0 to 1. The higher the scores, the more fluent and related the generated feedback sentences. The results revealed that the MCMC method is able to generate fluent and semantically-relevant sentences.

Table 8. NLL and rater evaluation on sentence generation

Evaluation Methods	Measures
NLL	10.01
Human Rating: Fluency	0.62
Human Rating: Relatedness	0.52

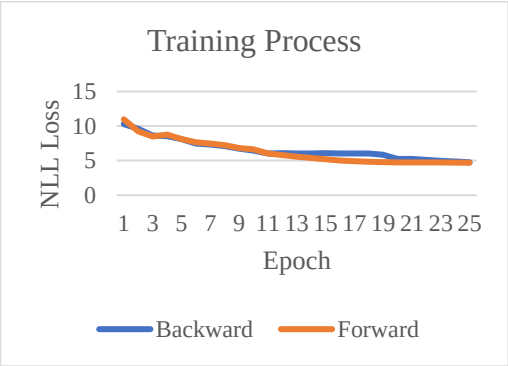

Figure 1. Learning curves of the training process of the language model (NLL convergence w.r.t. epochs).

5. DISCUSSION AND CONCLUSION

This study proposed and implemented a novel framework for an automated assessment and reporting framework with a combination of supervised deep learning models and unsupervised MCMC sampling method. Specifically, this study compared the performances of three models, namely CNN, CNN+LSTM, and CNN+Bi-LSTM, on AES tasks in the same context. Results revealed that CNN+LSTM demonstrated the highest performance on the AES tasks among the three algorithms. Moreover, the CNN+LSTM outperformed the baseline model on seven out of eight writing prompts, which demonstrates the potential of word-embeddings and deep learning models on automated essay scoring.

A recent literature review revealed that text-based feedback was more effective in improving performance [28]. Providing feedback within digital learning and assessment systems is essential for students' self-directed learning. However, it is laborious to manually devise a large amount of expert-derived quality feedback. Compared with sentence-generation supervised-learning methods, the CGMH sentence-paraphrasing unsupervised-learning method can augment the expert-driven feedback template corpus by generating feedback phrases with higher efficiency and flexibility. Thus, the proposed method is promising in promoting text-based feedback generation within automated assessment systems. Results of the current study could facilitate future implementations and validations of personalized automated feedback provision for ITSs and other virtual learning systems.

6. LIMITATIONS AND FUTURE WORK

We identified several limitations in the present study. First, this study does not empirically validate the AES and the automated feedback generation system in educational settings. Future research will be conducted to provide empirical evidence on the validity and efficiency of the framework. Second, the present framework generates feedback using a holistic score for essays. Future research will incorporate linguistic components into AES to enhance the interpretability of the scoring results and to generate more fine-grained feedback.

7. ACKNOWLEDGEMENT

We would like to thank the reviewers for their helpful feedback and the following granting agencies for supporting this research: the Social Sciences and Humanities Research Council of Canada - Insight Development Grant (SSHRC IDG) RES0034954 and Insight Grant (SSHRC IG) RES0048110, Natural Sciences and Engineering Research Council Discovery Grant (NSERC DG) RES0043209, Killam Cornerstone Operating Grant RES0043207, Alberta Innovates, and Alberta Advanced Education.

8. REFERENCES

[1] D. Alikaniotis, H. Yannakoudakis, and M. Rei. Automatic text scoring using neural networks. *arXiv:1606.04289*, 2016.

[2] Y. Attali and J. Burstein. Automated essay scoring with e-rater® V. 2. *The Journal of Technology, Learning and Assessment*, 4(3):i-21, 2006.

[3] T. Barnes and J. Stamper. Automatic hint generation for logic proof tutoring using historical data. *Journal of Educational Technology & Society*, 13(1):3-12, 2010.

[4] P. Blayney and M. Freeman. Automated formative feedback and summative assessment using individualised spreadsheet assignments. *Australasian Journal of Educational Technology*, 20(2):209-231, 2004.

[5] D. Boud and E. Molloy. (Eds.). *Feedback in Higher and Professional Education: Understanding it and Doing it Well*. Routledge, 2013.

[6] J. Burstein, J. Tetreault, and N. Madnani. The E-rater® automated essay scoring system. In *Handbook of Automated Essay Evaluation*, pages 77-89. Routledge, 2013.

[7] J. Debuse, M. Lawley, and R. Shibl. The implementation of an automated assessment feedback and quality assurance system for ICT courses. *Journal of Information Systems Education*, 18(4):491-502, 2007.

[8] B. Di Eugenio, D. Fossati, D. Yu, S. M. Haller, and M. Glass. Natural Language Generation for Intelligent Tutoring Systems: A case study. In *AIED*, pages 217-224, 2005.

[9] F. Dong and Y. Zhang. Automatic features for essay scoring–an empirical study. In *Proceedings of the 2016 Conference on EMNLP*, pages 1072-1077, 2016.

[10] F. Dong, Y. Zhang and J. Yang. Attention-based recurrent convolutional neural network for automatic essay scoring. In *Proceedings of the 21st Conference on* CoNLL, pages 153-162, 2017.

[11] M. Dzikovska, N. Steinhauser, E. Farrow, J. Moore, and G. Campbell, G. BEETLE II: Deep natural language understanding and automatic feedback generation for intelligent tutoring in basic electricity and electronics. *International Journal of Artificial Intelligence in Education*, 24(3):284-332, 2014.

[12] Y. Farag, H. Yannakoudakis, and T. Briscoe. Neural automated essay scoring and coherence modeling for adversarially crafted input. *arXiv:1804.06898*, 2018.

[13] P. Ferguson. Student perceptions of quality feedback in teacher education. *Assessment & Evaluation in Higher Education*, 36(1):51-62, 2011.

[14] J. Gao, B. Pang, and S. S. Lumetta. Automated feedback framework for introductory programming courses. In *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, pages 53-58. ACM, 2016.

[15] C. J. Geyer. Practical Markov Chain Monte Carlo. In *Statistical Science*, pages 473-483, 1992.

[16] M. J. Gierl, S. Latifi, H. Lai, A. P. Boulais, and A. De Champlain. Automated essay scoring and the future of educational assessment in medical education. *Medical Education*, 48(10):950-962, 2014.

[17] J. Hattie and H. Timperley. The Power of Feedback. Review of Educational Research. *Review of Educational Research*. 77(1):81-112, 2007.

[18] N. T. Heffernan and K. R. Koedinger. An intelligent tutoring system incorporating a model of an experienced human tutor. In *International Conference on Intelligent Tutoring Systems*, pages 596-608. Springer, 2002.

[19] R. Higgins, P. Hartley, and A. Skelton. The conscientious consumer: Reconsidering the role of assessment feedback in student learning. *Studies in Higher Education*, 27(1):53-64, 2002.

[20] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9(8):1735-1780, 1997.

[21] L. E. Holmes and L. J. Smith. Student evaluations of faculty grading methods. *Journal of Education for Business*, 78(6):318-323, 2003.

[22] C. Jin, B. He, K. Hui, and L. Sun. TDNN: a two-stage deep neural network for prompt-independent automated essay scoring. In *Proceedings of the 56th Annual Meeting of the ACL* (Volume 1: Long Papers), pages 1088-1097, 2018.

[23] H. Keuning, J. Jeuring, and B. Heeren. Towards a systematic review of automated feedback generation for programming exercises. In *Proceedings of the 2016 ACM Conference on ITiCSE*, pages 41-46. ACM, 2016.

[24] E. Kosba, V. Dimitrova, and R. Boyle. Adaptive feedback generation to support teachers in web-based distance education. *User Modeling and User-Adapted Interaction*, 17(4):379-413, 2007.

[25] T. K. Landauer, P. W. Foltz, and D. Laham. An introduction to latent semantic analysis. *Discourse Processes*, 25(2-3):259-284, 1998.

[26] M. Liu, Y. Li, W. Xu, and L. Liu. Automated essay feedback generation and its impact on revision. *IEEE Transactions on Learning Technologies*, 10(4):502-513, 2016.

[27] M. Maniktala, C. Cody, A. Isvik, N. Lytle, M. Chi, and T. Barnes. Extending the hint factory for the assistance dilemma: A novel, data-driven HelpNeed predictor for proactive problem-solving help. *Journal of Educational Data Mining*, 12(4): 24-65, 2020.

[28] S. Marwan, N. Lytle, J. J. Williams, and T. Price. The impact of adding textual explanations to next-step hints in a novice programming environment. In *Proceedings of the 2019 ACM Conference on ITiCSE*, pages 520-526, 2019.

[29] A. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the ACL: HLT*, pages 142-150, 2011.

[30] D. S. McNamara, S. A. Crossley, R. D. Roscoe, L. K. Allen, and J. Dai. A hierarchical classification approach to automated essay scoring. *Assessing Writing*, 23:35-59, 2015.

[31] D. C. Merrill, B. J. Reiser, M. Ranney, and J. G. Trafton. Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. *The Journal of the Learning Sciences*, 2(3):277-305, 1992.

[32] N. Miao, H. Zhou, L. Mou, R. Yan, and L. Li. CGMH: Constrained sentence generation by Metropolis-Hastings sampling. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 33), pages 6834-6842, 2019.

[33] E. B. Page. Computer grading of student prose, using modern concepts and software. *The Journal of Experimental Education*, 62(2):127-142, 1994.

[34] P. Phandi, K. M. A. Chai, and H. T. Ng. Flexible domain adaptation for automated essay scoring using correlated linear regression. In *Proceedings of the 2015 Conference on EMNLP*, pages 431-439, 2015.

[35] J. Pennington, R. Socher, and C. D. Manning, C. D. Glove: Global vectors for word representation. In *Proceedings of the Conference on EMNLP*, pages 1532-1543, 2014.

[36] I. Perikos, F. Grivokostopoulou, and I. Hatzilygeroudis. Assistance and feedback mechanism in an intelligent tutoring system for teaching conversion of natural language into logic. *International Journal of Artificial Intelligence in Education*, 27(3):475-514, 2017.

[37] T. W. Price, R. Zhi, and T. Barnes, T. Hint generation under uncertainty: The effect of hint quality on help-seeking behavior. In *International Conference on Artificial Intelligence in Education*, pages 311-322. Springer, 2017.

[38] S. Shatnawi, M. M. Gaber, and M. Cocea. Automatic content related feedback for MOOCs based on course domain ontology. In *International Conference on Intelligent Data Engineering and Automated Learning*, pages 27-35. Springer, 2014.

[39] J. Su, J. Xu, X. Qiu, and X. Huang. Incorporating discriminator in sentence generation: A Gibbs sampling method. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 32, No. 1), pages 5496-5503, 2018.

[40] K. Taghipour and H. T. Ng. A neural approach to automated essay scoring. In *Proceedings of the 2016 Conference on EMNLP*, pages 1882-1891, 2016.

[41] Vantage Learning. Research summary: IntelliMetricTM scoring accuracy across genres and grade levels, 2006.

[42] R. Williams and H. Dreher. Automatically grading essays with markit?. *Issues in Informing Science and Information Technology*, 1:0693-0700, 2004.

[43] D. M. Williamson, X. Xi, and F. J. Breyer. A framework for evaluation and use of automated scoring. *Educational Measurement: Issues and Practice*, 31(1):2-13, 2012.

[44] 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 ACL: HLT*, pages 180-189, 2011.