


Natural Language Enhancement for English Teaching Using Character-Level Recurrent Neural Network with Back Propagation Neural Network based Classification by Deep Learning Architectures

Zhiling Yang

(School of Foreign Studies, Wenzhou University, Wenzhou, Zhejiang, 325035, China,
College of Industrial Education, Technological University of the Philippines, Manila,
Philippines

 <https://orcid.org/0000-0002-0521-2681>, zlyang@wzu.edu.cn)

Abstract: Natural Language Processing (NLP) is an efficient method for enhancing educational outcomes. In educational settings, implementing NLP entails starting the learning process through natural acquisition. English teaching and learning have received increased attention from the relevant education departments as an integral aspect of the new curriculum reform. The environment of English teaching and learning is undergoing extraordinary changes as a result of the constant improvement and extension of teaching level and scale, as well as the growth of Internet information technology. As a result, the current research aims to look into techniques for efficiently using AI (artificial intelligence) apps to teach and learn English from the perspective of university students. This research can measure the levels as well as effectiveness of the employment of AI applications for teaching English based on deep learning techniques. There, the NLP based language enhancement has been carried out using Character-level recurrent neural network with back propagation neural network (Cha_RNN_BPNN) based classification. With the help of this DL (deep learning) technique, it is possible to use AI methods to assist teachers in analysing and diagnosing students' English learning behaviour, replacing teachers in part to answer students' questions in a timely manner, and automatically grading assignments during the English teaching process. Experimental analysis shows Word Perplexity, Flesch-Kincaid (F-K) Grade Level for Readability, Cosine Similarity for Semantic Coherence, gradient change of NN, validation accuracy, and training accuracy of the proposed technique.

Keywords: Natural Language Processing, artificial intelligence, language enhancement, English learning, deep learning

Categories: H.5.1, L.3.2, I.2.7, I.2.8, I.6.4, I.7.0, M.7

DOI: 10.3897/jucs.94162

1 Introduction

As a scientific subject, NLP has a 50-years old history, with applications in education dating to the 1960s. The initial focus was on autonomously grading student texts and building text-based dialogue tutoring systems, but later works covered spoken language technologies as well. While research in these classic application fields progresses, contemporary phenomena such as big data, mobile technologies, social media, and

MOOCs have created a slew of new research opportunities and problems [Sun et al. 2020]. High-stake text and speech evaluations, writing assistants, and online instructional environments are now commercial applications, with corporations actively reaching out to the research community. Research into the application of natural language processing to education often follows an iterative lifetime, as indicated in Figure 1 [Litman, 2016]. Technological innovation is driven by societal needs and then meets them. Similarly, technological innovation is influenced by and contributes to educationally relevant theories and facts. A research challenge in the area of NLP for educational applications is frequently driven by a real-world student or teacher requirement [Ding et al. 2021], as shown in the upper right of the picture.

For example, given the large student-to-teacher ratio in MOOCs, it is difficult for an instructor to read all postings in the discussion forums; can NLP determine the posts that require an instructor's intervention instead? Following that, at the bottom of the graphic, restrictions on problem solutions are specified using appropriate theory or data-driven results from the literature. Even before MOOCs, for example, there existed a pedagogical literature on teacher intervention. Finally, an NLP-based technology is built, implemented, and evaluated in the upper left of the figure. The cycle is likely to iterate according to an error analysis. An intervention system designed for a science MOOC, for example, may need to be revised to match the needs of a humanities instructor [Galanis et al. 2021].

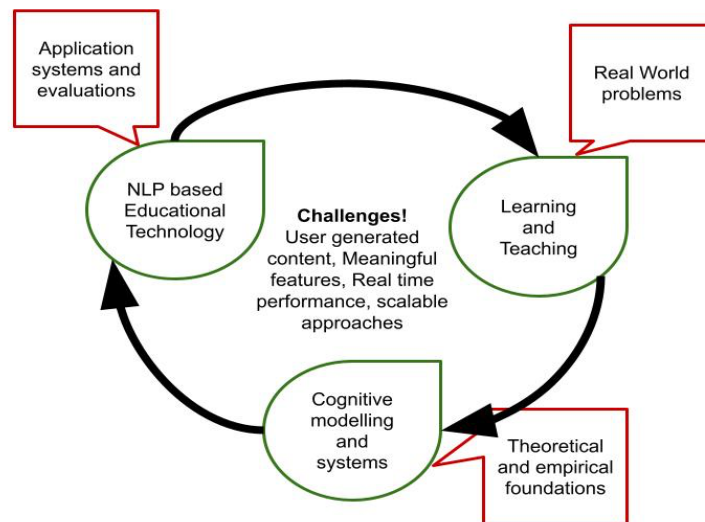


Figure 1: Typical NLP for education research lifecycle

The key of AI technology lies in intelligence. Intelligence in the field of English teaching as well as learning is also based on the basic points of the education industry. English teaching activities have diversity and complexity. Due to the different levels of English teachers, the implementation of educational teaching hardware, even various interests, cognitive styles of students will affect English learning [Wang, 2019].

Artificial intelligence can simulate as well as reproduce human thinking and reasoning and expression of thinking through targeted processing and analysis. Therefore, in the dynamic environment of English teaching and learning, AI can play a unique strategy in English teaching and learning. Secondly, due to the differences of language environment, English often brings many troubles to Chinese students in the process of actual communication and application, even affecting the daily teaching of English teachers. Individualized and oral English learning stimulates students' interest in English learning as well as improves the learning atmosphere in English classrooms. This is not only a significant reflection of the value of English teaching and learning in the education industry, but also an in-depth exploration of the value of modern information technology [Su et al. 2019].

The contribution of this research is as follows:

1. To investigate the use of NLG in conjunction with DL methods to give automatic support for students participating in English discussions.
2. To assess the extent to which AI apps are being used to teach English, as well as their effectiveness.
3. It has been completed the conception for the use of AI applications in the teaching of English. Basics, content, objectives, processors, and assessment methods would all be included.
4. To classify the English teaching-based data collected from 7,066 students, with a large number of 42,307 discussion forums using Character-level recurrent neural network with back Propagation neural network (Cha_RNN_BPNN).

2 Theoretical Background

There are a variety of effective ways in NLP that help in educational contexts, such as the function of empirical data, corpora, and other linguistic features that are important and effective in the language learning process. Reference [Li and Xing, 2021] addressed new prospects for improving natural language processing (NLP) and its utility in the development of educational tools such as reading and writing materials. The use of linguistics in the classroom can help students manage and deal with the challenges of reading and writing. This can be accomplished by examining syntactic and morphological parameters. Motivation in language acquisition is a powerful tool that may also be used to improve students' educational practises and academic achievement [Schmidhuber, 2015]. DL research has seen a significant increase in activity in recent years. In [Caterini, 2017], popular architectural models and training techniques were used to provide a general introduction to DL in NN (neural networks). Because of the rise in data volume and processing power, NNs with increasingly complicated topologies have got a lot of interest and have been used in a variety of sectors [Salah et al. 2010]. A large number of research investigations focused on practical applications, yielding a great number of study findings. DL methods have been utilised in image

analysis, text analysis, speech recognition, and other fields, providing solutions to a variety of real-world problems [Sun et al. 2021]. The basic knowledge of transfer learning, numerous types of methodologies utilised to accomplish transfer learning, as well as how transfer learning was applied in many subfields of medical image analysis were all evaluated by the authors [Boom et al. 2019]. The review demonstrates that recent developments in DL, particularly advances in transfer learning, have enabled the identification, classification, and quantification of specific patterns from a large number of medical pictures [Hwang and Sung, 2017]. The authors [Artemova et al. 2021] introduced the Character-level CNN with Shortcuts (Char-CNNs) method in an attempt to provide an automated approach for determining whether material in social media comprises cyberbullying. To assist people comprehend how DL methods may be tailored to meet the challenge of speaker recognition, [Diao and Hu, 2021] presented a new technique to extract speaker characteristics by developing CNN filters linked to speakers. Researchers have employed deep learning algorithms to build efficient techniques to evaluate students' learning state as well as behaviour [Lauriola et al. 2022], [Wang et al. 2020], [Yang et al. 2019]. The success of teaching and learning depends on interaction. DL methods are used by researchers to create methods to assist as well as encourage students' learning enthusiasm because of their strength in natural language generation as well as processing [Torfi et al. 2020].

3 Materials and Methods

The analytic descriptive method was utilised to explore and analyse a specific phenomenon in this study. The notion of AI, its components, and its applications in the field of teaching have all been described in previous research. To construct a study tool as well as to establish the study content, this study identified AI possibilities for teaching English. The article looked at how AI can be used to teach English as a second language, how effective it is, and what practical ways can be utilised to implement it. Figure 2 depicts the whole AI based English teaching platform design.

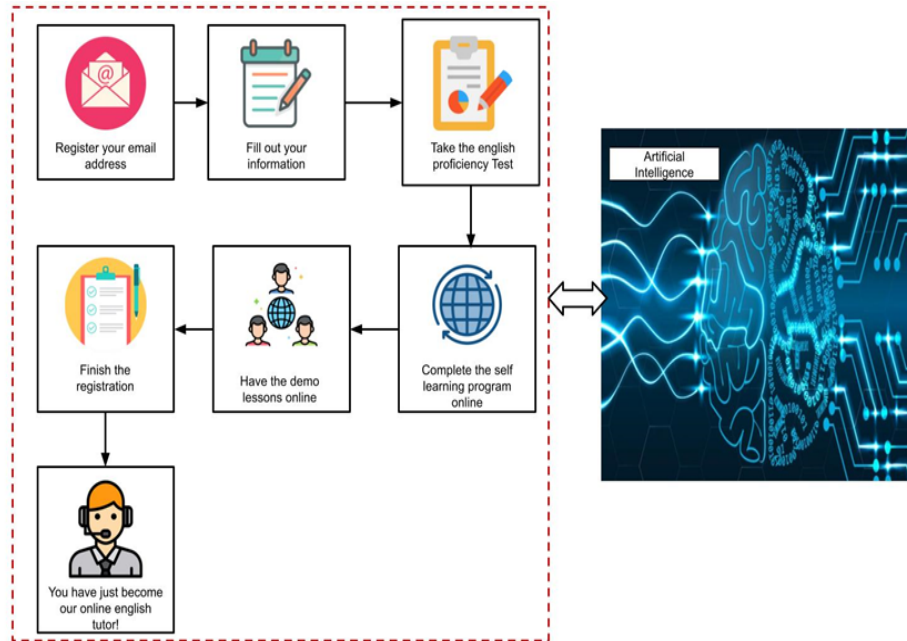


Figure 2: Overall AI based English teaching platform architecture

Advanced information processing technology is used in deep learning and knowledge discovery. It not only looks at historical data and discovers apparent parallels, but it also does a high-level inquiry to best assess, anticipate future trends and much more. With the quick advancement of Internet data innovation, electrical science, and technology, electronic archives are being used in colleges. The distinction between good and bad students is no longer based on the results of the test. The exceptional findings of assessment techniques play a critical and irreplaceable role throughout education. AI methods can summarise and analyse data from a wide range of complex data that are relevant to our article on the overall group situation as well as help with final decision-making using classification techniques.

3.1 Character-level recurrent neural network with back Propagation neural network (Cha_RNN_BPNN) based data classification

Due to the smaller units of tokens, CLMs (Character-Level Modelling) must examine a longer sequence of historical tokens to forecast the next token than WLMs (Word Level Modelling). RNN is then trained to predict the next character x_{t+1} by minimising cross-entropy loss of SoftMax output, which indicates the next character's probability distribution, as illustrated in Figure 3.

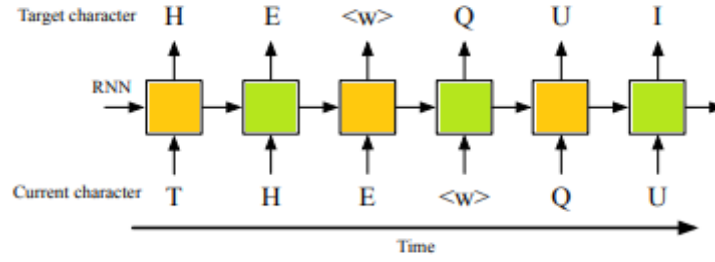


Figure 3: Training an RNN-based CLM

First, instead of using WLMs, we use character-level RNN LMs as our foundation model, which extends to include long-term contexts. As a result, character-level clocks are used to generate the output probabilities. For end-to-end voice recognition, this characteristic is particularly useful for character-level beam search. Because our model's inputs and outputs are identical to those of standard character-level RNNs, the same training techniques as well as recipes may be utilised without modification. Furthermore, when compared to WLM-based models, the suggested models contain much fewer parameters because the size of our model is not directly proportional to the word size of the training set. Expand the existing RNN structures with external clocks as well as reset signals to make them more general. The hierarchical RNNs' basic building components are the extended models. Eq. (1), (2) can be used to generalise most sorts of RNNs or recurrent layers.

$$\mathbf{s}_t = f(\mathbf{x}_t, \mathbf{s}_{t-1}), \mathbf{y}_t = g(\mathbf{s}_t) \tag{1}$$

$$\mathbf{y}_t = \mathbf{s}_t = \mathbf{h}_t = \sigma(W_{hx}\mathbf{x}_t + W_{hh}\mathbf{h}_{t-1} + \mathbf{b}_h) \tag{2}$$

Whx and Whh are weight matrices and bh is the bias vector, where ht is the hidden layer activation, $\sigma(\cdot)$ is the activation function, It is also possible to convert LSTMs with forget gates and peephole connections to the generalised version. The LSTM layer's forward equations are as follows: eq. (3):

$$\begin{aligned} \mathbf{i}_t &= \sigma(W_{ix}\mathbf{x}_t + W_{ih}\mathbf{h}_{t-1} + W_{im}\mathbf{m}_{t-1} + \mathbf{b}_i) \\ \mathbf{f}_t &= \sigma(W_{fx}\mathbf{x}_t + W_{fh}\mathbf{h}_{t-1} + W_{fm}\mathbf{m}_{t-1} + \mathbf{b}_f) \\ \mathbf{m}_t &= \mathbf{f}_t \circ \mathbf{m}_{t-1} + \mathbf{i}_t \circ \tanh(W_{mx}\mathbf{x}_t + W_{mh}\mathbf{h}_{t-1} + \mathbf{b}_m) \\ \mathbf{o}_t &= \sigma(W_{ox}\mathbf{x}_t + W_{oh}\mathbf{h}_{t-1} + W_{om}\mathbf{m}_t + \mathbf{b}_o) \\ \mathbf{h}_t &= \mathbf{o}_t \circ \tanh(\mathbf{m}_t) \end{aligned} \tag{3}$$

These equations are generalised by changing the variables. | $\mathbf{s}_t = [\mathbf{m}_t, \mathbf{h}_t]$ and $\mathbf{y}_t = \mathbf{h}_t$

Any generalised RNN is transformed to one with an external clock signal, ct, using the Eq. (4)

$$\mathbf{s}_t = (1 - c_t)\mathbf{s}_{t-1} + c_t f(\mathbf{x}_t, \mathbf{s}_{t-1}), \mathbf{y}_t = g(\mathbf{s}_t) \tag{4}$$

where c_t is 0 or 1. Only when $c_t = 1$ does the RNN update its state and output. When $c_t = 0$, the state and output values are the same as they were in the preceding phase. Setting s_t to 0 causes the RNNs to be reset by eqn. (5)

$$\mathbf{s}_t = (1 - c_t)(1 - r_t)\mathbf{s}_{t-1} + c_t f(\mathbf{x}_t, (1 - r_t)\mathbf{s}_{t-1}) \quad (5)$$

where reset signal $r_t = 0$ or 1. RNN forgets prior context when $r_t = 1$. Extended RNN equations with clock as well as reset signals are also differentiable if the original RNN equations are. As a result, current gradient-based RNN training techniques, such as BPTT, may be used without modification to train the expanded versions.

$$\begin{aligned} h_n &= f(x_n, h_{n-1}) \\ x_{n+1} &= g(h_n) \end{aligned} \quad (6)$$

The previous technique is utilized to produce an unlimited sequence of tokens given a sufficient initialization. To generate the next token in the sequence, one can either sample from this mass function or simply use eq. (7) to select the token with the highest probability.

$$\begin{aligned} h_n &= f(x_n, h_{n-1}) \\ x_{n+1} &= g(h_n) \end{aligned} \quad (7)$$

$$p(x_{1:N}) = \prod_{n=1}^N p(x_n | x_{1:n-1}) = \prod_{n=1}^N p(x_n, h_{n-1}) \quad (8)$$

The backpropagation technique is used to train regular feedforward neural networks. In this case, a specific input is initially propagated across the network before the output is computed. A differentiable loss function is then utilized to compare the output to a ground truth label. The gradients of loss with respect to all of the parameters in the network are evaluated using the chain rule in backward pass by eq. (9):

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \cdot \nabla_{\mathbf{w}} \mathcal{L}(y, \hat{y}) \quad (9)$$

The learning rate η , which regulates the magnitude of steps made on every weight update, is shown here. In practise, input samples are sampled from the training dataset in equal-sized batches, with losses averaged or summed, resulting in less noisy updates. Mini-batch gradient descent is the term for this method. The output activation function ψ in a regression model is usually the identity function; to be more general, by Eq (10)

$$\psi(a_1, \dots, a_K) = (g_1(a_1), \dots, g_K(a_K)) \quad (10)$$

where g_1, \dots, g_K are functions from \mathbb{R} to \mathbb{R} . Let us compute R_i 's partial derivatives with respect to the output layer's weights. Recalling that by Eq. (11)

$$\begin{aligned} a^{(L+1)}(x) &= b^{(L+1)} + W^{(L+1)}h^{(L)}(x) \\ \frac{\partial R_i}{\partial W_{k,m}^{(L+1)}} &= -2 \left(Y_{i,k} - f_k(X_i, \theta) \right) g'_k \left(a_k^{(L+1)}(X_i) \right) h_m^{(L)}(X_i) \end{aligned} \quad (11)$$

Differentiating now with relation to the preceding layer's weights by eq. (12)

$$\begin{aligned} \frac{\partial R_i}{\partial W_{m,l}^{(L)}} &= -2 \sum_{k=1}^K (Y_{i,k} - f_k(X_i, \theta)) g'_k(a_k^{(L+1)}(X_i)) \frac{\partial a_k^{(L+1)}(X_i)}{\partial W_{m,l}^{(L)}} \\ a_k^{(L+1)}(x) &= \sum_j W_{k,j}^{(L+1)} h_j^{(L)}(x), \\ h_j^{(L)}(x) &= \phi(b_j^{(L)} + \langle W_j^{(L)}, h^{(L-1)}(x) \rangle). \end{aligned} \tag{12}$$

This leads to Eq. (13)

$$\frac{\partial a_k^{(L+1)}(x)}{\partial W_{m,l}^{(L)}} = W_{k,m}^{(L+1)} \phi'(b_m^{(L)} + \langle W_m^{(L)}, h^{(L-1)}(x) \rangle) h_l^{(L-1)}(x). \tag{13}$$

Let us introduce the notations in eq. (14)

$$\delta_{k,i} = -2 (Y_{i,k} - f_k(X_i, \theta)) g'_k(a_k^{(L+1)}(X_i)) \quad s_{m,i} = \phi'(a_m^{(L)}(X_i)) \sum_{k=1}^K W_{k,m}^{(L+1)} \delta_{k,i}. \tag{14}$$

Then by eq. (15)

$$\begin{aligned} \frac{\partial R_i}{\partial W_{k,m}^{(L+1)}} &= \delta_{k,i} h_m^{(L)}(X_i) \\ \frac{\partial R_i}{\partial W_{m,l}^{(L)}} &= s_{m,i} h_l^{(L-1)}(X_i) \end{aligned} \tag{15}$$

The gradient values are used to update the gradient descent algorithm's parameters by eq. (16):

$$W_{k,m}^{(L+1,r+1)} = W_{k,m}^{(L+1,r)} - \varepsilon_r \sum_{i \in B} \frac{\partial R_i}{\partial W_{k,m}^{(L+1,r)}} \tag{16}$$

where B is a batch and $\varepsilon_r > 0$ is learning rate that satisfies $\varepsilon_r \rightarrow 0, \sum_r \varepsilon_r = \infty, \sum_r \varepsilon_r^2 < \infty$, for example $\varepsilon_r = 1/r$.

Use Back-propagation equations to evaluate the gradient by a two-pass technique. In forward pass, fix value of current weights $\theta^{(r)} = (W^{(1,r)}, b^{(1,r)}, \dots, W^{(L+1,r)}, b^{(L+1,r)})$ and evaluate predicted values $f(X_i, \theta^{(r)})$ and all intermediate values $(a^{(k)}(X_i), h^{(k)}(X_i) = \phi(a^{(k)}(\tilde{X}_i)))_{1 \leq k \leq L+1}$ that are stored.

Indeed for the sigmoid function by eq. (17)

$$\phi(x) = \frac{1}{1 + \exp(-x)}, \quad \phi'(x) = \phi(x)(1 - \phi(x)) \tag{17}$$

For the hyperbolic tangent function ("tanh") by eq. (18)

$$\phi(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}, \phi'(x) = 1 - \phi^{(L,r)} \tag{18}$$

Consider K class classification issue. Output is $f(x) = \begin{pmatrix} \mathbb{P}(Y = 1/x) \\ \vdots \\ \mathbb{P}(Y = K/x) \end{pmatrix}$. Consider the output activation function is SoftMax function given by eq. (19).

$$\text{softmax}(x_1, \dots, x_K) = \frac{1}{\sum_{k=1}^K e^{x_k}} (e^{x_1}, \dots, e^{x_K}) \tag{19}$$

$$\frac{\partial \text{softmax}(x)_i}{\partial x_j} = \text{softmax}(x)_i(1 - \text{softmax}(x)_i) \quad \text{if } i = j = -\text{softmax}(x)_i \text{softmax}(x)_j \text{ if } i \neq j \text{ by eq. (20)}$$

$$(f(l))_m = \sum_{k=1}^K \mathbf{1}_{y=k} (f(l))_k \tag{20}$$

Then $-\log(f(l))_m = -\sum_{k=1}^K \mathbf{1}_{y=k} \log(f(l))_k = \ell(f(l), m)$ for loss function l combined with cross-entropy. Output weights $\frac{\partial \ell(f(l), m)}{\partial W_{i,j}^{(L+1)}}$ Output biases $\frac{\partial \ell(f(l), m)}{\partial b_i^{(L+1)}}$

Hidden weights $\frac{\partial \ell(f(l), m)}{\partial W_{i,j}^{(h)}}$ Hidden biases $\frac{\partial \ell(f(l), m)}{\partial b_i^{(h)}}$, for $1 \leq h \leq L$. if $z(x) = \phi(a_1(x), \dots, a_j(x))$, then $\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial a_j} \frac{\partial a_j}{\partial x_i} = \left\langle \nabla \phi, \frac{\partial a}{\partial x_i} \right\rangle$

for $1 \leq h \leq L$. Use chain-rule: if $z(x) = \phi(a_1(x), \dots, a_j(x))$, then by eq. (21)

$$\frac{\partial z}{\partial x_i} = \sum_j \frac{\partial z}{\partial a_j} \frac{\partial a_j}{\partial x_i} = \left\langle \nabla \phi, \frac{\partial a}{\partial x_i} \right\rangle. \tag{21}$$

Hence by eq. (22)

$$\begin{aligned} \frac{\partial \ell(f(l), m)}{\partial (a^{(L+1)}(l))_i} &= - \sum_j \frac{\mathbf{1}_{y=j}}{(f(y))_x} \frac{\partial \text{softmax}(a^{(L+1)}(l))_j}{\partial (a^{(L+1)}(a))_i} \\ &= - \frac{1}{(f(l))_x} \frac{\partial \text{softmax}(a^{(L+1)}(l))_m}{\partial (a^{(L+1)}(l))_i} \\ &= - \frac{1}{(f(l))_x} \text{softmax}(a^{(L+1)}(a))_m (1 - \text{softmax}(a^{(L+1)}(a))_m) \mathbf{1}_m \\ &= i + \frac{1}{(f(l))_m} \text{softmax}(a^{(L+1)}(a))_i \text{softmax}(a^{(L+1)}(a))_m \mathbf{1}_{1 \neq i} \\ \frac{\partial \ell(f(l), m)}{\partial (a^{(L+1)}(m))_i} &= (-1 + f(a)_m) \mathbf{1}_l = i + f(l)_i \mathbf{1}_m \neq l \end{aligned} \tag{22}$$

Hence by eq. (23),

$$\nabla_{a^{(L+1)}(m)} \ell(f(a), m) = f(a) - e(m) \tag{23}$$

The PD of the loss function with respect to the output bias can now be easily obtained by eq. (24)

$$\frac{\partial(a^{(L+1)(l)})_k}{\partial W_{i,j}^{(L+1)}} = a^{(L)(l)} \mathbf{1}_{i=k} \tag{24}$$

$$\nabla_{b^{(L+1)}} \ell(f(l)m) = f(l) - e(m)$$

Now compute the loss function's gradient at the hidden layers given by eq. (25)

$$\frac{\partial \ell(f(a), m)}{\partial (h^{(k)}(a))_j} = \sum_i \frac{\partial \ell(f(a), m)}{\partial (a^{(k+1)}(a))_i} \frac{\partial (a^{(k+1)}(a))_i}{\partial (h^{(k)}(a))_j} \tag{25}$$

Now compute the loss function's gradient at hidden layers given by eq. (26).

$$(a^{(k+1)}(a))_i = b_i^{(k+1)} + \sum_j W_{i,j}^{(k+1)} (h^{(k)}(a))_j$$

$$\frac{\partial \ell(f(a), m)}{\partial (h^{(k)}(a))_j} = \sum_i \frac{\partial \ell(f(a), m)}{\partial (a^{(k+1)}(a))_i} \frac{\partial (a^{(k+1)}(a))_i}{\partial (h^{(k)}(a))_j} \tag{26}$$

Hence by eq. (27)

$$\frac{\partial \ell(f(a), m)}{\partial h^{(k)}(m)_j} = \sum_i \frac{\partial \ell(f(a), m)}{\partial a^{(k+1)}(m)_i} W_{i,j}^{(k+1)}$$

$$\nabla_{h^{(k)}(x)} \ell(f(a), m) = (W^{(k+1)})' \nabla_{a^{(k+1)}(m)} \ell(f(a), m) \tag{27}$$

Recalling that $h^{(k)}(m)_j = \phi(a^{(k)}(m)_j)$ by eq. (28),

$$\frac{\partial \ell(f(a), m)}{\partial a^{(k)}(m)_j} = \frac{\partial \ell(f(a), m)}{\partial h^{(k)}(m)_j} \phi'(a^{(k)}(m)_j). \tag{28}$$

Hence by eq. (29), (30)

$$\nabla_{a^{(k)}(m)} \ell(f(l), m) = \nabla_{h^{(k)}(m)} \ell(f(a), m) \odot (\phi'(a^{(k)}(a)_1), \dots, \phi'(a^{(k)}(a)_j), \dots)' \tag{29}$$

where \odot denotes element-wise product.

$$\frac{\partial \ell(f(a), m)}{\partial W_{i,j}^{(k)}} = \frac{\partial \ell(f(a), m)}{\partial a^{(k)}(a)_i} \frac{\partial a^{(k)}(a)_i}{\partial W_{i,j}^{(k)}}$$

$$= \frac{\partial \ell(f(a), m)}{\partial a^{(k)}(a)_i} h_j^{(k-1)}(l) \tag{30}$$

Finally, the loss function's gradient of hidden weights is given by eq. (31)

$$\nabla_{W^{(k)}} \ell(f(l), m) = \nabla_{a^{(k)}(x)} \ell(f(a), m) h^{(k-1)}(a)'. \tag{31}$$

The gradient must be computed about the hidden biases as the final step given by eq. (32).

$$\frac{\partial \ell(f(a), m)}{\partial b_i^{(k)}} = \frac{\partial \ell(f(a), m)}{\partial a^{(k)}(a)_i}$$

$$\nabla_{b^{(k)}} \ell(f(a), m) = \nabla_{a^{(k)}(x)} \ell(f(a), m) \quad (32)$$

3.2 Stochastic gradient-based back propagation algorithm:

Forward pass: fix value of current weights $\theta^{(r)} = (W^{(1,r)}, b^{(1,r)}, \dots, W^{(L+1,r)}, b^{(L+1,r)})$, and evaluate predicted values $f(X_i, \theta^{(r)})$ and all intermediate values $(a^{(k)}(X_i), h^{(k)}(X_i) = \phi(a^{(k)}(X_i)))_{1 \leq k \leq L+1}$ that are stored.

Backpropagation algorithm:

- Evaluate output gradient $\nabla_{a^{(L+1)}}(x) \ell(f(x), y) = f(x) - e(y)$.
- For $k = L + 1$ to 1
Evaluate gradient at the hidden layer k

$$\begin{aligned} \nabla_{W^{(k)}} \ell(f(x), y) &= \nabla_{a^{(k)}}(x) \ell(f(x), y) h^{(k-1)}(x)' \\ \nabla_{b^{(k)}} \ell(f(x), y) &= \nabla_{a^{(k)}}(x) \ell(f(x), y) \end{aligned}$$

Compute the gradient at the previous layer

$$\begin{aligned} \nabla_{h^{(k-1)}(x)} \ell(f(x), y) &= (W^{(k)})' \nabla_{a^{(k)}}(x) \ell(f(x), y) \text{ and} \\ \nabla_{a^{(k-1)}}(x) \ell(f(x), y) &= \nabla_{h^{(k-1)}}(x) \ell(f(x), y) \\ &\quad \odot (\dots, \phi'(a^{(k-1)}(x)_j), \dots)' \end{aligned}$$

Fix parameters ε : learning rate, m : batch size, nb: number of epochs.

For $l = 1$ to nb epochs

For $l = 1$ to n/m

Take a random batch of size m without replacement in learning sample: $(X_i, Y_i)_{i \in B_l}$

Compute the gradients with the backpropagation algorithm

$$\tilde{\nabla}_{\theta} = \frac{1}{m} \sum_{i \in B_l} \nabla_{\theta} \ell(f(X_i, \theta), Y_i).$$

Update the parameters

$$\theta^{\text{new}} = \theta^{\text{old}} - \varepsilon \tilde{\nabla}_{\theta}.$$

The gradient update is frequently terminated after traversing a defined number of time steps to scale back-propagation across time for use with extended sequences. Truncated back-propagation across time is the name for this method. Limit the frequency of gradient updates in addition to preventing them from back-propagating all way to the beginning of the sequence. The following is how the shortened BPTT works for a particular training sequence. The RNN processes a new token every time step, and after k_1 tokens are processed in so-called forward pass as well as the hidden state has been updated k_1 times truncated, these parameters will be used throughout the study. Figure 4 presents a graphical explanation of the truncated BPPT, in which the gradients are back-propagated for three (k_2) time steps per two (k_1) time steps. BPTT is started by back-propagating gradient for k_2 time steps. Note that k_1 should ideally be less than or equal to k_2 to maintain the maximum data efficiency, as otherwise some data points might be omitted during training.

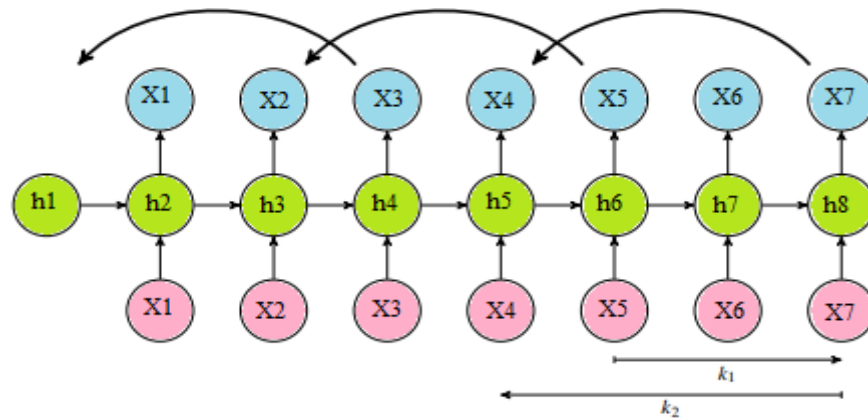


Figure 4: Example of truncated back propagation through time for $k_1 = 2$ and $k_2 = 3$. Thick arrow represents one back-propagation through time update.

3.3 Training and sampling methods for character-level RNNs with BPNN

Four techniques for training character-level RNNs as well as sampling new tokens from them are presented. The goal of the RNN model is to predict the next symbol or letter in a sequence given all preceding tokens and it is independent of these methods. Training as well as sampling techniques are thus essentially a practical means to accomplish the same goal, and the effect of the chosen scheme on RNN model's performance as well as efficiency is investigated in later sections. The 4 methods described are among the most fundamental as well as practical ways for training and sampling from RNNs, although there are undoubtedly many more combinations or versions. This result is a vector containing a probability distribution for all characters. Each scheme is a reasonable approximation of the original method and fits into the

truncated BPNN framework. It is also vital to remember that all schemes have the same goal in mind: to forecast the next token in a sequence.

4 Performance Analysis

Models of Cha_RNN_BPNN were calculated on the testing dataset after training using 3 metrics to assist researchers in model selection. Using the 2770 postings in the testing dataset, we first generated responses using the proposed technique. The postings in the testing set were utilized as "prompts" for Cha_RNN_BPNN methods, which meant they would generate responses to those prompts. Then, for evaluation, word perplexity, F-K grade level and cosine similarity were calculated.

Parameters	RNN	CNN	Cha_RNN_BPNN
Word Perplexity	84	88	89.8
F-K Grade Level for Readability	77	83	84
Cosine Similarity for Semantic Coherence	74	76	77
Gradient change of NN	65	69	70
Validation accuracy	87	93	95
Training accuracy	90	92	95

Table 1: Comparative analysis of English language teaching based on AI with existing and proposed techniques

The above Table 1 shows a comparative analysis of English language teaching based on AI between the existing and proposed techniques. Here the parameters analysed are Word Perplexity, F-K Grade Level for Readability, Cosine Similarity for Semantic Coherence, gradient change of NN, validation accuracy, and training accuracy.

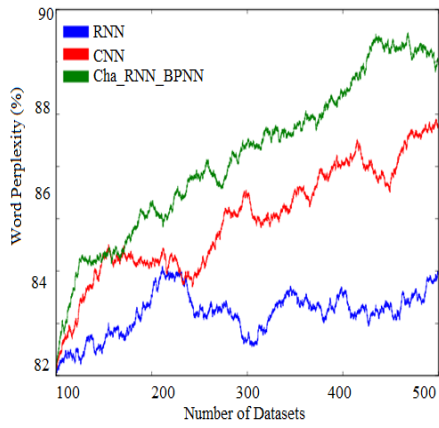


Figure 5: Comparative analysis of word perplexity

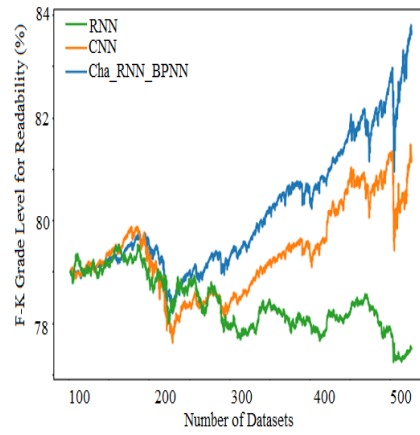


Figure 6: Comparative analysis of F-K grade level of Readability

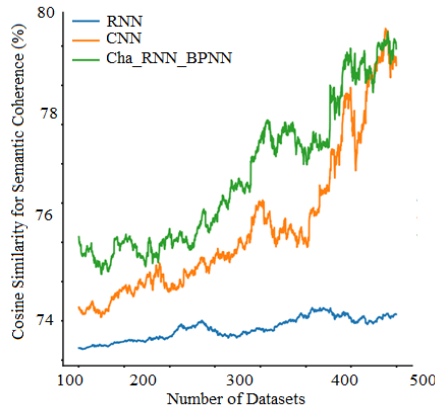


Figure 7: Comparative analysis of Cosine Similarity

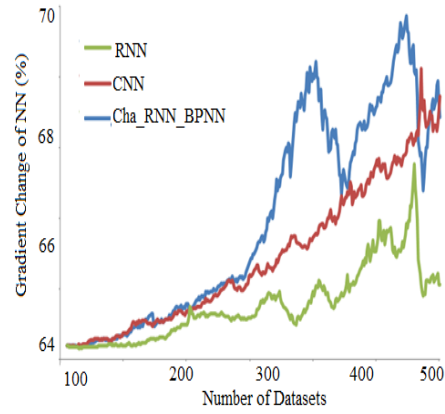


Figure 8: Comparative analysis of Gradient change of NN

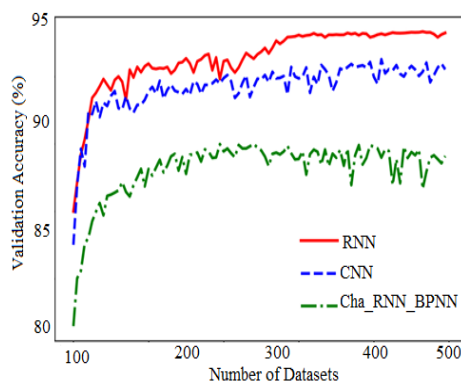


Figure 9: Comparative analysis of validation accuracy

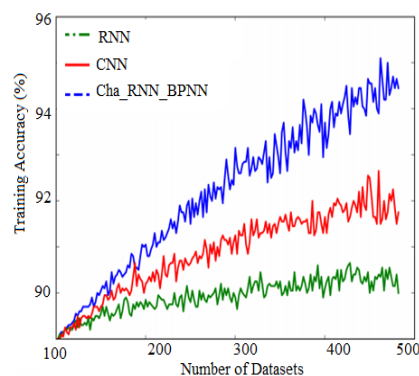


Figure 10: Comparative analysis of training accuracy

The above Figures 5-10 show the comparative analysis in terms of Word Perplexity, F-K Grade Level for Readability, Cosine Similarity for Semantic Coherence, gradient change of NN, validation accuracy, and training accuracy based on deep learning evaluation. While this method suggests 30 words to select from when producing the next word, it is said to have a word perplexity of 30. In practise, lower word perplexity is desired because it indicates that the model is more confident in its ability to predict the next word. The probability of a word x_i given a method is stated as s , and N is the number of alternative words. In both NLP and educational settings, F-K grade level is extensively utilised. Python package textstat was used to construct F-K grade levels of the produced texts from the testing dataset in this study. Because off-topic responses are unlikely to be helpful and can be confusing for pupils, they should be avoided. Manual effort has traditionally been used to assess the semantic coherence of created texts. Word embedding is a method for representing words with vectors so that machine learning algorithms can process numeric representations of textual words. The angle between vectors A and B is used to calculate the similarity. The cosine similarity value goes from -1 to 1. A similarity of 1 indicates that the two texts are identical, whereas a similarity of -1 indicates that they are completely opposed. Results analysis shows the validation and training accuracy with gradient adjustment of neural networks, with the proposed technique achieving the best outcomes.

4.1 Discussion

The fundamental characteristic of college English online teaching is that it places the students in the centre, taking into account all of their demands as well as providing them with convenient and intelligent teaching services. As a result, any college English online teaching method must do everything possible to satisfy students' individual learning needs and provide timely feedback and evaluation. In the college English online teaching model, capabilities such as automatic homework correction, online question responding, speech recognition, evaluation, and data gathering and analysis can all be useful. DL is directly linked to the realisation and optimization of these

functions. The DL method suggested in this research, which combines a clustering algorithm with a NN, may be utilised to continually optimise these functions required by the college English online teaching method. Results of our earlier quantitative and qualitative studies were then validated through an experiment. Two females and two males were among participants. Four upper-level graduate students with a background in educational methods were recruited for the project. Participants were invited to complete a survey that consisted of ten postreply pairs (1) whether they believe a response came from a human or a machine, (2) whether a response gave information, emotional support, or social support (3) a 10-point Likert scale to indicate the quality of a response in terms of syntax, readability and consistency with conversation setting. Participants were given the option of selecting various support categories for a response if applicable, or none if no obvious support could be found. Before the experiment, participants were given a definition and examples of 3 categories of support.

5 Conclusion

This research proposes a novel technique in English teaching based on AI. This study aims to see how NLG may be combined with DL methods to give automatic support for students in English discussions. The degree and effectiveness of using AI applications to teach English have been measured. It has been completed the conception for the use of AI applications in the teaching of English. Basics, content, objectives, strategies, processors, and assessment techniques would all be included. Then, to classify the English teaching-based data collected from 7,066 students, with a large quantity of 42,307 discussion forums using Character-level recurrent neural network with back Propagation neural network (Cha_RNN_BPNN). The experimental analysis shows the comparative analysis in terms of Word Perplexity, F-K Grade Level for Readability, Cosine Similarity for Semantic Coherence, gradient change of NN, validation accuracy, and training accuracy based on classification evaluation.

References

- [Artemova et al. 2021] Artemova, E., Apishev, M., Sarkisyan, V., Aksenov, S., Kirjanov, D., & Serikov, O.: “Teaching a Massive Open Online Course on Natural Language Processing”, arXiv preprint arXiv:2104.12846.
- [Boom et al. 2019] De Boom, C., Demeester, T., & Dhoedt, B.: “Character-level recurrent neural networks in practice: comparing training and sampling schemes”, *Neural Computing and Applications*, 31(8), (2019), 4001–4017.
- [Caterini, 2017] Caterini, A.: “A novel mathematical framework for the analysis of neural networks”, Master's thesis, University of Waterloo.
- [Diao and Hu, 2021] Diao, L., & Hu, P.: “Deep learning and multimodal target recognition of complex and ambiguous words in automated English learning system”, *Journal of Intelligent & Fuzzy Systems*, 40(4), (2021), 7147–7158.
- [Ding et al. 2021] Ding, H., Chen, Y., & Wang, L.: “College English Online Teaching Model Based on Deep Learning”, *Security and Communication Networks*, 2021.

- [Galanis et al. 2021] Galanis, N. I., Vafiadis, P., Mirzaev, K. G., & Papakostas, G. A.: “Machine Learning Meets Natural Language Processing-the Story so Far”, In IFIP International Conference on Artificial Intelligence Applications and Innovations,(2021), (pp. 673-686). Springer, Cham.
- [Hwang and Sung, 2017] Hwang, K., & Sung, W.:”Character-level language modeling with hierarchical recurrent neural networks”, In 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), (2017), (pp. 5720-5724). IEEE.
- [Lauriola et al. 2022] Lauriola, I., Lavelli, A., & Aiolli, F.: “An introduction to deep learning in natural language processing: models, techniques, and tools”, *Neurocomputing*, 470, (2022), 443–456.
- [Li and Xing, 2021] Li, C., & Xing, W.: “Natural language generation using deep learning to support MOOC learners”, *International Journal of Artificial Intelligence in Education*, 31(2), (2021), 186–214.
- [Litman, 2016] Litman, D.: “Natural language processing for enhancing teaching and learning”, In: Thirteenth AAAI conference on artificial intelligence.
- [Salah et al. 2010] Salah, M. B., Mitiche, A., & Ayed, I. B.: “Multiregion image segmentation by parametric kernel graph cuts”, *IEEE Transactions on Image Processing*, 20(2),(2010), 545–557.
- [Schmidhuber, 2015] Schmidhuber, J.: “Deep learning in neural networks: An overview”, *Neural networks*, 61(2015), 85–117.
- [Su et al. 2019] Su, Z., Miao, L., & Man, J.: “Artificial intelligence promotes the evolution of English writing evaluation model”, In IOP Conference Series: Materials Science and Engineering (Vol. 646, No. 1, p. 012029), (2019) IOP Publishing.
- [Sun et al. 2020] Z. Sun, M. Anbarasan, and D. P. Kumar, “Design of online intelligent English teaching platform based on artificial intelligence techniques,” *Computational Intelligence*, vol. 37, no. 3(2020), pp. 1166–1180.
- [Sun et al., 2021] Sun, Z., Anbarasan, M., & Praveen Kumar, D.: “Design of online intelligent English teaching platform based on artificial intelligence techniques”, *Computational Intelligence*, 37(3),(2021), 1166–1180.
- [Torfi et al. 2020] Torfi, A., Shirvani, R. A., Keneshloo, Y., Tavaf, N., & Fox, E. A.: “Natural language processing advancements by deep learning: A survey”, *arXiv preprint arXiv:2003.01200*.
- [Wang et al. 2020] Wang, D., Su, J., & Yu, H.: “Feature extraction and analysis of natural language processing for deep learning english language”, *IEEE Access*, 8, (2020), 46335–46345.
- [Wang, 2019] Wang, R.: “Research on artificial intelligence promoting English learning change”, In 3rd International Conference on Economics and Management, Education, Humanities and Social Sciences (EMEHSS 2019), (2019), Atlantis Press.
- [Yang et al. 2019] Yang, H., Luo, L., Chueng, L. P., Ling, D., & Chin, F.: “Deep learning and its applications to natural language processing. In *Deep learning: Fundamentals, theory and applications*”, Springer, Cham, (2019), (pp. 89–109).