Journal Impact Factor (JIF)™ 2023: 0.2 | 5 Year JIF: 0.2 | Quartile: Q2 | *Source: JCR™ 2024 (Clarivate Analytics)*

RUPKATHA JOURNAL
ON INTERDISCIPLINARY STUDIES IN HUMANITIES

AESTHETIX MEDIA

ISSN 0975-2935 | Indexed in the Web of Science Core Collection™ Emerging Sources Citation Index (ESCI)

https://doi.org/10.21659/rupkatha.v16n2.04| Volume 16, Number 2, 2024 | Dynamic Impact Dimensions

Research Article

Detection of English Grammatical Errors and Correction using Graph Dual Encoder Decoder with Pyramid Attention Network

Hema M[1]*, Kandasamy Sellamuthu[2], Vijayarajeswari R[3]

[1]Department of English, KPR Institute of Engineering and Technology, Coimbatore, India. *Corresponding author.

[2]Department of CSE, KPR Institute of Engineering and Technology, Arasur, Coimbatore-641407, India.

[3]Department of Computer Science and Engineering, Velalar College of Engineering and Technology, India.

Abstract

In English, grammatical errors pose a significant challenge, prompting the exploration of diverse detection and correction methods. Existing approaches, however, often fall short of delivering satisfactory results and achieving high accuracy. An innovative solution, the Optimized Graph Dual Encoder Decoder with Pyramid Attention (OGDED-PA), is introduced to overcome these limitations. The model utilizes the C4_200M synthetic dataset for input data, followed by preprocessing and applying hybrid Squared Root of Term Frequency Variants with Mean Semi-absolute Deviation Factors for morphological feature extraction. Bidirectional long short-term memory with conditional random field segmentation is employed, and OGDED-PA, integrating a dual encoder-decoder architecture and pyramid attention mechanism, is then applied. This model aims to enhance accuracy in identifying and correcting grammar, syntax, punctuation, and spelling errors by capturing intricate linguistic patterns. The graph-based representation leverages Improved Border Collie Optimization (IBCO) to optimize the weight parameter, allowing the model to analyze syntactic and semantic relationships and address a broad spectrum of grammatical errors. The proposed method is implemented using the Python platform. Compared to existing methods, the proposed approach achieves 99.3% accuracy, 98.7% precision and 98.6% $F_{0.5}$.

Keywords: English grammatical error detection and correction, Morphological features, Pyramid attention mechanism, Improved Border Collie Optimization, Dual encoder and decoder.

Conflicts of Interest: The authors declared no conflicts of interest.

Funding: No funding received.

Article History: Received: 08 March 2024. Revised: 27 June 2024. Accepted: 28 June 2024. First published: 29 June 2024.

Published by: Aesthetix Media Services, India

Citation: Hema, M. & Sellamuthu, K. & Vijayarajeswari, R. (2024). Detection of English Grammatical Errors and Correction using Graph Dual Encoder Decoder with Pyramid Attention Network. Rupkatha Journal 16:2. https://doi.org/10.21659/rupkatha.v16n2.04

1. Introduction

Writing and reading in English poses a significant challenge for non-native English speakers. To address this, a Deep Learning (DL) model proposes an open grammar-checking method that provides deep analysis to train English grammar. With the widespread use of English (Lin et al., 2020; Hassan et al., 2022), the expansion of Natural Language Processing (NLP) techniques has played a vital part in the advancement of grammar-checking methods (He, 2021). English writing reviews encompass a comprehensive examination of content and structural aspects at a higher level, along with sentence and spelling correction and error detection (Kharmilah & Narius, 2019; Solyman et al., 2023). Common types of grammatical mistakes in English writing include preposition errors, article misuse or omission, mismatching singular and plural nouns, and errors resulting from subject-verb agreement issues (Lin et al., 2021). Automatic error correction in English writing is widely recognized as a highly challenging task (Lee et al., 2021; Zhou & Liu, 2021). The traditional Machine Learning (ML) approaches are unable to handle the complexities of multi-layered and diverse language representations (Wu et al., 2022; Rozovskaya & Roth, 2019).

The expansion of English writing error correction brings new opportunities, reducing the impact of quantity on the challenge of correcting English errors to some extent (Zhou & Quan, 2022; Zabolotskikh et al., 2021). Statistical data analysis has been widely explored as a technique for grammar checking. With the rapid development of statistical corpus linguistics, ML techniques that leverage corpora as the research basis and object have seen significant advancements (Qiu & Qu, 2019; Agarwal et al., 2020; Hong et al., 2020). To address these challenges and advance automatic English grammar correction, the proposed method plays a vital role. Leveraging the advancements in technology, science, and NLP, using processors for automatic error correction in English is now more feasible than ever before (Lee, 2020; Chen & Zhang, 2022). In terms of word segmentation processing, a Bi-directional Long Short-Term Memory with a Conditional Random Field (Bi-LSTM-CRF) approach for English word segmentation has been developed (He et al., 2024). Experimental findings indicate that this strategy enhances the effectiveness of NLP. Several approaches for English grammatical Error Detection and Correction (EDC) have been proposed. However, existing methods fall short in providing satisfactory results, resulting in a high error rate (Dashtipour et al., 2021; Hu et al., 2022). To overcome these issues and address the problem at hand, this research work aims to offer essential solutions and insights. In contrast to conventional encoders, the proposed work introduces a Graph Dual Encoder Decoder with Pyramid Attention (GDED-PA) to independently capture information from the source and context sentences.

The following lists the primary contribution of the research work.

- The OGDED-PA technology aids English learners by providing an accessible solution for improving grammar proficiency.
- The Bi-LSTM-CRF segmentation process automatically identifies and corrects various grammar errors, requiring less training time than large transformer models.
- The decoder utilizes a graph structure for efficient encoder output integration, incorporates a pyramid attention mechanism, and introduces IBCO for optimized weight parameters of GDED-PA, enhancing word prediction through dynamic extraction techniques.

IBCO develops a Border Collie-inspired metaheuristic algorithm for efficient exploration, balancing exploration/exploitation, converging to high-quality solutions, and adapting to diverse optimization problems.

The remainder part of the research is explained in below section. Section 2 explains the review of various recent technologies; The proposed methodology is demonstrated in section 3; Section 4 discuss about the results and discussion; The summary of this research is explained in section 5.

2. Related work

A few recent studies related to English grammar EDC are reviewed in this section,

In 2022, Hu et al. optimized neural network-based grammar correction, reducing English grammar errors. Their work integrates grammar correction requirements with a neural network algorithm, examining feature impacts on article error correction and focusing on term vector structures. In 2022, Zhang proposed a novel approach using a Seq2Seq model and feedback filtering to correct English grammar errors. The method combines a Seq2Seq model with attention and a response filtering model, enhancing both tolerance and accuracy in grammar error correction.

In 2022, He et al. showcased an ML-based grammar error detection system for English composition. The approach incorporates an ML error-reducing module and a multilayer rule error-reducing module, providing swift and objective feedback to enhance students' awareness of grammatical errors in English study. In 2021, Wu and Pan suggested an English grammar error finding depending on the LSTM-CRF ML type to notice and examine English grammar. Afterwards, a Neural Network (NN) approach and prediction model were used to analyze and study English grammar. It also included a brief overview of the NN and deep learning algorithms' growth trends.

In 2022, Wang et al. introduced DL research on identifying grammatical errors, addressing resource scarcity for English education through internet technology. The proposed approach evaluated verb, noun, article, and preposition components using various models. In 2023, Zhang et al. proposed a model for English grammatical error correction, utilizing multiple hypotheses interaction and self-attention mechanisms to assess quality. A computational network for English grammar correction was proposed by Wu in 2022. Recurrent Neural Network (RNN) is summarized by this approach. The NN is more capable of fitting the data distribution. In 2022, Zeng proposed an intelligent English writing test using semantics and neural networks, addressing grammatical challenges through confusion sets like prepositions and articles. Sequence annotation with the help of the sequence is necessary for this procedure. In 2024, Li et al. devised a method to enhance the English composition system, reducing teacher workload and effectively correcting errors, though it doesn't detect syntax and grammar errors efficiently. In 2024, Al-Khalifa et al. created a neural machine translation to examine the error patterns and estimate the system in a parallel corpus of English–Arabic sentences. One of the issues is it cannot effectively detect the error in grammar. Table 1 displays the comparison of the existing methods.

Table 1: Comparison of related works

Author name with reference	Techniques	Merits	Limitations
Hu et al. (2022)	Neural network	The suggested method minimizes grammatical errors.	Accuracy of the error correction is quite complex.
Zhang et al. (2022)	Seq2seq model	The seq2seq model increases the effectiveness of reducing grammar errors.	It requires better model to achieve error correction efficiency.
He and Maia Det al. (2021)	Machine learning	Error detection through the analysis and comparison of various algorithmic models.	More English grammar rules are needed to improve grammar error correction.
Wu and Pan et al. (2022)	LSTM-CRF	Improve the efficiency of grammar detection.	Significant changes in the accuracy.
Wang & Zhang et al. (2023)	Deep learning	DL technology used to detect grammar error quickly and efficiently.	Accuracy of the error correction is dropped low.
Zhang (2022)	Machine translation model self-attention mechanisms	Errors detection through analysis and word generation.	Effectiveness of error correction is complex.
Wu et al. (2022)	Computational Neural Network	Neural network technologies are utilized to detect and correct college English grammar errors.	The prediction of error is not deep enough and less improvement in error correction.
Zeng et al. (2021)	Neural network	Grammatical faults in English writing is corrected by NN, it Detects Automatically and Correct (DAC) error in grammar.	Noise occurred in decoding and overcorrection issues are arise.
Li, et al. (2024)	Improved Method	Employed to lessen the workload of teachers and enhance the effectiveness of the English composition grading system.	grammar errors and syntax errors cannot be effectively detected.
Al-Khalifa, et al.	Neural machine translation	It was used to evaluate the error patterns in state-of-the-art PLMs while translating from English to Arabic	Detection of grammar error is quite complex

The comparison in Table 1 demonstrates how inaccurately the current technique corrects and detects grammatical errors.

Problem Statement

The discussed methods faced limitations, notably in error rates affecting overall detection accuracy. While the current approach rectifies coronal and prepositional errors, it addresses a limited error range. Neural machine translation surpasses in handling long-distance dependencies, enhancing EDC model performance, yet faces training complexity challenges. This research targets English grammatical error detection and correction, aiming to improve precision and reliability. The OGDED-PA model enhances correction accuracy using a dual encoder-decoder architecture, pyramid attention mechanisms, and a graph-based representation for advanced syntactic and semantic understanding. Overall, OGDED-PA innovates traditional grammatical error correction with its unique architecture and attention mechanisms.

3. Proposed Methodology

The introduced technique for identifying and fixing grammar errors in English. This technique is employed to identify and rectify grammatical errors in English. The block diagram for the proposed technique is shown in Figure 1. The synthetic dataset C4-200M is the source of the input data. An NLP-based preprocessing step is used to improve the input data quality. The most significant features are extracted by hybrid Squared root of Term Frequency Variants with Mean Semi absolute Deviation Factors (STFV-MSDF). Bi-LSTM-CRF is used for segmentation. IBCO is utilized to enhance the performance of GDED-PA.

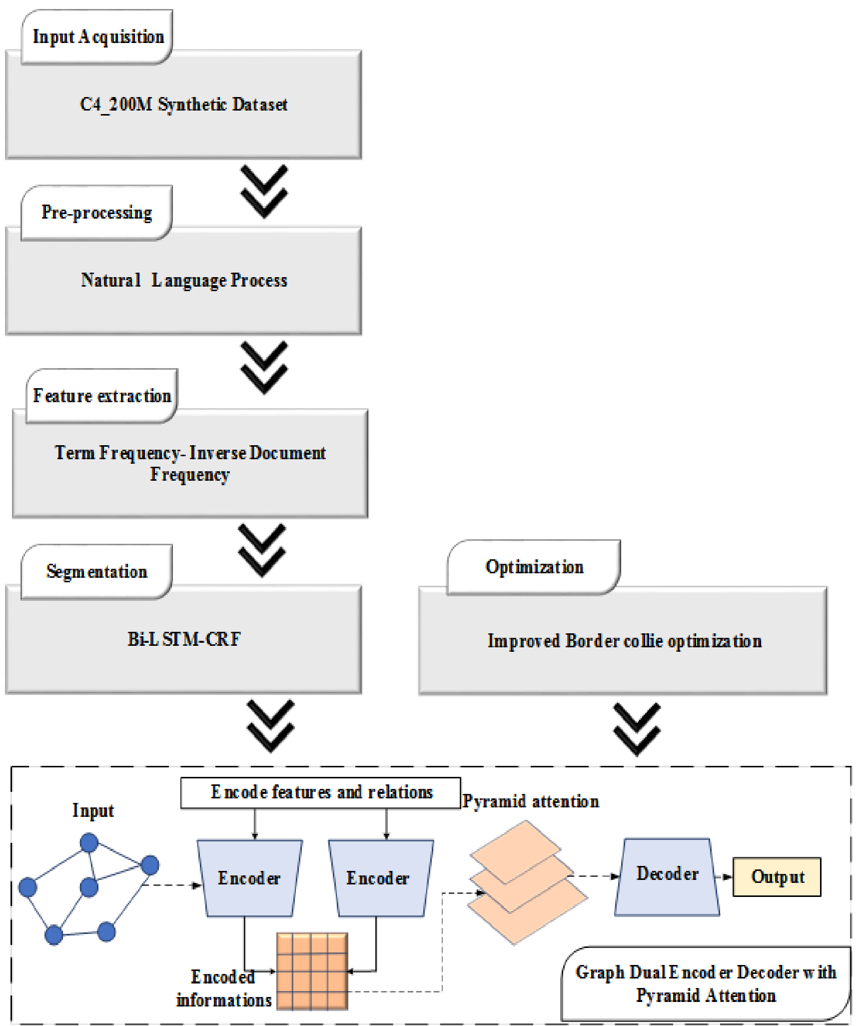


Figure 1: Work flow of introduced method

3.1 Pre-Processing

This stage is essential within the grammatical error correction model. This initial phase involves the analysis and processing of corpus information to furnish foundational data for constructing subsequent models. At this stage, it is imperative to delineate the text tokenization, stop-words removal: phrase tagging sets, the Part-of-Speech (PoS) tagging set and the word stemming corpus information are utilized for phrase partition. Once these prerequisites are met, corpus pre-processing ensues, encompassing tasks such as sentence and PoS tagging, word segmentation, counting the word frequency, and analyse the syntax (Zhang, 2022). Overall, pre-processing stage lays the foundation for subsequent modelling and analysis tasks in grammatical error correction. After preprocessing, features from the data are extracted by hybrid Squared root of Term Frequency Variants with Mean Semi absolute Deviation Factors (STFV-MSDF).

3.2 STFV-MSDF based feature extraction

The STFV-MSDF approach is utilized in this manuscript to extract the morphological features; this method takes into account the importance of words within a document relative to a larger corpus. When applied to English grammar EDC, STFV-MSDF helps the system understand the context of specific words within the text. Initially, normalized frequency term $NFT(T_j, D_i)$ (Murty et al., 2023) is computed in order to avoid biasing for lengthy documents by equation (1)

$$NFT(T_j, D_i) = \frac{TF(T"_j, D"_i)}{\sum_{T_w \in D_i} TF(T"_q, D'_i)} \tag{1}$$

where $TF(T"_j, D"_i)$ signifies the occurrence number of terms $T"_j$ in document $D"_i$, $T"_q$ represents the next term of document $D"_i$. After that, the square root of the normalized Term Frequency (TF) is calculated to adjust the influence of excessively high term frequency in equation (2).

$$SQ\_NFT(T_j, D_i) = SRT.\frac{TF(T"_j, D"_i)}{\sum_{T_w \in D_i} TF(T"_q, D"_i)} \tag{2}$$

The mean of the square root of the normalized TF in each class, denoted as the class TF $\left(CTF(T"_j, C"_k)\right)$ is computed by equation (3):

$$CTF(T"_j, C"_k) = \frac{\sum_{i=1}^{N} SQ\_NFT(T"_j, D"_i) I(D"_i, C"_k)}{M_k} \tag{3}$$

in which, $I(D"_i, C"_k)$ represents an indicator that determines whether the document $D"_i$ belongs to class $C"_k$ or not. M_k indicates the total documents in class $C"_k$. The class TF of a term is utilized to assess the categorical data carried by the term at the TF level. Moreover, class term frequencies lack the capability to measure the discriminatory power of terms across classes. For evaluating the discrimination of terms, STFV-MSDF is developed for representing the semi mean absolute deviation among every class TF and a central reference point. The central reference point of the term $T"_j$ depending on TF, signified as $CT_t(T"_j)$, is determined by the semi mean of class term frequencies. Hence, it is expressed in equation (4) as follows:

$$CT_t(T"_j) = \frac{\sum_{k=1}^{h} CTF(T"_j, C"_k)}{h} \tag{4}$$

Finally, the developed STFV-MSDF score of $T"_j$ based on $C"_k$ is written in equation (5) as follows:

$$SADF_t\left(T"_j, C"_k\right) = \frac{1}{n}\sum_{i=1}^{n}\left|CTF\left(T"_j, C"_k\right) - CT\left(T"_j\right)\right| \quad (5)$$

where n represents the total data points, $SADF_t$ represents the semi-absolute deviation factors and CT represents the class term. The morphological characteristics are extracted by applying equation (5). The data is sent to Bi-LSTM-CRF for the segmentation procedure after the features have been extracted.

3.3 Bi-LSTM-CRF For Segmentation Process

Following feature extraction, the Bi-LSTM-CRF facilitates word segmentation, excelling in identifying grammatical error boundaries. Its comprehensive process includes sentence segmentation, word frequency computation, syntactic analysis, and part-of-speech tagging. Pinpointing error locations ensures precise corrections, avoiding overcorrection or omissions. Highly effective, it addresses various grammar errors, like verb tense, spelling and subject-verb agreement, enhancing overall correction accuracy. Bi-LSTM-CRF is characterized by three specialized layers: (1) bidirectional LSTM neural network layer and (2) CRF tag decoder layer

Bi-LSTM layer: The embeddings of all the context characters are grouped into a single vector as input of both forward and backward LSTM neural networks. The matrix of scores is obtained by sending the input into a series of hidden layers.

CRF layer: In this, evaluation score is obtained by defining transition score with the score of matrices from the aforementioned method. Using the chain CRF modelling the input sequence and output label. The one with maximum evaluation score is the predicted label from system.

- **BI-LSTM-CRF**

Bi-LSTM and CRF systems are merged together to form this model. In this architecture, sentence-level tag information is combined with past and forward information using two Bi-LSTM layers and a CRF layer.

Next, consider the score matrix produced by the network, $f_\theta([X_1^T])$. The network's scoring output with parameters θ for the phrase X_1^T for the i^{th} tag, at t^{th} word is represented by the component $[f_\theta]_{i,t}$ the matrix. When a location shifts from the i^{th} and t^{th} position, the evolution score $[M]_{i,j}$ is displayed. The new network parameter is then $\tilde{\theta} = \theta \cap \{[A]_{i,j} \forall i, j\}$. Equation (6) yields the scores of a sequence of labels i_1^T and a track of $[X]_1^T$.

$$S(X_1^T, [i]_1^T, \tilde{\theta}) \sum_{t=1}^{T}([M]_{[i]_{t-1}}[i]_t + [f_\theta]_{[i]_t}, t) \quad (6)$$

where, $[M]_{[i]_{t-1}}$ is determined as the score matrix of the transition scores from the $(t-1)$ and $[i]_t$ denotes the tag at time t . To efficiently compute the algorithm, a program is used $[M]_{i,j}$ and optimal tag sequences for inference. The segmented data is sent to GDED-PA for error detection following the word segmentation procedure.

3.4GDED-PA for English grammatical error correction and detection

Following the word segmentation process, the GDED-PA method is used for the segmented data. To design a dual-encoder and dual-decoder structure named GDED-PA.

Graph Dual Encoder and Decoder (GDED): The model's encoder consists of the context information encoder, utilizing the Attention mechanism for broader context focus, and the Graph

Convolution Layers (GCL) encoder. GCL encodes the source sentence bidirectionally, capturing comprehensive semantic information. Subsequent sections elaborate on these components.

Context Encoder: Inspired by the transformer, the context information encoder resolves long-distance dependency using the attention mechanism. Unlike fixed-size windows, it minimizes distance calculation, effectively processing lengthy sentences with nuanced semantic information. Structured like the transformer, it relies on pyramid attention for extracting essential contextual information, as illustrated in Figure 2. The bidirectional encoding by the graph autoencoder ensures comprehensive semantics of the source sentence. The dual decoder utilizes extensive contextual information for accurate predictions, addressing variable length issues. Pyramid attention and correlation extraction with the graph encoder refine the final target sequence. The input of the graph encoder contains M sentences that is $R = R_1 + R_2 + R_3 \ldots\ldots R_M$. For the L^{th}sentence the number of tokens is $|R_L|$, that is $R_L = R_{L,1} \ldots. R_{L,|R_L|}$, assuming the probability of correcting the target is $V_L = v_{l,1}, \ldots v_{l,|V_L|}$. In the model, a context information encoder structure is employed, and the representation of the hidden layer state is illustrated by equation (7).

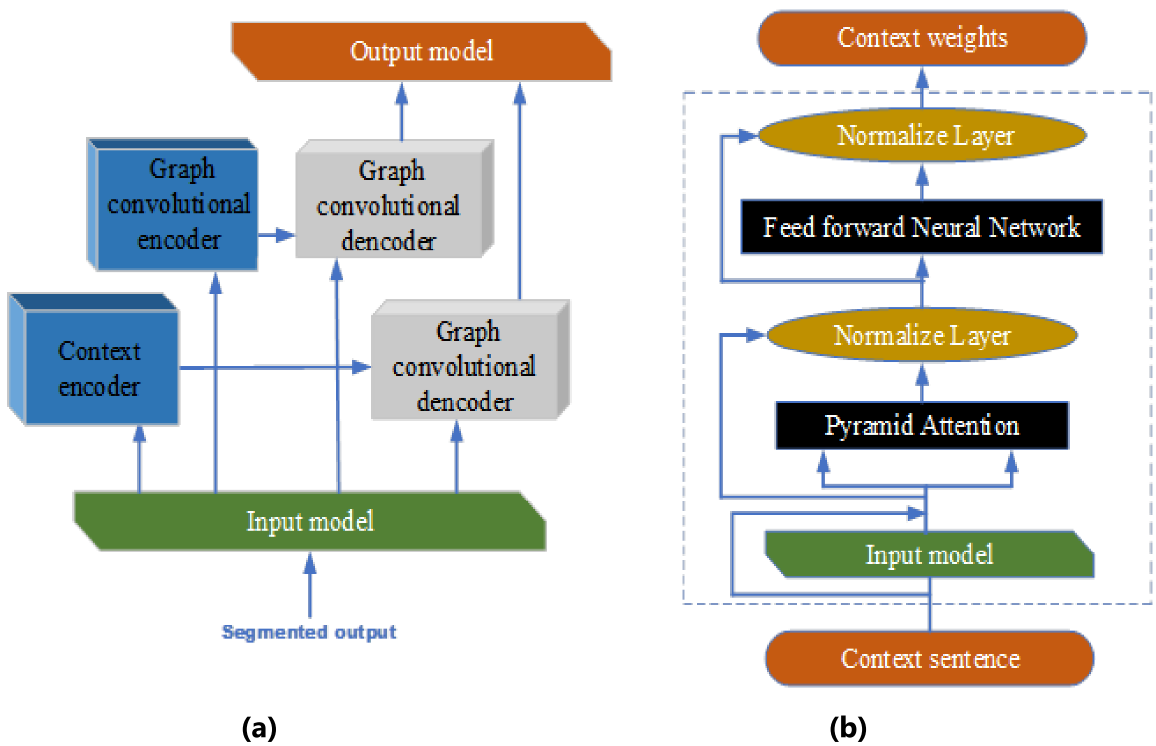


Figure 2: GDED-PA (a) structure of GDED-PA (b) Context information encoder structure.

$$C_m = trans_{enc}(C_{m-1}) \forall m \subset [1,2] \quad (7)$$

where, $C_m$is hidden layer, $trans_{enc}(C_{m-1})$ represents the transformer encoder state of the forward layer, m is a variable denoting the layer index count within the context information encoder structure

GCL: The GCL encoder is utilized in English grammatical error correction and detection to enhance contextual understanding. It refines the correction process by capturing intricate language dependencies. The formal expression for the graph encoder, denoted as a_0, is given by equation (8)

$$a_0(Y,B) = G_{C_2}\left(G_{C_1}(Y,B),B\right) \tag{8}$$

where, $Y$is feature matrix, B is the adjacency matrix, and G_C is graph convolution.

Pyramid attention: Pyramid attention is an attention mechanism that operates at multiple scales or levels. In encoder-decoder architecture, pyramid attention involves capturing information from different levels of abstraction or hierarchical representations are given in equation (9).

$$x^i = \frac{1}{\partial(y",F)} \sum_{u\in F} \sum_{j\in F} \varphi"\left(y^i,u^j\right) \theta"\left(u^j\right) \tag{9}$$

where, F represents the feature pyramid, ∂ represents the scalar function, $y"$ represents the input features, $\varphi"$ represents the pair-wise affinity between input features and pixel-wise features $y^i andu^j$, $\theta"$is a feature transformation function, $iandj$are the index of input features and pixel-wise features.

Decoder: The GCL decoder, using graph-based convolutional layers, improves grammatical error correction by analysing linguistic structures, capturing contextual dependencies, and enhancing language correction model accuracy. The representation u aims to closely resemble the original matrix B, ensuring flexibility and efficiency in decoder selection. The inner product of u measures node similarity, indicating likely edges between highly similar nodes. The reconstructed matrix B' is calculated by equation (10).

$$B' = \sigma(u,u^T) \tag{10}$$

where, u is the node representation obtained from the encoder, u^T denotes the transpose of the vector u, σ represents the nonlinear function. In decoding, pyramid attention is applied to the input, obtaining post-mask attention weights. The GCL encoder, context information encoder, and gating mechanism extract correlation details, generating the target sequence. Computational steps, detailed with a formula, include pyramid attention and a normalization layer in the decoder. Then the final output at time t is contingent on the decoder's output before time$t-1$, as depicted by equation (11).

$$Output = \eta Normalizedlayer(Pyramid\ \ attention(M_{t-1}) \tag{11}$$

where $\eta$is represented as the weight parameter of attention, M_{t-1} represents the input text with decoder time constant, equation (11) represents the classification output for classifying text and non-text data from the input text. While classifying the text data, some errors occur in the output that reduce the accuracy. To improve the accuracy, the weight parameter η of the GDED-PA is improved with the support of IBCO.

3.5Optimizing GDED-PA using Improved Border Collie Optimization

In this section, the IBCO algorithm is used to improve GDED-PA accuracy. One of the new algorithms for enhancing GDED-PA accuracy is the IBCO (Sheng et al., 2023). Border Collies, highly active herding dogs, not only obediently follow commands but also showcase reasoning skills. Their herding techniques efficiently minimize node distances. Figure 3 gives the IBCO flowchart and the Border Collie optimization procedure.

Step 1: Initialization: First the original variables of the IBCO is initialized for optimizing the weight parameters of GDED-PA in equation (12).

$$P_{m,m''} = \left(p_m^{low} p_m^{upp}\right) + \left(p_{m''}^{n-3}\right) \tag{12}$$

where p_m^{low} and p_m^{upp} represents the upper and lower boundaries of m^{th} individuals, $P_{m,m''}$ indicates that the initialization of a variable in the context of the IBCO, then $p_{m''}^{n-3}$ represents the n individuals for 3 dogs and one sheep of IBCO.

Step 2: At Random Creation: Following that, the initialization process, the input parameters of IBCO are randomly generated.

Step 3: Compute the Fitness function: The fitness function is used to derive the objective function in equation (13).

$$Fitness\ Function = \min\left(\ \eta Pyramid\ \ attention(M_{t-1})\right) \tag{13}$$

Step 4: Location searching of IBCO: Three dogs and three sheep's locations in IBCO are initialised using random variables. From one location in the field, they move to the farm. The factors of velocity, acceleration, and time govern the direction and distance travelled by dogs and sheep, which can be computed in equations (14) and (15), respectively.

$$V'_{ld,rt,lt}(t''+1) = \sqrt{V'_{ld,rt,lt}(t'')^2 2 \times A_{cc_{ld,rt,lt}}(t'') \times P_{os}{'}_{ld,rt,lt}(t'')} \tag{14}$$

where, $V'_{ld,rt,lt}(t''+1)$ is denoted as the speed of dogs at the time $(t''+1)$, $A_{cc_{ld,rt,lt}}(t'')$ represents the acceleration of the main dog, right dog and left dog at time (t'') and $P_{os}{'}_{ld,rt,lt}(t'')$ represents the location at time (t'').

$$V'_{sh}(t''+1) = \sqrt{V_{lt}(t'')^2 + 2 \times A_{cc_{lt}}(t'') \times P_{os}{'}_{lt}(t'')} \tag{15}$$

Where, $V'_{sh}(t''+1)$ represents the velocity sheep at time$(t''+1)$, $A_{cc_{lt}}$the acceleration of the left dog at time t'', $P_{os}{'}_{lt}$represents the location of the left dog at time.

Step 5: Updating the position of IBCO: The three dogs' movements control the exploring potential of the IBCO algorithm. Therefore, they are able to identify the most promising regions in the search space. The sheep's movements are directly influenced by the three dogs. Thus, they focus on devising more efficient methods of searching in dog-populated areas. The position updating of sheep and dog are calculated in below equations (16) and (17).

$$P_{os}{'}_{ld,rt,lt} = V'_{ld,rt,lt}(t''+1) \times Tim_{ld,lt,rt}(t''+1) + \frac{1}{2} A_{cc_{ld,rt,lt}}(t''+1) \times Tim_{ld,lt,rt}(t''+1)^2 \tag{16}$$

$$P_{os}{'}_{s} = V'_s(t''+1) \times Tim_s(t''+1) + \frac{1}{2} A_{cc_s}(t''+1) \times Tim_s(t''+1)^2 \tag{17}$$

From equations (16) and (17), the distance among the nodes is decreased with the help of position updating IBCO, $Tim_{ld,lt,rt}(t''+1)$ represents time required for main dog, right dog and left dog at time (t''^{+1}) and $Tim_s(t''+1)$ represents the time required by eyed sheep to move to position $P_{os}{'}_{s}$

Step 6: Termination: Here the IBCO is utilized to improve the weight parameter of GDED-PA to improve the accurateness. In this proposed method, the Detection of English Grammatical Error and Correction utilizing enhanced GDED-PA are established accurately. Thus, the rate of Recall, F-Score, and Precision is enhanced. Until met the termination conditions, the algorithm repeats the steps 5 to step 3 till $T = T + 1$ is met.

The C4-200M synthetic dataset serves as the initial source of input data. The quality of input data is improved by pre-processing using NLP. The most important features are extracted using a feature extraction technique based on STFV-MSDF. Bi-LSTM-CRF is utilized in segmentation. GDED-PA and IBCO are employed to develop the performance. Finally, the introduced technique enhances the precision and reliability of error detection, thereby elevating the overall effectiveness of the grammatical correction process.

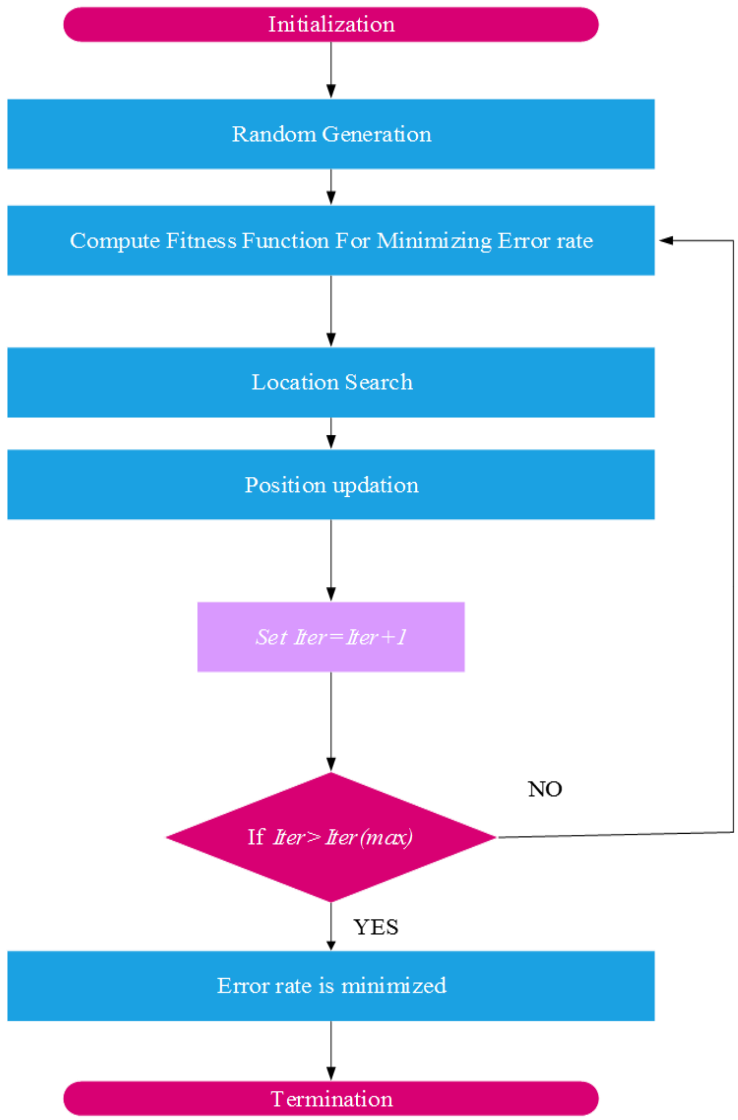


Figure 3: Flowchart for IBCO algorithm

4. Results and discussions

In this section, OGDED-PA based English Grammatical EDC is discussed. The implementation process is done in PYTHON environment. The valuation metrics are accuracy, sensitivity, precision, specificity, and $f_{0.5}$-score. And the OGDED-PA compared with current approaches such as Unifying the detection of Missing, Redundant, and Spelling Correction (UMR-Spell) (Yinghao Li et al., 2023), Template Grammatical Error Correction (TemplateGEC) (Tao Fang T et al., 2023) Multimodal Grammatical Error Correction (Multimodal GEC) (Qorib et al., 2023), Gammaticality scorer for RE-ranking Corrections (GRECO) (Stahlberg & Kumar, 2021) and Sequence transduction as a sequence of Edits (Seq2Edits) (Li, 2024) respectively.

4.1 Dataset Description

This manuscript utilizes the C4_200M Synthetic Dataset (Jin, 2023), with 200M sentences. Training involves 80%, and 20% is for testing. The dataset, consisting of approximately 200 million web documents, is a substantial resource for NLP research and large-scale language model training. Exactly designed for grammatical error correction, it employs a tagged corruption model to intentionally introduce errors into initially clean sentences from C4. Table 2 presents C4_200M Dataset Statistics.

Table 2: Statistics of C4_200M Dataset

Description	Values
Duration of sentence (Minimum)	1 word
Duration of sentence (Maximum)	7092 words
Token count in sentences that are correct	8774798
The quantity of tokens in incorrect sentences	8861928

4.2 Comparison of performance of proposed OGDED-PA with existing methods using C4_200M Synthetic Dataset

Figure 4 (a) displays the recall analysis of the proposed method. GRECO exhibits a commendable recall of 63.72%, signaling its efficacy in identifying alterations. Seq2Edits, with a recall of 40.6%, demonstrates a comparatively lower sensitivity to changes, detecting only 20% of the alterations. 40% more changes than GRECO, promising effective grammatical error correction.

The proposed OGDED-PA method boasts an impressive 99.2% recall and captures 40% more changes than GRECO. This suggests that OGDED-PA presents a promising solution for robust grammatical error correction and detection. Figure 4 (b) shows the $F_{0.5}$ analysis of the proposed method. The proposed OGDED-PA method outperforms both, attaining an $F_{0.5}$ percentage of 99.35% and successfully identifying 40% more changes than GRECO. This highlights the efficacy of OGDED-PA in achieving a balance between precision and recall, making it a promising approach for accurate English grammatical error correction and detection. Figure 4 (c) shows the precision analysis of the proposed method. OGDED-PA excels with 99.3% precision, surpassing others by detection.

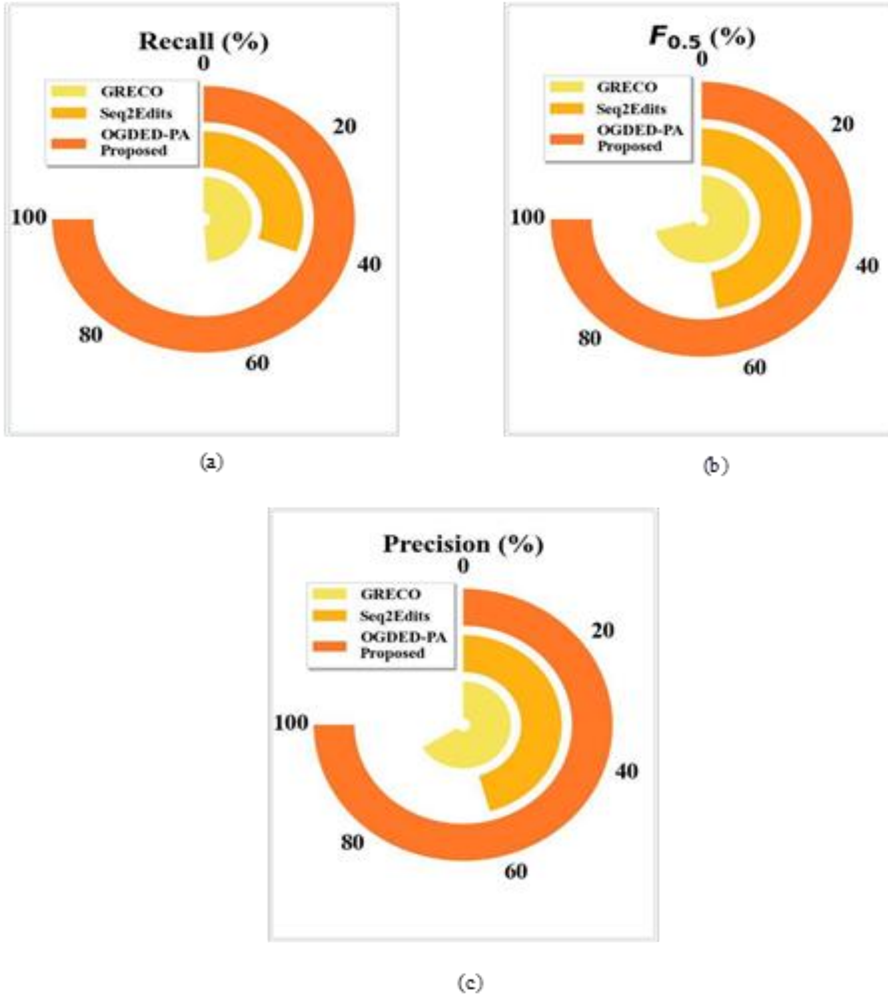


Figure 4: (a) Recall (b) $F_{0.5}$ and (c) Precision analysis

4.3 Comparison of performance of introduced OGDED-PA with previous English grammar correction approaches

Figure 5(a), reveals varying levels of precision across different approaches, where OGDED-PA stands out with the highest precision at 97%, whereas TemplateGEC exhibits the lowest precision at 68.8%. In comparison, UMR-Spell exhibits a precision of 72.2%, surpassing TemplateGEC but falling short of the precision achieved by Multimodal GEC and proposed OGDED-PA approaches. Figure 5(b) depicts the proposed method's recall analysis, demonstrating its effectiveness against existing English grammar error correction approaches. TemplateGEC has a 64.6% recall, and UMR-Spell achieves 77.2%, surpassing TemplateGEC and Multimodal GEC, but falls short of the proposed OGDED-PA, which attains an impressive 98% recall. Figure 5(c) presents an analysis of the F0.5 score for the proposed method in English grammatical error correction and detection. Notably, the proposed approach consistently outperforms existing methods, achieving a superior $F_{0.5}$ score of 96%. Comparatively, Multimodal GEC surpasses TemplateGEC but falls short of UMR-Spell and the innovative proposed OGDED-PA method, which exhibits the greatest $F_{0.5}$ score among all the approaches.

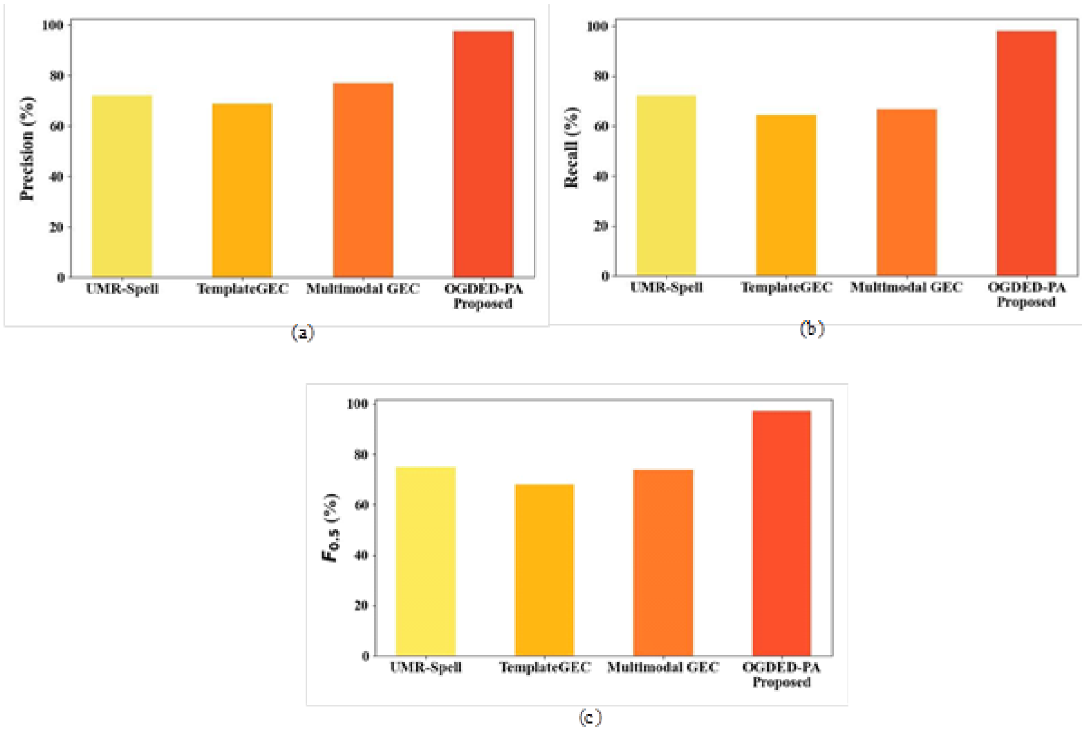


Figure 5: (a) Precision (b) Recall and (c) $F_{0.5}$ analysis

4.4 Comparison of performance of proposed OGDED-PA with Vanilla transformer

The performance analysis comparison between the vanilla transformer and the introduced technique is displayed in Figure 6 (Rokbani et al., 2021). When compared to the vanilla transformer, the proposed technique achieves a greater precision value, according to the precision analysis.

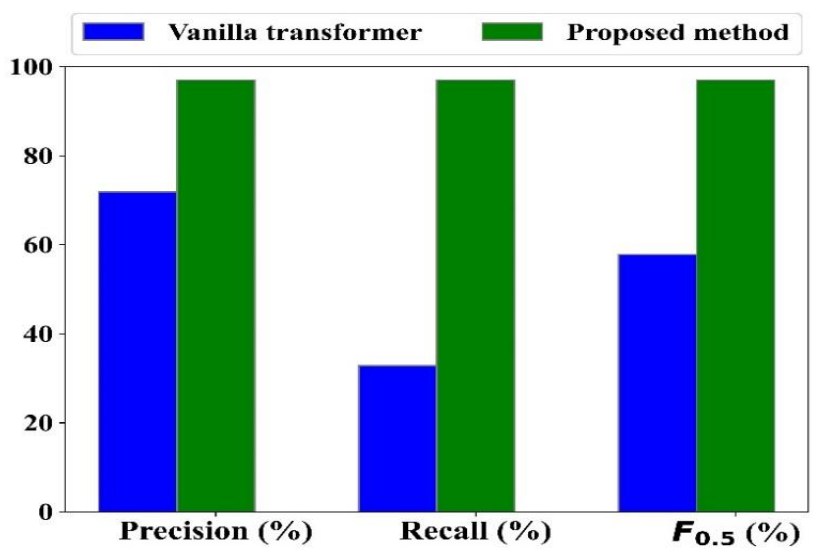


Figure 6: Comparison between vanilla transformer and proposed approach

The precision value of the vanilla transformer is measured at 71.11%, while the introduced method achieves a precision value of 97.36%. In terms of recall analysis, the introduced method outperforms the vanilla transformer with a higher recall value. The introduced method achieves a recall value of 98.24%, which is 18.68% higher than the recall value of the vanilla transformer. Furthermore, the $F_{0.5}$ analysis demonstrates that the introduced method achieves a higher $F_{0.5}$ than the vanilla transformer. This indicates that the introduced method balances recall and precision better. The $F_{0.5}$ of the introduced method surpasses the vanilla transformer in terms of overall performance.

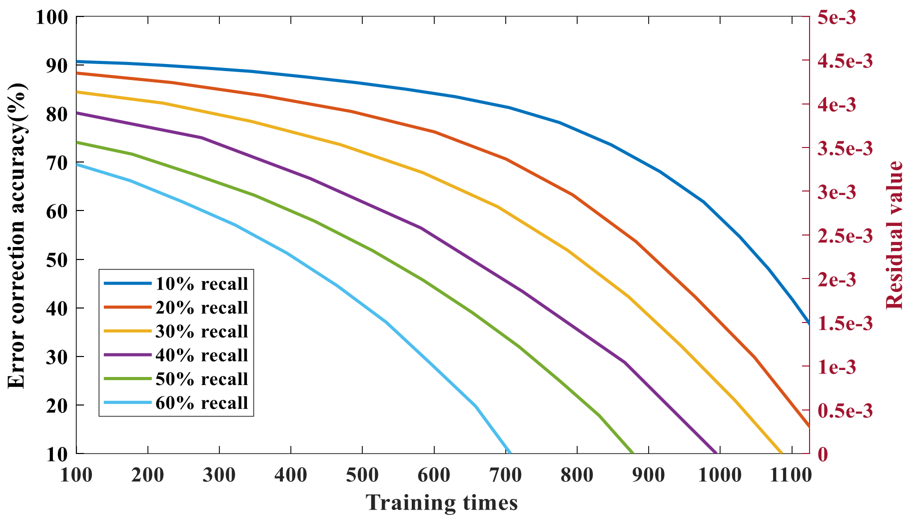


Figure 7: Accuracy analysis of error correction

In Figure 7, the error correction accuracy analysis of the proposed technique is depicted for several recall rates. At a recall rate of 10% and a training time of 700s, the error correction accuracy is measured at 70%. Increasing the recall rate to 20% and the training time to 890s, the error correction accuracy improves to 75%. At a recall rate of 30% and a training time of 1000s, the error correction accuracy reaches 80%. Further, at a recall rate of 40% and a training time of 1090s, the introduced method achieves an error correction accuracy of 85%. Notably, at a recall rate of 60% and a training time of 1100s, the error correction accuracy significantly increases to 92%. These findings indicate that the introduced method demonstrates the highest error correction accuracy for different recall rates and varying training times.

Figure 8 illustrates the convergence curve of IBCO alongside the convergence curves of other optimization algorithms such as ACO (Ant Colony Optimization) (Pozna et al., 2022), PSO (Particle Swarm Optimization (Bannò et al., 2024), and (WOA) Whale Optimization Algorithm (Sun L et al., 2022). The analysis reveals that IBCO exhibits a significantly faster convergence speed than the other optimization methods. Figure 9 (a) presents the precision analysis of the dataset used in the OGDED-PA, as well as a comparison with datasets utilized in another technique called Self-Refinement (SR). The SR approach utilizes three different datasets: EFCamDAT (Bannò & Matassoni, 2024), BEA (Wang et al., 2023), and Lang 8 (Rothe et al., 2021). When using the EFCamDAT dataset alone, the precision of the SR method is quite low. However, combining EFCamDAT with BEA leads to a slight increase in precision. When all three datasets (EFCamDAT, BEA, and Lang 8) are combined, the SR method achieves a precision of 61.7%. In contrast, the C4-

200M dataset applied in the introduced approach achieves a significantly higher precision of 97.36%. This demonstrates the effectiveness of the C4-200M dataset in achieving higher precision for English grammatical EDC.

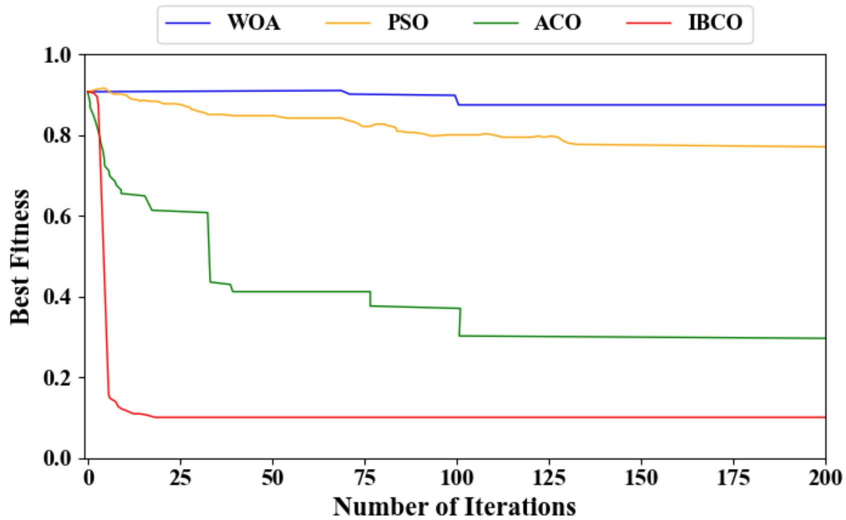
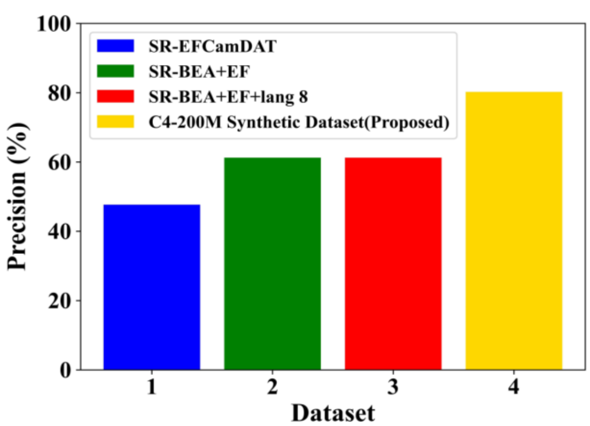


Figure 8: IBCO's Convergence Curve



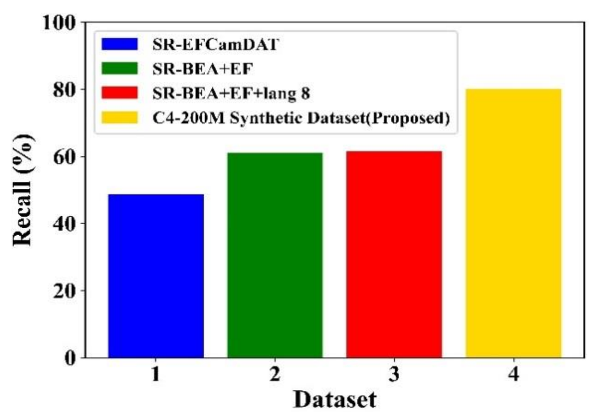


Figure 9: Analysis of C4-200M dataset (a) Precision (b) Recall

Figure 9 (b) illustrates the recall analysis of the dataset utilized in the introduced method. The SR approach using the EFCamDAT dataset achieves a lower recall value than the other approaches. However, when BEA is combined with EFCamDAT, the recall value of SR slightly increases to 38.0%. In contrast, the proposed technique achieves a recall of 98.24%, which is greater than the datasets used in the SR technique. This highlights the effectiveness of the dataset utilized in the introduced method for capturing and detecting grammatical errors.

In Figure 10, the $F_{0.5}$ analysis of the dataset used in the proposed technique is presented, showcasing the efficiency of the dataset in the proposed method. The SR-EFCamDAT dataset attains a lower $F_{0.5}$ of 40.3%, which is inferior to all the other datasets. The combination of BEA and EF achieves an $F_{0.5}$ of 54.5%, which is less than the BEA+EF+lang 8 and the dataset used in the proposed method. The BEA+EF+lang 8 dataset attains an $F_{0.5}$ of 58.8%. Notably, the dataset applied in the introduced technique attains the highest $F_{0.5}$ of 99.13% among all the datasets compared. This analysis demonstrates that the introduced approach outperforms other datasets in terms of $F_{0.5}$, emphasizing its effectiveness in English grammatical error correction and detection.

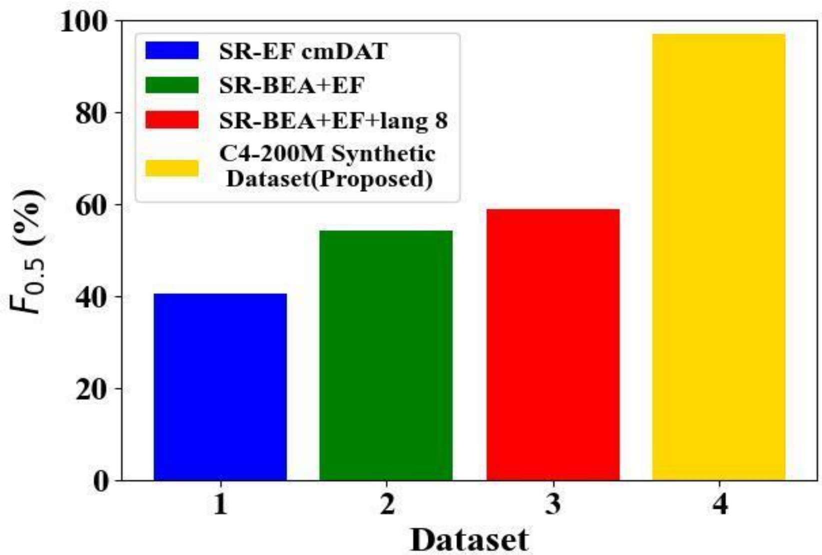


Figure 10: $F_{0.5}$ analysis of C4-200M dataset

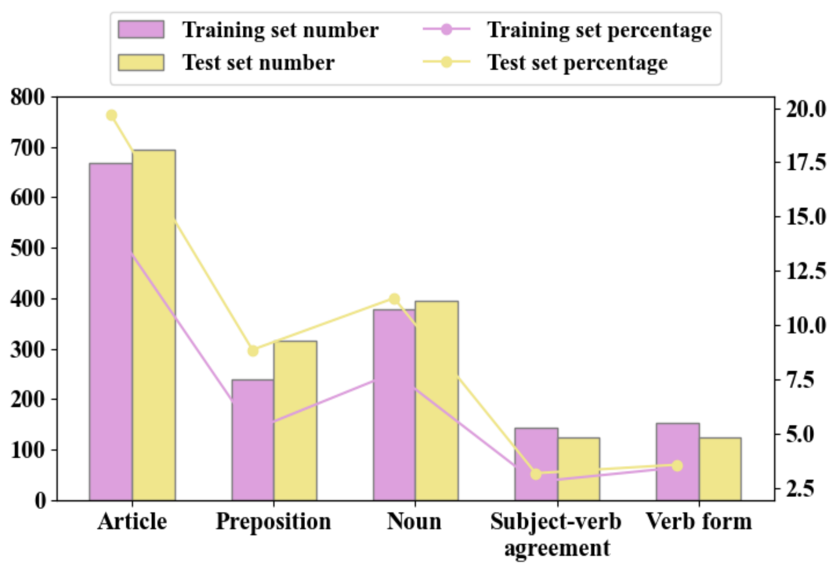


Figure 11: Statistical graph depicting the experimental results.

Figure 11 presents the statistical analysis for error types in both test and training data. These encompass errors in articles, subject-verb agreement, prepositions, names and various verb forms. The comprehensive set includes verb tense errors, possessive words, missing verbs, articles, agreement between subject and verb, plural nouns, singular nouns, and pronoun forms.

- **Statistical analysis of preposition check results**

This research's preposition checking model focuses on extracting features related to 10 commonly used prepositions, addressing attachment information issues. Evaluation includes statistical testing on each preposition, as displayed in Table 3.

Table 3. Statistical analysis for preposition check results

	Preposition									
	about	**by**	**from**	**at**	**with**	**of**	**in**	**for**	**to**	**on**
Number of PE found	16	9	9	10	8	15	8	12	13	12
Total errors	16	10	10	13	10	18	10	14	17	15
Correctly correct the PE	9	7	7	9	7	13	6	9	10	10
Recall rate (%)	93.56	94.2	90.43	92	82	95	86	87.45	88.90	92
Correct rate (%)	86.23	88.34	83.3	90	89	87	95	93	95.34	91

- **Statistical analysis of grammar check model for dataset outcomes**

The comprehensive outcomes of error correction derived from the statistical-based grammatical error correction model outlined in this study are presented in Table 2.

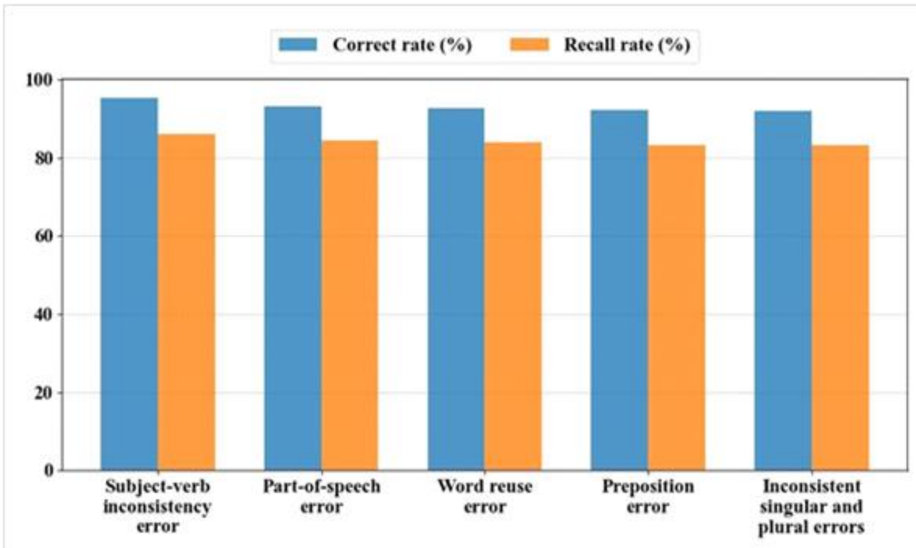
Table 4. Statistical analysis of grammar check for dataset outcomes.

Error type	**Quantity of machine findings**	**Quantity of manual findings**	**Number of alterations**	**Correct rate (%)**	**$F_{0.5}$ score (%)**	**Recall rate (%)**
AE	72	86	61	98.56	92.12	93.45
PE	100	124	79	97.28	94.57	90.45
FE	78	86	73	95.45	93.42	97.37
CSAE	75	83	62	95.35	94.43	96.45
VCE	67	80	55	94.5	94.23	93.28
RE	77	87	64	94.32	90.23	95.66
PoSE	78	92	73	96.43	90.56	93.24
VFE	77	90	71	96.42	92.45	93.45
AVE	65	81	54	95.26	99.23	96.25
ISPE	81	91	59	96.34	99.56	98.23
SVIE	75	86	57	92.33	95.54	99.26
AV	75	89	63	97.43	95.23	98.56

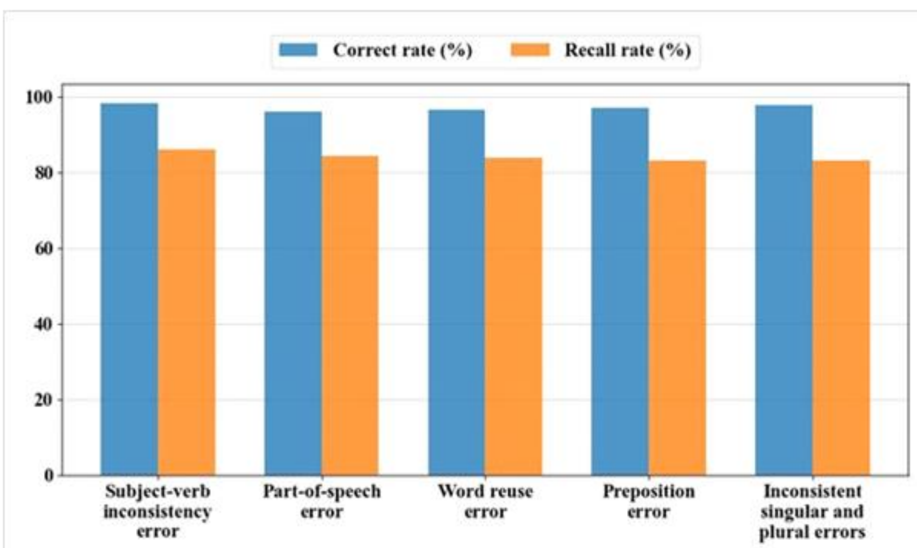
The errors in the dataset are Preposition Error (PE), Article Error (AE), Part-of-Speech Error (PoSE), Verb Form Error (VFE), Auxiliary Verb Error, Subject Verb Inconsistency Error (AVESVIE), Inconsistent Singular and Plural Errors (ISPE), Fragment Error (FE), Comparative Superlative Adjective Error (CSAE), Verb Collocation Errors (VCE), Repeat Error (RE) and Average Value(AV). While there remains considerable opportunity for enhancing the range and precision of error detection, the statistical-based grammatical error correction model has largely achieved its language-checking objectives. Furthermore, the final accuracy rate is comparatively high, underscoring its practical utility. Table 4 shows the Statistical analysis of grammar check model test data outcomes.

- **Segmentation output**

Figures 12 (a, b) show the phrase segmentation output analysis. The approach takes into account dependencies preceding and following certain words, expanding beyond word-to-word relationships and encompasses noun and verb phrases. English text undergoes part-of-speech tagging during training, followed by phrase segmentation and syntax parse tree construction. This evaluation shows the system error correction accuracy and recall rates pre- and post-phrase segmentation to assess overall performance.



(a)



(b)

Figure 12: Segmentation output (a) result before using phrase segmentation (b) result after using phrase segmentation

4.5. Ablation study

The proposed method conducts an ablation study on English grammatical EDC system components, analyzing three network models. The Generalized Language Evaluation Understanding (GLEU) metric assesses their impact on overall performance across four models in the test set. The equations for calculating GLEU are given in equation (18),

$$GLEU_{Score}(CT, ST, RT) = BQ.e\left(\sum_{m=1}^{M} X_m \, log[{}^{Q_m}/_{S_m}]\right) \tag{18}$$

where, BQ represents the brevity penalty, applied if the generated text is shorter than the reference text, X_m Weights assigned to different n-grams, Q_m represents the number of n-grams in the generated text that match the reference text, S_m Number of n-grams in the reference text, ST is the source text, RT is the standard text, and CT represents the output text. The GLEU score considers both precision (matching n-grams) and recall (n-grams present in the reference). Table 5 shows the proposed models' ablation study.

From Table 5, the proposed OGDED-PA model achieves the highest GLEU score of 0.6325. Comparing the GCL encoder and decoder, adding Pyramid attention slightly improves the GLEU score (0.4427 vs. 0.4312). This suggests that using attention mechanisms within the model helps identify grammatical errors more effectively. The OGDED-PA significantly improves the GLEU score (0.5797) compared to Pyramid attention alone. This indicates that the dual encoder-decoder architecture provides a significant advantage in identifying and correcting errors. The proposed OGDED-PA achieves the highest GLEU score (0.6325), further enhancing performance compared to the basic dual encoder-decoder. This suggests that the IBCO optimizations implemented in the proposed model provide additional benefits for accurate error detection and correction. Finally, the OGDED-PA model achieves superior performance in English grammatical error correction tasks.

Table 5. Ablation study for introduced model

Model	**GLEU**
Graph Dual Encoder and Decoder (GDED)	0.4427
Pyramid attention (PA)	0.4312
GDED-PA	0.5797
OGDED-PA (proposed)	0.6325

5. Conclusion

English grammar detection and correction using OGDED-PA has been successfully implemented in this research. The method is implemented in Python, leveraging its powerful platform for efficient execution. By incorporating IBCO, the introduced method achieves improved accuracy in error correction. Comparative analysis demonstrates the superiority of the OGDED-PA based on the sensitivity, specificity, accuracy, recall, precision and $F_{0.5}$. Furthermore, the efficiency of OGDED-PA is evaluated against the vanilla transformer, where the former outperforms the latter. In the future, a large vocabulary set can be used to improve grammatical detection and correction. Additionally, incorporating the reinforcement learning agent into established writing tools for creating independent applications can broaden its practical usefulness. This seamless integration of grammar error correction directly into the writing process enables users to receive real-time refinement and assistance, enhancing the overall writing experience.

References

Agarwal, N., Wani, M. A., & Bours, P. (2020). Lex-pos feature-based grammar error detection system for the English language. Electronics, 9(10), 1686.

Alammary, A. S. (2021). Arabic questions classification using modified TF-IDF. *IEEE Access*, *9*, 95109–95122. https://doi.org/10.1109/access.2021.3094115

Al-Khalifa, H., Al-Khalefah, K., & Haroon, H. (2024). Error Analysis of Pretrained Language Models (PLMs) in English-to-Arabic Machine Translation. Human-Centric Intelligent Systems, 1-14.

Bannò, S., & Matassoni, M. (2024). Back to grammar: Using grammatical error correction to automatically assess L2 speaking proficiency. Speech Communication, 157, 103025.

Chen, B., & Zhang, J. (2022). Pre-Training-Based Grammatical Error Correction Model for the Written Language of Chinese Hearing Impaired Students. IEEE Access, 10, 35061-35072.

Dashtipour, K., Gogate, M., Cambria, E., & Hussain, A. (2021). A novel context-aware multimodal framework for Persian sentiment analysis. *Neurocomputing*, *457*, 377–388. https://doi.org/10.1016/j.neucom.2021.02.020

Fang, T., Hu, J., Wong, D. F., Wan, X., Chao, L. S., & Chang, T. H. (2023). Improving grammatical error correction with multimodal feature integration. In *Findings of the Association for Computational Linguistics: ACL 2023* (pp. 9328-9344).

Geng, B. (2022). Text segmentation for patent claim simplification via bidirectional long-short term memory and conditional random field. *Computational Intelligence*, *38*(1), 205-215.

Hassan, A., & Abbas, A. (2022). The Impact of Written Corrective Feedback on Second Language Composition in English. In *English Language Teaching in Pakistan* (pp. 181-201). Singapore: Springer Nature Singapore.

He, H., & Maia, D. (2022). Application of Grammar Error Detection Method for English Composition Based on Machine Learning. Security and Communication Networks, 2022.

He, L., Liu, F., Liu, J., Duan, J., & Wang, H. (2024). Self-distillation and pinyin character prediction for Chinese spelling correction based on multimodality. *Applied Sciences*, *14*(4), 1375. https://doi.org/10.3390/app14041375

He, Z. (2021). English grammar error detection using recurrent neural networks. *Scientific Programming*, *2021*, 1–8. https://doi.org/10.1155/2021/7058723

Hong, J. C., Hwang, M. Y., Liu, Y. H., & Tai, K. H. (2022). Effects of gamifying questions on English grammar learning mediated by epistemic curiosity and language anxiety. *Computer Assisted Language Learning*, *35*(7), 1458-1482.

https://Github.Com/Google-Research-Datasets/C4 *200M-Synthetic-Dataset-for Grammatical-Error-Correction*.

Hu, L., Tang, Y., Wu, X., & Zeng, J. (2022). Considering optimization of English grammar error correction based on neural network. *Neural Computing and Applications*, 1-13.

Jin, L., Zhang, L., & Zhao, L. (2023). Feature selection based on absolute deviation factor for text classification. *Information Processing & Management*, *60*(3), 103251. https://doi.org/10.1016/j.ipm.2022.103251

Kharmilah, P., & Narius, D. (2019). Error analysis in writing discussion text made by students at English department of Universitas Negeri Padang. *Journal of English Language Teaching*, *8*(3), 327–335.

Lee, M., Shin, H., Lee, D., & Choi, S. P. (2021). Korean grammatical error correction based on transformer with copying mechanisms and grammatical noise implantation methods. *Sensors*, *21*(8), 2658.

Lee, S. M. (2020). The impact of using machine translation on EFL students' writing. *Computer assisted language learning*, *33*(3), 157-175.

Li, B. (2024). An improved method for extracting essay tangency features in Intelligent scoring of English essays. *Intelligent Systems with Applications*, *21*, 200328. https://doi.org/10.1016/j.iswa.2024.200328

Li, Y., Liu, X., Wang, S., Gong, P., Wong, D. F., Gao, Y., ... & Zhang, M. (2023, July). TemplateGEC: Improving grammatical error correction with detection template. In Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (pp. 6878-6892).

Lin, C. J., Hwang, G. J., Fu, Q. K., & Cao, Y. H. (2020). Facilitating EFL students' English grammar learning performance and behaviors: A contextual gaming approach. *Computers & Education*, *152*, 103876.

Lin, N., Chen, B., Lin, X., Wattanachote, K., & Jiang, S. (2021). A framework for Indonesian grammar error correction. *Transactions on Asian and Low-Resource Language Information Processing*, *20*(4), 1-12.

Muaad, A. Y., Al-antari, M. A., Lee, S., & Davanagere, H. J. (2021, September). A novel deep learning ArCAR system for Arabic text recognition with character-level representation. In *Computer sciences & mathematics forum* (Vol. 2, No. 1, p. 14). MDPI.

Murty, S., Sharma, P., Andreas, J., & Manning, C. (2023). Grokking of hierarchical structure in Vanilla Transformers. *arXiv Preprint arXiv:2305.18741.* https://doi.org/10.18653/v1/2023.acl-short.38

Pozna, C., Precup, R. E., Horváth, E., & Petriu, E. M. (2022). Hybrid particle filter–particle swarm optimization algorithm and application to fuzzy controlled servo systems. *IEEE Transactions on Fuzzy Systems*, *30*(10), 4286-4297.

Qiu, Z., & Qu, Y. (2019). A two-stage model for Chinese grammatical error correction. *IEEE Access*, *7*, 146772–146777. https://doi.org/10.1109/access.2019.2940607

Qorib, M. R., & Ng, H. T. (2023). System combination via quality estimation for grammatical error correction. *arXiv Preprint arXiv:2310.14947.* https://doi.org/10.18653/v1/2023.emnlp-main.785

Rokbani, N., Kumar, R., Abraham, A., Alimi, A. M., Long, H. V., Priyadarshini, I., & Son, L. H. (2021). Bi-heuristic ant colony optimization-based approaches for traveling salesman problem. *Soft Computing*, *25*, 3775-3794.

Rothe, S., Mallinson, J., Malmi, E., Krause, S., & Severyn, A. (2021). A simple recipe for multilingual grammatical error correction. *arXiv Preprint arXiv:2106.03830*. https://doi.org/10.18653/v1/2021.acl-short.89

Rozovskaya, A., & Roth, D. (2019). Grammar error correction in morphologically rich languages: The case of Russian. Transactions of the Association for Computational Linguistics, 7, 1-17.

Sheng, L., Xu, Z., Li, X., & Jiang, Z. (2023). EDMSpell: Incorporating the error discriminator mechanism into chinese spelling correction for the overcorrection problem. *Journal of King Saud University-Computer and Information Sciences*, *35*(6), 101573.

Wu, X. (2022). A Computational Neural Network Model for College English Grammar Correction. Computational Intelligence and Neuroscience, 2022.

Stahlberg, F., & Kumar, S. (2021). Synthetic data generation for grammatical error correction with tagged corruption models. *arXiv Preprint arXiv:2105.13318*.

Sun, L., Zeng, L., Zhou, H., & Zhang, L. (2022). Strip thickness prediction method based on improved Border Collie Optimizing LSTM. *PeerJ Computer Science*, *8*. https://doi.org/10.7717/peerj-cs.1114

Wang, H., Kurosawa, M., Katsumata, S., Mita, M., & Komachi, M. (2023). Chinese Grammatical Error Correction Using Pre-trained Models and Pseudo Data. *ACM Transactions on Asian and Low-Resource Language Information Processing*, *22*(3), 1-12.

Wang, X., & Zhong, W. (2022). Research and implementation of English grammar check and error correction based on Deep Learning. *Scientific Programming*, *2022*, 1–10. https://doi.org/10.1155/2022/4082082

Wu, F., Huang, S., & Cheng, L. (2022). Analyzing the application of multimedia technology assisted English grammar teaching in Colleges. *Scientific Programming*, *2022*, 1–11. https://doi.org/10.1155/2022/4422754

Wu, L., & Pan, M. (2021). English grammar detection based on LSTM-CRF Machine Learning Model. *Computational Intelligence and Neuroscience*, *2021*, 1–10. https://doi.org/10.1155/2021/8545686

Wu, X. (2022). A computational neural network model for college English grammar correction. *Computational Intelligence and Neuroscience*, *2022*, 1–9. https://doi.org/10.1155/2022/9592200

Wu, Y., Ma, Y., Liu, J., Du, J., & Xing, L. (2019). Self-attention convolutional neural network for improved mr image reconstruction. *Information Sciences*, *490*, 317–328. https://doi.org/10.1016/j.ins.2019.03.080

Zabolotskikh, A., Zabolotskikh, A., Dugina, T., & Tavberidze, D. (2021). Creating individual learning paths in the Moodle plugin for undergraduate students to study English grammar. *Education and Information Technologies*, *26*(1), 617-637.

Zeng, G. (2022). Intelligent test algorithm for English writing using English semantic and Neural Networks. *Mobile Information Systems*, *2022*, 1–11. https://doi.org/10.1155/2022/4484201

Zhang, A. (2022). Analysis of the Application of Feedback Filtering and Seq2Seq Model in English Grammar. *Wireless Communications and Mobile Computing*, *2022*(1), 9530379.

Zhang, C., Xu, T., & Wu, G. (2023). Neural Quality Estimation Based on Multiple Hypotheses Interaction and Self-Attention for Grammatical Error Correction. IEEE Access, 11, 8718-8726.

Zhou, Shanchun, & Liu, W. (2021). English grammar error correction algorithm based on classification model. *Complexity*, *2021*, 1–11. https://doi.org/10.1155/2021/6687337

Zhou, S., & Quan, Z. (2022). On Teaching Patterns of English Grammar under the Background of "Internet+". *Wireless Communications and Mobile Computing*, *2022*(1), 9028764.

Dr. Hema M has been an Associate Professor and Head of the English department at KPRIET for the past 10 years. Her areas of interest are Indian Writing in English and English Language Teaching.

Dr. Kandasamy Sellamuthu is an Assistant Professor (Sr. G) in the Department of Computer Science and Engineering. She has ten years of experience in teaching. She has published 12 research papers in international journals (11 SCOPUS-indexed journals and 1 SCI) and 4 papers in international and national conferences. Her area of interest is machine learning with IoT concepts.

Dr. Vijayarajeswari R is a Professor at the Department of Computer Science and Engineering at Velalar College of Engineering and Technology, Tamil Nadu, India. Her research interests span neural networks, machine learning, image processing and security in networks. She holds first class in her Bachelor's degree in Information Technology and first class with distinction in her Master's degree in Computer Science and Engineering, both from Anna University, Chennai. She completed her Ph.D. at Anna University, Chennai, with around 15 years of teaching and research experience. She has carried out several sponsored research and consultancy projects.