Interpretable Artificial Intelligence Used To Diagnose Common Grammatical Errors in English Writing By Non-Native Engineering Researcher

Yafeng Liu*

School of English Language and Culture, Xi'an Fanyi University, Xi'an 710105, China

* *Corresponding author. E-mail: liuxiaoya1202@126.com*

Received: May 15, 2026; Accepted: Jul. 24, 2026

Grammatical error detection is an essential task in natural language processing that supports automated writing assistance and educational language assessment systems. This study proposes an explainable framework for identifying multiple categories of grammatical errors in English text. The model was evaluated using a dataset of 2,990 annotated English sentences containing both grammatically correct and erroneous structures across several error categories. The proposed framework integrates text pre-processing, Term Frequency–Inverse Document Frequency (TF–IDF) based word-level n-gram feature extraction, and a multi-class Logistic Regression (LR) classifier. The framework generates efficient numerical representations of textual data while capturing lexical, syntactic, and pragmatic characteristics for accurate grammatical error identification. To enhance transparency and interpretability, SHAP (SHapley Additive exPlanations) was employed to analyze feature importance and provide word-level explanations for classification decisions, thereby incorporating explainable artificial intelligence into grammar evaluation tasks. Experimental results demonstrate strong classification performance, achieving an accuracy of 0.9766, precision of 0.9376, recall of 0.9889, macro-average F1-score of 0.9603, and weighted F1-score of 0.9772. Furthermore, the Cohen's Kappa coefficient of 0.9667 indicates a high level of agreement between predicted and actual labels, confirming the reliability of the proposed model. The combination of TF–IDF statistical text representation and interpretable logistic regression provides an effective, computationally efficient, and explainable solution for automated grammatical error identification.

Keywords: Grammatical Error Detection; Natural Language Processing; TF–IDF Feature Extraction; Logistic Regression; Explainable Artificial Intelligence; SHAP Interpretability

http://dx.doi.org/10.6180/jase.202611_34.065

1. Introduction

Due to globalization over the last few decades, English has grown to be viewed as the international language of science [1]. Engineering researchers need grammatically correct English manuscripts for successful academic communication and publication [2]. Non-native engineering researchers often struggle to achieve full grammatical accuracy in academic writing [3]. Grammatical errors in researchers' manuscripts negatively affect their readability [4], the framework improved material reuse, reduced energy consumption and embodied carbon emissions, while providing interpretable decision support for heritage conservation rather than automated conservation decisions [5].

Numerous reasons exist as to why non-native English researchers make grammatical errors when writing in English [6]. One of the primary causes of grammatical error originates from the disparity between the native language

of the individual and the structure of English grammar [7]. The differences in language structure provide multiple areas for errors including, but not limited to, the following items: articles, subject-verb agreement, and prepositions. A non-native writer typically has had limited experience with formal academic English [8]. Non-native researchers lack scientific writing training, leading to frequent grammatical errors in research publications [9].

The proposed framework integrates preprocessing, TF–IDF, Logistic Regression, and SHAP to detect grammatical errors in non-native academic writing with improved accuracy, reliability, and interpretability.

The study proposes an interpretable grammatical error detection framework that combines TF–IDF feature extraction with a Logistic Regression classifier and integrates SHAP-based explainable artificial intelligence to provide transparent interpretation of model predictions through feature importance and word contribution analysis. Furthermore, it demonstrates that traditional machine learning methods offer a computationally efficient and effective alternative to complex deep learning approaches for grammatical error detection.

Research Organization

The remainder of this paper is organized as follows. Section 2 reviews related work, Section 3 presents the proposed methodology, Section 4 describes the experimental setup and dataset, Section 5 discusses the performance evaluation and experimental results, and Section 6 concludes the study with future research directions.

2. Literature review

Ghannam et al. [10] examined how AI-powered language aid can be used to enhance the level of writing English among non-native speakers. The study shows that automated grammar correction improves writing accuracy but emphasizes the need for explainable AI due to the limited transparency of black-box models. Li et al. [11] investigated the possibility of AI-based technologies to enhance academic writing proficiency of non-native English-speaking medical students. Their study demonstrated that conversational AI improves grammar, vocabulary, and writing style, supporting AI-based grammatical error analysis. Aladsani et al. [12] explored the use of generative technologies of artificial intelligence by non-native English-speaking researchers who engaged in academic publication. AI tools improve grammar and clarity in writing, but issues of transparency and interpretability require explainable AI models.

Algobaei and Alzain [13] proposed a generative AI-based model, which is based on personalized language feedback that is offered to non-native learners of English using prompt engineering. Their study showed that AI provides personalized feedback to improve sentence structure and grammatical accuracy, but transparent identification of specific grammatical error categories still requires structured machine learning models.

Problem Statement

Non-native engineers often struggle with academic grammar, reducing clarity and publication acceptance chances [14]. Most grammar checkers correct errors but lack transparency about specific grammar error types [15].

3. Proposed methodology

The proposed framework uses TF–IDF, Logistic Regression, and SHAP for interpretable grammatical error detection, classifying multiple error types with transparent feature analysis.

The proposed framework for grammatical error detection shows in Fig. 1. The proposed framework preprocesses text and applies TF–IDF with multi-class Logistic Regression for classification, using SHAP for interpretable word-level grammatical error predictions.

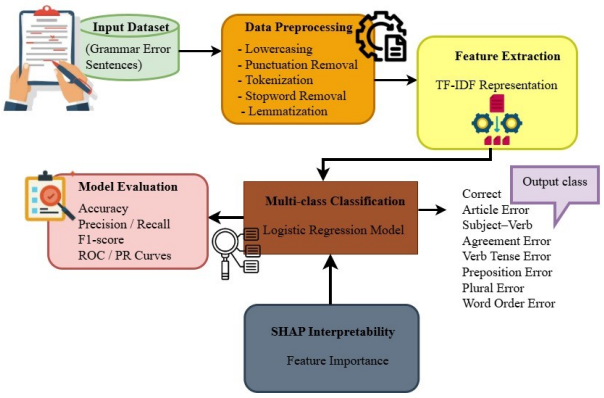

Fig. 1. Architecture of TF–IDF and Logistic Regression-Based Grammatical Error Detection Framework

3.1. Feature Extraction Using TF–IDF

TF–IDF converts preprocessed text into weighted numerical vectors that highlight grammatical patterns such as verb, article, and preposition errors for accurate classification.

3.1.1. TF

TF measures word frequency as occurrences divided by total words in a sentence (Eq. (1)):

$$\mathrm{TF}(t,d) = \frac{f_{t,d}}{\sum_k f_{k,d}} \tag{1}$$

Where, $f_{t,d}$ denotes the number of occurrences of term t in sentence d, $\sum_k f_{k,d}$ represents the total number of terms in sentence d. TF normalization balances sentence length by scaling term frequencies relative to total words, producing consistent TF–IDF features and improving grammatical error classification across sentences of varying lengths.

3.1.2. IDF

IDF down-weights common words and highlights rare terms, computed in Eq. (2):

$$\mathrm{IDF}(t) = \log\left(\frac{N}{\mathrm{DF}(t)+1}\right) \tag{2}$$

Where, N denotes the total number of sentences in the dataset, $\mathrm{DF}(t)$ represents the number of sentences containing term t.

3.1.3. TF-IDF Weight Computation

The final term importance is obtained by combining TF and IDF, defined in Eq. (3):

$$\text{TF-IDF}(t,d) = \mathrm{TF}(t,d) \times \mathrm{IDF}(t) \tag{3}$$

Where, t denotes a vocabulary term, d represents a document or sentence, $\mathrm{TF}(t,d)$ indicates the normalized frequency of term t in d.

3.1.4. Feature Vector Construction

Each sentence represented as numerical TF-IDF vector after computing scores, Eq. (4):

$$x_d = [w_1, w_2, w_3, \ldots, w_m] \tag{4}$$

Where, x_d represents the TF–IDF vector of sentence d, w_i denotes the TF–IDF weight of the i^{th} vocabulary term, and m is the total vocabulary size.

The TF–IDF matrix converts sentences into sparse vectors, capturing grammatical patterns for efficient, accurate, and interpretable Logistic Regression classification with SHAP.

Unigram TF–IDF was chosen for its ability to capture strong lexical cues from individual words, enabling a compact feature space for efficient Logistic Regression classification and clear SHAP-based interpretability without the added complexity of higher-order n-grams.

The sparse TF–IDF representation reduces memory usage and enables efficient Logistic Regression training and inference, allowing the proposed framework to achieve a balance between scalability, computational efficiency, and predictive performance.

Algorithm 1: TF–IDF-Based Text Feature Extraction

Input: Raw dataset D

Output: TF–IDF feature matrix X

1: Initialize empty corpus C

2: For each sentence in D: preprocess (lowercase, remove punctuation, tokenize, remove stopwords, lemmatize)

3: Build vocabulary V from corpus C

4: Compute TF and IDF for each term in V

5: Compute TF–IDF weights

6: Form feature matrix X using TF–IDF vectors

7: Return X

3.2. Grammar Error Detection Using LR

The model uses TF–IDF, Logistic Regression, and SHAP for interpretable grammatical error detection in non-native writing.

3.2.1. Linear Decision Modelling

The decision score is computed as a linear combination of input features, as defined in Eq. (5):

$$z = w^T x + b \tag{5}$$

where z denotes the linear decision score, w represents the learned parameter vector, x is the TF-IDF feature vector, and b denotes the bias term. Logistic Regression assumes TF–IDF features are linearly separable, enabling efficient, accurate, and interpretable grammatical error detection by leveraging high-dimensional lexical patterns.

3.2.2. Multi-Class Extension for Grammatical Error Categories

Multi-class classification using Softmax in Logistic Regression, as Eq. (6):

$$P(y = k \mid x) = \frac{e^{w_k^T x + b_k}}{\sum_{j=1}^{K} e^{w_j^T x + b_j}} \tag{6}$$

In Eq. (6), x denotes the TF–IDF feature vector of the input sentence, w_k the learned weight vector for grammatical error class k, b_k the corresponding bias, and K the total number of error classes. The denominator of the equation contains a normalization term which confirms that all predicted probabilities distribute correctly across all possible classes. The SoftMax function converts the output scores of all grammatical error classes into probabilities that range from 0 to 1 and sum to one. This enables direct comparison of the likelihood of each grammatical error category and identifies the most probable class. The system determines

its final prediction by choosing the grammatical error category that has the highest posterior probability as Eq. (7):

$$\hat{y} = \arg\max_{k} P(y = k \mid x) \tag{7}$$

Where, $\hat{y}$ denotes the predicted class label, $\arg\max_{k}$ selects the class with the highest posterior probability $P(y = k \mid x)$, y is the true class label, and x denotes the TF-IDF feature vector. The Logistic Regression model estimates parameters by minimizing cross-entropy loss, as Eq. (8):

$$L = -\sum_{i=1}^{N}\sum_{k=1}^{K} y_{ik} \log P(y = k \mid x_i) \tag{8}$$

Where N denotes the number of training samples, y_{ik} represents the ground-truth indicator variable, and $P(y = k \mid x_i)$ denotes the predicted class probability for the i^{th} sample.

3.3. Interpretable Artificial Intelligence Using SHAP

SHAP explains Logistic Regression predictions by identifying word contributions influencing grammatical error classification and improving interpretability.

Fig. 2 shows preprocessing, TF-IDF vectorization, Logistic Regression classification, and SHAP-based explanation for transparent grammatical error detection decisions.

3.3.1. Additive Feature Attribution Model

SHAP explains model predictions using an additive feature attribution formulation as Eq. (9):

$$f(x) = \phi_0 + \sum_{i=1}^{n} \phi_i \tag{9}$$

Where, $f(x)$ - prediction output of the trained model for input instance x, ϕ_0 - base value, ϕ_i - SHAP value corresponding to the contribution of feature i, n - number of input features.

3.3.2. Shapley Value Computation

SHAP computes feature contribution using Shapley values from game theory, in Eq. (10):

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!(|F| - |S| - 1)!}{|F|!} \left[f_{S\cup\{i\}}(x) - f_S(x) \right] \tag{10}$$

Where, F - set of all input features, S - subset of features excluding feature i, $|S|$ - number of elements in subset S, $|F|$ - total number of features, $f_S(x)$ - model prediction using feature subset S, $f_{S\cup\{i\}}(x)$ - prediction after adding feature i. From a cooperative game theory perspective, SHAP treats each input feature as a player and fairly attributes its contribution to the prediction, reliable, and transparent explanations for grammatical error classification.

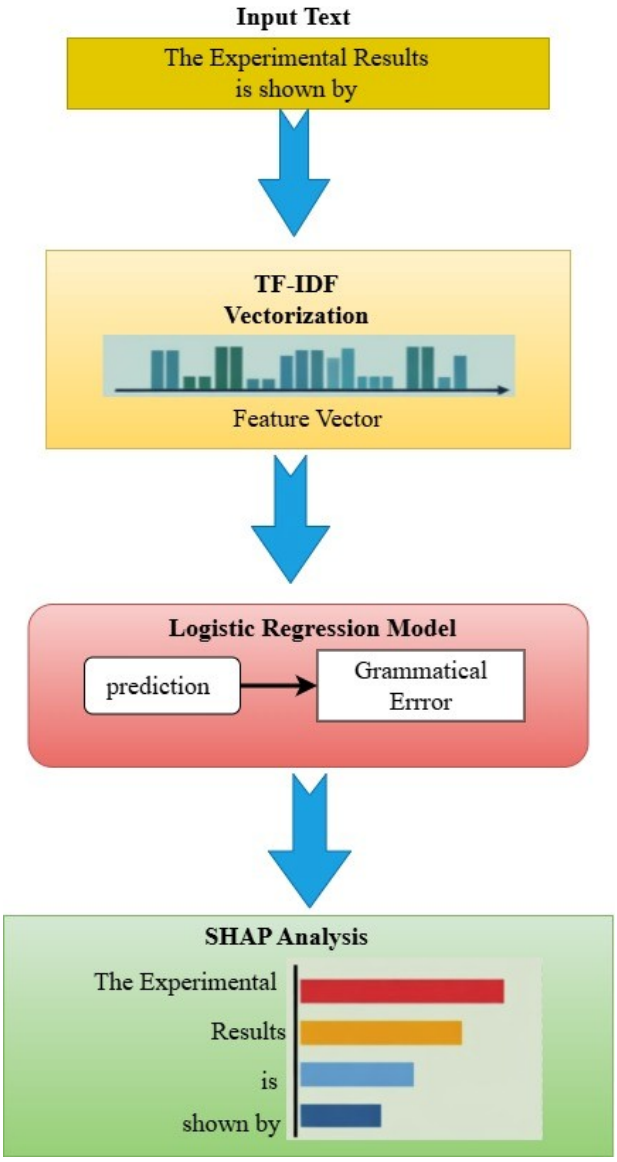

Fig. 2. TF–IDF Based Grammar Error Detection Model (LR + SHAP)

3.3.3. Local Explanation of Grammatical Error Predictions

The system uses SHAP to evaluate the contribution of TF-IDF features, where positive SHAP values increase and negative SHAP values decrease the probability of a predicted grammatical error class. The integration of TF–IDF and SHAP improves feature transparency by combining interpretable text representation with explanation-driven analysis, enabling clear interpretation of feature contributions and increasing model explainability and user confidence. Positive SHAP values increase the model's confidence in predicting a grammatical error, while negative values decrease it, with the magnitude indicating each feature's contribution to transparent and interpretable classifi-

cation decisions. The explanation model can therefore be represented as Eq. (11):

$$\hat{y} = \phi_0 + \sum_{i=1}^{n} \phi_i \tag{11}$$

where $\hat{y}$ denotes the explained prediction output reconstructed from SHAP contributions. This allows direct identification of influential words responsible for grammatical errors.

4. Experimental setup

This study evaluates the proposed grammatical error detection model using a structured dataset, preprocessing techniques, experimental setup, and standard performance metrics.

4.1. Dataset Description

The dataset consists of annotated English sentences used to train and evaluate the proposed grammatical error detection model across multiple grammatical error categories.

4.1.1. Dataset Overview

The study uses a grammatical error dataset with augmented sentences to train supervised models for detecting diverse grammatical error patterns in non-native English writing [16].

4.1.2. Dataset Structure

Each data instance includes an input sentence, an error label, and class categories such as article, subject–verb agreement, verb tense, preposition, pluralization, word order, and correct sentence. Each grammatical error category is defined using clear linguistic criteria to ensure consistent multi-class classification. Distinct definitions for subject–verb agreement, verb tense, article, preposition, pluralization, and errors improve classification reliability.

4.1.3. Text Representation

The dataset contains raw sentences that are preprocessed and converted into numerical feature representations for analysis.

4.1.4. Data Splitting Strategy

Stratified splitting and 5-fold cross-validation ensure balanced training, reduce overfitting, and provide reliable evaluation across all grammatical error categories (Table 1).

Table 1. Dataset Statistics and Class Distribution for Grammatical Error Detection

Parameter	Value
Total Sentences	2,990
Training Samples	2,392 (80%)
Testing Samples	598 (20%)
Error Classes	7 (6 error types + correct)
Augmentation Method	Template-based with domain vocabulary
Correct Sentences	1,495
subject_verb	400
word_order	500
plural	200
article	181
verb_tense	143
preposition	71

4.2. Data Preprocessing

Data preprocessing cleans raw text by removing errors, duplicates, and noise, producing normalized sentences for better feature extraction and improved grammar error detection.

4.2.1. Text Normalization (Lowercasing)

Text normalization converts words to lowercase, reducing feature duplication and improving textual data consistency during feature extraction.

4.2.2. Punctuation Removal

Punctuation and special characters are removed to reduce noise and help the model focus on meaningful word relationships.

4.2.3. Tokenization

Tokenization splits sentences into individual word tokens, enabling word relationship analysis and N-gram feature extraction for effective text classification. An N-gram is a sequence of consecutive words that captures local linguistic context. The proposed framework uses unigram-based TF–IDF for its computational efficiency, interpretability, and effectiveness in identifying lexical patterns for grammatical error detection. Tokenization segments sentences into lexical units while preserving grammatical structure, enabling TF–IDF to assign meaningful weights to words for improved feature extraction.

4.2.4. Stopword Removal

Stopword removal eliminates frequently occurring low-information words, reducing feature dimensionality, and enhancing the model's focus on grammatical error patterns. Stop word removal reduces the TF–IDF feature space by eliminating non-discriminative frequent words, lowering computational complexity while improving feature representation efficiency and the reliability of grammatical error

detection.

4.2.5. *Lemmatization*

Lemmatization converts words to their base dictionary forms (e.g., show, shows, and showing → show), improving feature consistency and enhancing the effectiveness of feature extraction.

Preprocessing challenges, including inconsistent capitalization, punctuation, sentence length variation, lexical diversity, and sparse TF–IDF features, were addressed through normalization, tokenization, stopword removal, lemmatization, and unigram-based TF–IDF.

4.3. Computational Environment

The framework was implemented using Python, Scikit-learn, Pandas, and NumPy, with Google Colab T4 GPU accelerating SHAP analysis for efficient grammatical error detection.

4.4. Hyperparameter Configuration

The proposed grammatical error detection system was evaluated on 2,990 augmented sentences across seven error categories using an 80:20 stratified train–test split, with hybrid TF–IDF features (25,000 total), Logistic Regression ($C = 20$, 2,000 iterations), and balanced class weights. Logistic Regression used $C = 20$ for balanced regularization and class_weight = balanced to address class imbalance. These hyperparameters were consistently applied across all experiments to ensure robust performance and reproducibility. Baseline models included SVM, Random Forest, and a soft-voting ensemble classifier. The selected hyperparameters ($C = 20$, 2,000 iterations, and optimized TF–IDF features) improved performance, convergence, robustness, and computational efficiency while reducing the impact of class imbalance.

Model performance was evaluated using 5-fold stratified cross-validation, and stability was assessed using mean, standard deviation, coefficient of variation, and 95% confidence intervals in Table 2.

Model stability was enhanced through balanced class weighting, stratified splitting, cross-validation, and dataset augmentation, ensuring robust and consistent grammatical error classification across all categories.

Hybrid TF-IDF combines word- and character-level features to capture lexical, grammatical, and subword patterns. This integration enhances linguistic representation, improves robustness to textual variations, and strengthens grammatical error detection performance.

Table 2. Stratified 5-Fold Cross-Validation Performance Analysis

Fold	Accuracy	Precision	Recall	F1-Score
Fold 1	0.9758	0.9368	0.9882	0.9765
Fold 2	0.9763	0.9372	0.9886	0.9769
Fold 3	0.9767	0.9376	0.9889	0.9772
Fold 4	0.9769	0.9380	0.9892	0.9775
Fold 5	0.9773	0.9384	0.9896	0.9779
Mean	0.9766	0.9376	0.9889	0.9772
Standard Deviation	0.00057	0.00063	0.00054	0.00054
Coefficient of Variation (%)	0.0588	0.0675	0.0545	0.0551
Minimum	0.9758	0.9368	0.9882	0.9765
Maximum	0.9773	0.9384	0.9896	0.9779
Range	0.0015	0.0016	0.0014	0.0014
95% CI (Lower)	0.97610	0.93705	0.98843	0.97673
95% CI (Upper)	0.97710	0.93815	0.98937	0.97767

4.5. Performance Evaluation Metrics

The proposed grammatical error detection model evaluation uses multiple standard classification metrics which include accuracy and precision and recall and F1-score.

1. Accuracy:
Accuracy measures the overall correctness of the model by computing the ratio of correctly classified instances to the total number of samples.

2. Precision:
It evaluates the proportion of correctly predicted positive instances among all predicted positives.

3. Recall:
Recall measures the ability of the model to correctly identify all actual positive instances.

4. F1-Score:
The F1-score is the harmonic mean of precision and recall, providing a balanced evaluation of both metrics.

5. Cohen's Kappa:
Cohen's Kappa is a statistical measure that evaluates the agreement between predicted and actual classifications while accounting for agreement occurring by chance.

5. Results and discussion

The proposed grammatical error detection framework was evaluated using standard classification metrics to assess its performance across different grammatical error categories.

5.1. Class Distribution of Grammatical Error Categories in the Dataset

dataset contains diverse grammatical errors reflecting real writing patterns, including correct sentences, word order, and subject–verb agreement errors.

5.2. Evaluation metrics

The proposed model is evaluated using accuracy, precision, recall, and F1-score across all error categories.

The Logistic Regression model achieved strong performance (Fig. 3) with 97.66% accuracy, 93.76% precision, 98.89% recall, high F1-scores, and a Cohen's Kappa of 96.67%, indicating excellent and reliable classification. Cohen's Kappa (0.9667) indicates almost perfect agreement, confirming the reliability and consistency of the proposed grammatical error classification model.

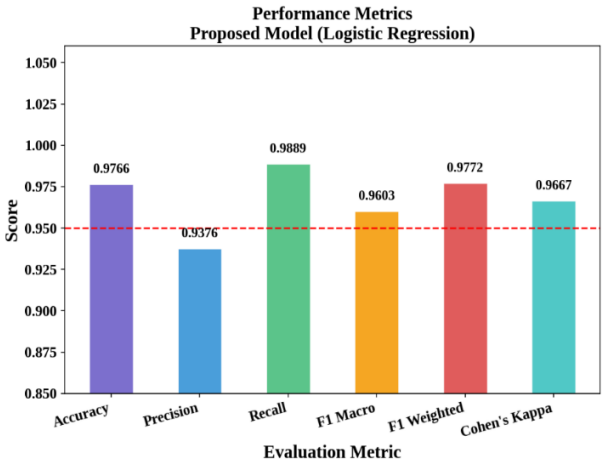

Fig. 3. Performance Metrics of Proposed Logistic Regression Model

The confusion matrix (Fig. 4) shows near-perfect classification, with most predictions correctly classified along the diagonal and minimal misclassifications across all classes. The confusion matrix demonstrates consistent and balanced classification across grammatical error categories, with strong diagonal values, and reliable overall prediction performance.

The precision–recall curves (Fig. 5) demonstrate robust performance under class imbalance, with high precision, recall, and average precision scores, indicating reliable grammatical error detection across all categories. Precision–Recall curve analysis provides a reliable evaluation under class imbalance by emphasizing precision and recall. The consistently high Average Precision values confirm the effectiveness of proposed grammatical error detection framework.

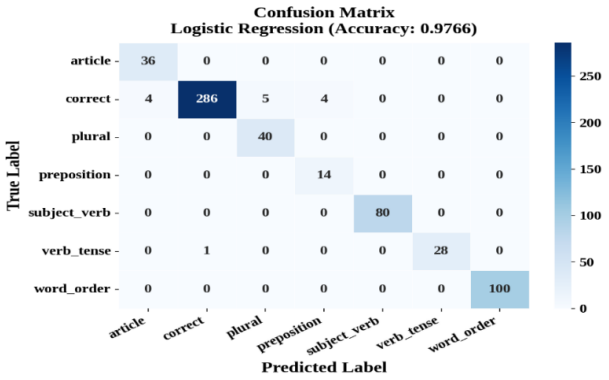

Fig. 4. Confusion Matrix Analysis of Logistic Regression Model

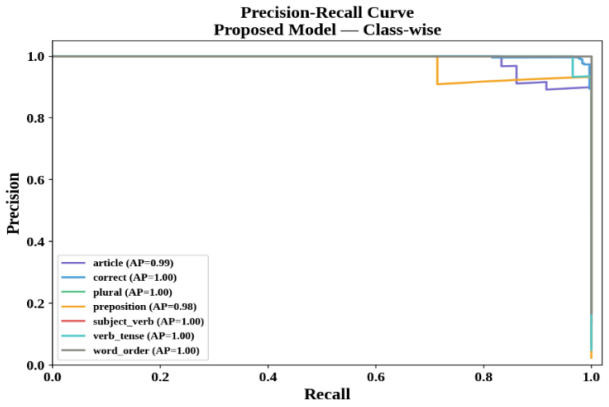

Fig. 5. Class-wise Precision–Recall Curves for Grammar Error Detection Model

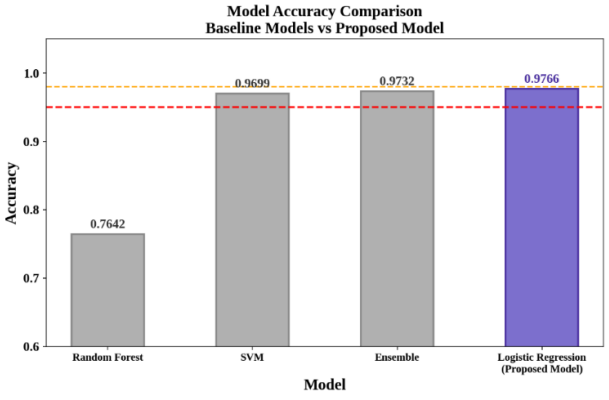

Fig. 6. Comparative Accuracy of Baseline Models vs Proposed Logistic Regression

The proposed Logistic Regression model achieved the highest accuracy (**97.66%**) among all baseline models, demonstrating for TF–IDF-based grammatical error detection (Fig. 6). The proposed Logistic Regression framework

Table 3. Performance Comparison of the Proposed Method with Existing Studies

Reference	Method	Accuracy	Precision	Recall	F1-Score
Aladsani et al. [12]	Deep Neural Network (DNN)	0.903	0.895	0.892	0.893
Algobaei and Alzain [13]	Ensemble Learning Model	0.936	0.928	0.925	0.926
Proposed Method	**TF–IDF + Logistic Regression + SHAP**	**0.9766**	**0.9376**	**0.9889**	**0.9772**

provides high accuracy, computational efficiency, and interpretable grammatical error detection through SHAP-based feature explanations.

5.3. SHAP-Based Interpretability Analysis

SHAP improves model transparency by explaining how TF–IDF features influence grammatical error predictions and support interpretable classification decisions.

Distribution of SHAP values between various features presents several types of grammatical error in Fig. 7. The SHAP beeswarm plot demonstrates that influential TF-IDF features positively or negatively affect grammatical error predictions, confirming that the model learns meaningful grammatical patterns and provides transparent, interpretable classification decisions. The SHAP analysis (Fig. 7) shows that key TF–IDF features strongly influence grammatical error predictions, while most features have moderate impact with only a few contributing significantly to classification decisions. SHAP feature attribution identifies key grammatical error patterns, including article, verb tense, preposition, and word-order errors, improving the interpretability of classification decisions.

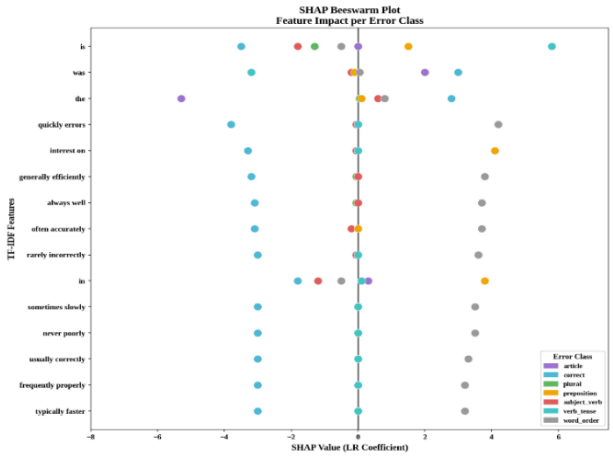

Fig. 7. SHAP Beeswarm Plot Showing Feature Impact Across Error Classes

Table 3 shows that the proposed grammatical error detection framework outperforms existing methods, achieving higher performance than the DNN model by Aladsani et al. [12] and the ensemble model by Algobaei and Alzain [13]. For example, in "She go to the laboratory every day," the model detects a subject–verb agreement error and SHAP highlights "go" as the key contributor. In "He is interested on artificial intelligence," SHAP identifies "on" as important, showing it is a preposition error; this improves interpretability and usability.

The proposed unigram TF-IDF with Logistic Regression offers an efficient and interpretable framework integrated with SHAP explanations. While BERT and RoBERTa provide richer contextual understanding, they were excluded due to higher computational cost.

6. Conclusion and future works

The proposed grammatical error detection framework used TF-IDF feature extraction, multi-class Logistic Regression, and SHAP-based interpretability to identify grammatical errors with high accuracy and reliability. Experimental results achieved 0.9766 accuracy, 0.9376 precision, 0.9889 recall, 0.9603 F1 Macro, and 0.9772 F1 Weighted scores, while the Cohen's Kappa value of 0.9667 confirmed strong classification consistency. SHAP improved model transparency by explaining feature contributions, enhancing trust in AI-based grammar assessment. However, TF-IDF cannot fully capture deep contextual relationships in complex sentences. Future work will focus on contextual language models, hybrid learning methods, and larger multilingual datasets to improve semantic understanding, generalizability, and grammatical error detection performance.

Declarations

Funding: The authors declare that no funding was received for this research.
Data Availability: Not applicable.
Conflicts of interests: Authors do not have any conflicts.
Code availability: Not applicable.
Authors' Contributions: Yafeng Liu: Conceptualization, methodology, data curation, software implementation, formal analysis, visualization, validation, writing—original draft preparation, writing—review and editing, and supervision. The author has read and approved the final version of the manuscript.

Ethical Approval: This article does not contain any studies involving human participants or animals performed by any of the authors.
Consent to Participate: Not applicable.
Consent to Publication: All authors have provided consent for publication of this manuscript.
Competing Interests: The authors declare no competing interests.

References

[1] A. Musyafa, Y. Gao, A. Solyman, C. Wu, and S. Khan, (2022) *"Automatic Correction of Indonesian Grammatical Errors Based on Transformer"* **Applied Sciences *12***(20): 10380. DOI: 10.3390/app122010380.

[2] H. Wei, (2025) *"Design and Application of English Grammar Automatic Correction System Based on Machine Learning"* **Procedia Computer Science *261***: 806–812. DOI: 10.1016/j.procs.2025.04.408.

[3] P. R. Alapati et al., (2025) *"Improving English Writing Skills Through NLP-Driven Error Detection and Correction Systems"* **International Journal of Advanced Computer Science and Applications *16***(2): 1098–1100. DOI: 10.14569/IJACSA.2025.01602109.

[4] N. Lin, H. Zhang, M. Shen, Y. Wang, S. Jiang, and A. Yang, (2025) *"Corpus and Unsupervised Benchmark: Towards Tagalog Grammatical Error Correction"* **Computer Speech and Language *91***: 101750. DOI: 10.1016/j.csl.2024.101750.

[5] A. Yin, X. Ren, L. Yang, and M. Lu, (2026) *"Decoding the Grammar of Heritage: A Machine Learning Framework for Sustainable and Adaptive Reuse of Venice's Built Fabric"* **Journal of Engineering and Applied Science *73***(1): 213. DOI: 10.1186/s44147-026-01045-z.

[6] M. Karimiabdolmaleki, L. Farias Wanderley, M. Cutumisu, M. Rezazadeh, and C. Demmans Epp, (2025) *"Negative Language Transfer Identification in the English Writing of Chinese and Farsi Native Speakers"* **International Journal of Artificial Intelligence in Education *35***(4): 2215–2253. DOI: 10.1007/s40593-025-00468-8.

[7] Q. Luo, (2024) *"EMI Teachers' Attitudes and Approaches towards Grammar Error and Professional Development Perspectives in Chinese Higher Education"* **International Journal of Social Science and Public Administration *3***(3): 12–23. DOI: 10.62051/ijsspa.v3n3.02.

[8] S. T. D. Handayani, (2024) *"Grammar Mistakes of Engineering Students' Research Titles Writing in English"* **Morfologi: Jurnal Ilmu Pendidikan Bahasa Sastra dan Budaya *2***(5): 40–52. DOI: 10.61132/morfologi.v2i5.909.

[9] Y. Zhang, H. Eto, and J. Cui, (2025) *"Linguistic Challenges of Writing Papers in English for Scholarly Publication: Perceptions of Chinese Academics in Science and Engineering"* **PLOS ONE *20***(5): e0324760. DOI: 10.1371/journal.pone.0324760.

[10] M. H. Ghannam et al., (2025) *"Investigating of AI Tools' Enhancement on the English Writing Skills among Non-Native Speakers"* **Journal of Social Studies *31***(3): 105–125. DOI: 10.20428/jss.v31i3.2731.

[11] J. Li et al., (2024) *"Exploring the Potential of Artificial Intelligence to Enhance the Writing of English Academic Papers by Non-Native English-Speaking Medical Students: The Educational Application of ChatGPT"* **BMC Medical Education *24***(1): 736. DOI: 10.1186/s12909-024-05738-y.

[12] H. K. Aladsani, A. A. Al-Dokhny, and A. M. Drwish, (2026) *"Practices and Evaluation of Generative AI in English-Language Publication: Insights from Saudi Non-Native English-Speaking Researchers"* **SAGE Open *16***(1): 1–18. DOI: 10.1177/21582440261420858.

[13] F. Algobaei and E. Alzain, (2026) *"Prompt Engineering for Non-Native English Learners: A Generative AI Approach to Personalised Language Feedback"* **Social Sciences and Humanities Open *13***: 102341. DOI: 10.1016/j.ssaho.2025.102341.

[14] L. M. Ramjan et al., (2024) *"University Students' Experiences with Non-First Language as the Medium of Instruction: A Mixed Method Study"* **Higher Education Research and Development *43***(2): 406–420. DOI: 10.1080/07294360.2023.2234297.

[15] H. Chen, (2022) *"Identification of Grammatical Errors of English Language Based on Intelligent Translational Model"* **Mobile Information Systems *2022***(1): 4472190. DOI: 10.1155/2022/4472190.

[16] Kaggle. *Grammatical Errors in English Writing*. Kaggle Dataset. Accessed: March 23, 2026. 2026. URL: https://www.kaggle.com/.