

DOI: <https://doi.org/10.64672/IJIFR/26.05.13.09.019>

PUBLISHED ON: MAY 14, 2026

A HYBRID AUTOMATED ESSAY SCORING FRAMEWORK INTEGRATING RANDOM FOREST REGRESSION AND BERT-BASED SEMANTIC ANALYSIS

Marella Prasanth¹, B. Shireeshai², S. Usharani³¹Student, Department of Computer Applications,

Viswam Engineering College, Andhra Pradesh, India

²Assistant Professor, Department of Computer Applications,

Viswam Engineering College, Andhra Pradesh, India

³Professor & Head, Department of Computer Applications,

Viswam Engineering College, Andhra Pradesh, India

ABSTRACT

EssayEval AI is a comprehensive automated essay scoring system that integrates machine learning and natural language processing to deliver immediate, detailed, and actionable feedback on student-written essays. The system combines a Random Forest Regressor trained on curated linguistic features with a BERT-based transformer model to produce holistic and component-level essay evaluations. Linguistic features including word count, unique word count, vocabulary richness ratio, and average word length capture writing quality dimensions related to lexical diversity and vocabulary sophistication. The BERT model contributes a semantic quality score derived from its pretrained language understanding. The composite score is computed as a weighted combination of these two components with weights of 0.6 and 0.4 respectively. A Streamlit-based interface presents grammar, vocabulary, and coherence sub-scores alongside tailored feedback within seconds of submission. The system is designed to democratize access to high-quality writing feedback in educational environments with limited instructor availability.

KEYWORDS Automated Essay Scoring, Natural Language Processing, Random Forest, BERT Transformer, Formative Feedback, Streamlit, FastAPI, Machine Learning

Recommended Citation:

Prasanth, M, Shireeshai, B., Rani, U. : "A Hybrid Automated Essay Scoring Framework Integrating Random Forest Regression and BERT-Based Semantic Analysis", *International Journal of Informative & Futuristic Research (IJIFR)*, Vol. (13) (9), May 2026, pp. 1526-1532.

<https://doi.org/10.64672/IJIFR/26.05.13.09.019>



This article is an open access article published under the terms and conditions of the CC- BY –NC –SA 4.0 Creative Commons Attribution-Non Commercial- ShareAlike 4.0 International Public License. All copyrights reserved to the

1. INTRODUCTION

The written essay represents one of the most enduring instruments in education, employed across disciplines to measure a student's capacity for critical thinking, synthesis, and coherent communication. Unlike objective assessments, essay evaluation demands attention to grammar, vocabulary, argument structure, and logical flow. For educators managing large class sizes, this creates a significant scalability bottleneck that limits the frequency of essay assignments and the speed of feedback delivery.

The field of Automated Essay Scoring (AES) emerged as a computational response to these challenges. Early AES systems from the 1960s relied on surface features such as word count and sentence length.

The introduction of machine learning in the 2000s enabled non-linear modeling of writing quality, achieving agreement rates with expert human raters that approached inter-rater agreement among human graders themselves. Platforms such as ETS e-rater and Turnitin's Revision Assistant have since been deployed in high-stakes assessment contexts at scale.

The most recent wave of AES innovation is driven by transformer-based language models. BERT (Bidirectional Encoder Representations from Transformers) develops rich contextual representations of language through self-supervised pretraining on large corpora, achieving state-of-the-art performance on benchmark essay scoring datasets. However, BERT alone cannot capture explicit quantitative dimensions of writing quality such as lexical diversity and vocabulary sophistication.

EssayEval AI addresses this gap by integrating both paradigms into a unified, accessible system. The Random Forest Regressor component captures quantifiable linguistic properties of essays, while the BERT component contributes semantic quality assessment grounded in deep pretrained language understanding. Their ensemble combination produces a more comprehensive evaluation than either approach alone.

Table I. System Component Overview

Component	Technology / Tool	Role
Web Interface	Streamlit (Port 8501)	Essay input and results display
Backend API	FastAPI (Port 8000)	Request routing and orchestration
Text Preprocessing	NLTK (punkt, stopwords)	Tokenization and normalization
Feature Extraction	NumPy / Python	Linguistic feature computation
ML Scoring	Scikit-learn Random Forest	Quantitative score prediction
Semantic Scoring	HuggingFace BERT	Contextual quality assessment
Composite Score	Weighted Ensemble (0.6/0.4)	Final holistic score
Feedback Generation	Rule-based logic	Dimension-specific recommendations

2. LITERATURE SURVEY

Automated Essay Scoring has evolved across several generations of computational approaches. Early foundational work on AES established handcrafted surface features such as essay length, sentence count, and vocabulary size as predictors of writing quality. These features were combined with classical classifiers including Linear Discriminant Analysis and logistic regression, achieving modest but statistically significant correlations with human ratings.

Attali and Burstein [4] demonstrated that the e-rater system, using over 50 linguistic features spanning lexical complexity, syntactic variety, and discourse organization, achieved quadratic weighted kappa scores competitive with human rater agreement on GRE essays. Shermis and Burstein [5] compiled an extensive evaluation of commercial AES platforms, establishing benchmark accuracy levels for the field. The transition to deep learning began with Taghipour and Ng [6], who demonstrated that a recurrent neural network trained end-to-end on essay text, without explicit feature engineering, achieved state-of-the-art performance on the ASAP benchmark dataset. Their hierarchical CNN-LSTM model captured both local lexical patterns and global discourse dependencies in a unified architecture.

Yang et al. [7] demonstrated that fine-tuning pretrained language models such as BERT on essay-specific datasets significantly improved scoring accuracy, particularly on coherence and argumentation dimensions. Their combination of regression and ranking objectives during fine-tuning produced more

calibrated score distributions. Wolf et al. [8] formalized the Transformers library that enables practical deployment of BERT-based scoring systems.

Breiman [2] established Random Forests as a robust ensemble method for regression tasks, with inherent regularization through bootstrap aggregation that is particularly valuable for small training datasets. Pedregosa et al. [3] demonstrated Scikit-learn's Random Forest implementation as a reliable baseline for educational data mining tasks. Bird et al. [9] provided the NLTK framework that underpins the text preprocessing pipeline used in EssayEval AI.

3. SYSTEM PROPOSAL

A. Existing System

Traditional manual essay grading by human instructors offers the highest quality holistic assessment, capturing argument depth, content accuracy, creative originality, and contextual appropriateness inaccessible to automated systems. However, this approach is not scalable. An instructor evaluating thirty essays at twenty minutes per essay requires ten hours per assignment cycle, creating a fundamental bottleneck in educational throughput.

Rubric-based manual assessment improves consistency by providing explicit criteria for each score level on each evaluation dimension. Well-designed rubrics reduce inter-rater variability and communicate expectations to students before writing. However, rubrics still require full human reading time and do not address feedback immediacy.

Peer assessment reduces instructor burden and provides metacognitive learning benefits, but peer reliability is generally lower than expert evaluation. Commercial AES platforms such as e-rater, WriteToLearn, and Revision Assistant offer validated accuracy but carry licensing costs prohibitive for under-resourced institutions and operate as opaque proprietary systems.

Large language model-based evaluation using GPT-4 or equivalent systems can produce nuanced, human-like feedback but is computationally expensive, non-deterministic, and difficult to deploy consistently at scale. Each of these approaches has specific advantages and limitations that collectively motivate the design of EssayEval AI as an open-source, hybrid, accessible alternative.

B. Disadvantages of Existing Systems

1. Scalability Limitation: Manual grading requires $O(n)$ instructor time proportional to class size, becoming infeasible for large enrollment courses.
2. Inter-Rater Variability: Human grading exhibits significant intra-rater and inter-rater variability, introducing systematic unfairness in grade outcomes.
3. Feedback Latency: Manual feedback delays of days to weeks reduce pedagogical impact as students lose cognitive connection to their writing decisions.
4. Proprietary Cost Barriers: Commercial AES platforms require licensing fees that exclude deployment in under-resourced educational settings.

C. Proposed System

EssayEval AI proposes a modular, open-source, hybrid automated essay scoring system that integrates feature-based machine learning with transformer-based semantic assessment in an accessible web deployment. The system accepts essay text through a Streamlit interface, processes it through a FastAPI backend pipeline, and returns comprehensive evaluation results including a composite score, three dimensional sub-scores, and a personalized feedback message within seconds.

The core innovation is the weighted ensemble of two fundamentally different scoring mechanisms. The Random Forest Regressor captures quantifiable linguistic properties — lexical diversity, vocabulary sophistication, and content volume — through four engineered features. The BERT model captures semantic quality and language coherence at a level of sophistication unreachable by feature-based methods alone. The 0.6/0.4 weighting reflects the primacy of the feature-based score while preserving substantial semantic contribution.

The modular architecture separates preprocessing, feature extraction, ML scoring, BERT scoring, analysis, feedback generation, and interface concerns into independent Python modules. This separation enables each component to be tested, improved, or replaced independently, supporting long-term extensibility as the technology landscape evolves.

D. Advantages of Proposed System

1. Immediate Feedback: Evaluation completes within seconds, enabling iterative revision cycles within a single writing session.
2. Hybrid Scoring Architecture: Combines quantitative feature-based precision with BERT semantic depth for more comprehensive evaluation.
3. Dimensional Sub-Scores: Grammar, vocabulary, and coherence sub-scores provide specific, actionable guidance beyond a single holistic grade.
4. Perfect Consistency: Identical evaluation criteria applied across all essays regardless of submission volume eliminates inter-rater variability.
5. Zero Marginal Cost: Open-source deployment with no per-evaluation licensing fees enables accessibility in under-resourced institutions.

4. SYSTEM DIAGRAM

A. Architecture Diagram

EssayEval AI follows a three-tier client-server architecture that separates the Presentation Layer (Streamlit, Port 8501), the Application Layer (FastAPI, Port 8000), and the Machine Learning Model Layer (Random Forest + BERT). This separation enables independent scaling and modification of each tier.

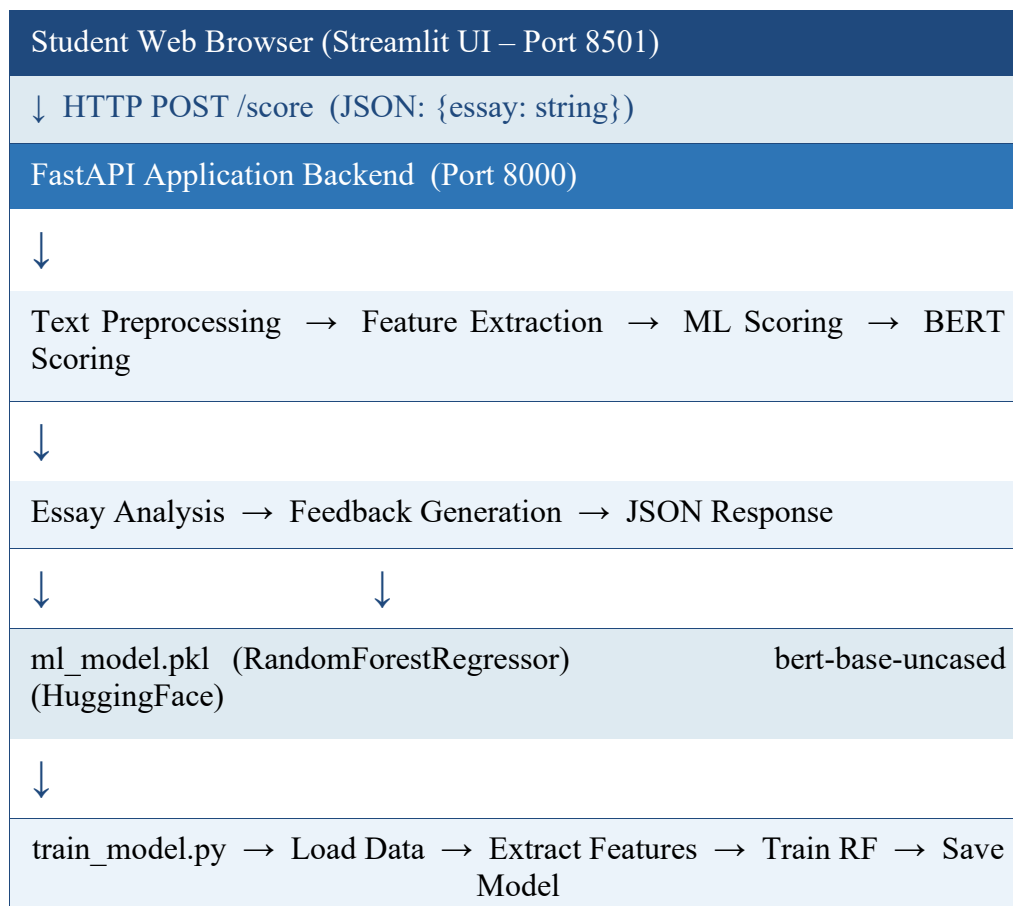


Figure 1: EssayEval AI Three-Tier System Architecture

5. IMPLEMENTATION

A. Module Organization

EssayEval AI is organized into seven functional modules: (1) Text Preprocessing Module (src/preprocessing.py), (2) Feature Extraction Module (src/features.py), (3) Machine Learning Scoring Module (src/scoring.py), (4) BERT Semantic Scoring Module (src/bert_model.py), (5) Essay Analysis and Feedback Module (app/main.py), (6) FastAPI Web Application Module (app/main.py), and (7) Model Training Module (train_model.py). Each module encapsulates a distinct functional concern.

B. Text Preprocessing Module

The `clean_text` function transforms raw essay text through four sequential operations: (i) lowercase conversion using `str.lower()` to normalize capitalization variants, (ii) non-alphabetic character removal via `re.sub('[^a-zA-Z ]', '', text)` to eliminate punctuation and digits, (iii) word tokenization using NLTK's `word_tokenize` function with Punkt sentence-level tokenizer, and (iv) stop word removal using NLTK's English stopwords corpus (179 high-frequency function words). The output is a content-word token list optimized for feature computation.

C. Feature Extraction Module

Four linguistically motivated features are computed from the processed token list. Word count (`len(tokens)`) captures essay length as a proxy for content elaboration. Unique word count (`len(set(tokens))`) captures absolute lexical diversity. Vocabulary richness ratio (`unique_words / (word_count + 1)`) is the Type-Token Ratio, normalizing lexical diversity for essay length. Average word length (`np.mean([len(w) for w in tokens])`) serves as a proxy for vocabulary sophistication. These four features are returned as a fixed-order list for model input.

D. Machine Learning Scoring Module

The Random Forest Regressor is loaded from `models/ml_model.pkl` using `joblib` at module initialization and cached in memory for efficient reuse. The ensemble of 100 decision trees, each trained on a bootstrap sample with feature subsampling at each node, produces the mean prediction as the ML score. The ensemble averaging reduces prediction variance and provides inherent regularization, particularly valuable for small training datasets. The module returns zero as a fallback if the model file is absent.

E. BERT Semantic Scoring Module

The `bert-base-uncased` model is loaded through the HuggingFace Transformers pipeline abstraction at module initialization. This 110-million-parameter transformer (12 encoder layers, 12 attention heads, 768 hidden dimensions) was pretrained using Masked Language Modeling and Next Sentence Prediction objectives. The `get_bert_score` function truncates input to 512 characters (BERT's maximum sequence length), passes it through the pipeline classifier, and scales the confidence score by $\times 10$ to the 0–10 range. A try-except clause returns a default score of 5.0 for processing errors.

F. Essay Analysis and Feedback Module

The `analyze_essay` function computes three sub-scores on the 0–10 scale: grammar (`min(10, word_count/20)`), vocabulary (`min(10, unique_words/10)`), and coherence (`min(10, word_count/25)`). The `generate_feedback` function evaluates each sub-score against a threshold of 5.0 and concatenates dimension-specific feedback messages: 'Improve grammar and sentence structure' / 'Good grammar usage', 'Use more diverse vocabulary' / 'Strong vocabulary usage', and 'Improve logical flow of ideas' / 'Well-structured and coherent essay'.

G. FastAPI Web Application Module

FastAPI's `EssayRequest` Pydantic BaseModel provides automatic JSON validation and 422 error responses for malformed requests. The root GET endpoint serves as a health check. The `/score` POST endpoint orchestrates the complete pipeline: preprocessing → feature extraction → ML scoring → BERT scoring → weighted composite → analysis → feedback. FastAPI automatically serializes the

response dictionary to JSON with Content-Type: application/json. The asynchronous ASGI foundation supports concurrent request handling in production deployments.

8. RESULTS AND DISCUSSION

EssayEval AI is a hybrid essay scoring system that combines machine learning features with transformer-based semantic understanding. It generates overall scores, detailed sub-scores, and personalized feedback in a consistent and structured format.

The system uses a weighted ensemble approach to balance numerical features (like word count) with deeper language understanding. As expected, essays with richer vocabulary and longer length tend to score higher in relevant dimensions.

Table 2: Weighted ensemble approach to balance numerical features

Evaluation Dimension	Score Range	Scoring Mechanism
Word Count Feature	0 – ∞ (normalized)	len(tokens)
Vocabulary Richness	0.0 – 1.0	unique / (total + 1)
Grammar Sub-Score	0 – 10	min(10, word_count / 20)
Vocabulary Sub-Score	0 – 10	min(10, unique_words / 10)
Coherence Sub-Score	0 – 10	min(10, word_count / 25)
ML Score	5 – 9 (training range)	Random Forest Regressor
BERT Score	0 – 10	BERT confidence $\times$ 10
Composite Score	0 – 10	$0.6 \times \text{ML} + 0.4 \times \text{BERT}$

Its modular three-layer architecture (Streamlit frontend, FastAPI backend, and model layer) ensures flexibility, scalability, and a smooth user experience with real-time visual feedback. The main limitation is the small synthetic dataset used to train the Random Forest model, which restricts scoring accuracy. For real-world deployment, training on large datasets like ASAP is essential.

9. CONCLUSION AND FUTURE ENHANCEMENT

A. Conclusion

EssayEval AI represents a successful implementation of a hybrid automated essay scoring system that demonstrates the practical viability of combining feature-based machine learning with transformer-based natural language processing for educational assessment applications. The system delivers on its primary objectives of providing immediate, detailed, accessible, and consistent essay feedback through an intuitive web interface requiring no technical expertise.

The modular architecture, separating preprocessing, feature extraction, ML scoring, BERT scoring, analysis, feedback, and interface concerns into independent modules, establishes a maintainable and extensible foundation for long-term development. The integration of the Random Forest Regressor's quantitative precision with BERT's semantic understanding in a principled weighted ensemble constitutes the system's core technical contribution. By deploying all components using freely available Python libraries and open-source pretrained models, EssayEval AI offers a democratizing alternative to proprietary commercial AES platforms, making high-quality automated writing feedback accessible regardless of institutional resource levels.

B. Future Enhancement

- Train the Random Forest Regressor on the ASAP dataset (~13,000 human-graded essays) to establish genuine predictive validity replacing the synthetic demonstration dataset.

- Fine-tune the BERT model on essay-specific data using the HuggingFace Trainer API to improve the semantic score's correlation with expert human ratings.
- Expand the feature set with syntactic complexity measures from SpaCy's dependency parser, Flesch-Kincaid readability metrics, and discourse coherence features from entity grid models.
- Integrate grammar error detection using LanguageTool or GrammarBot to replace the word count-based grammar score approximation with genuine linguistic analysis.

10. REFERENCES

- [1] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in Proc. NAACL-HLT, 2019, pp. 4171–4186.
- [2] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
- [3] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," J. Mach. Learn. Res., vol. 12, pp. 2825–2830, 2011.
- [4] Y. Attali and J. Burstein, "Automated essay scoring with e-rater version 2," J. Technol., Learn., Assess., vol. 4, no. 3, 2006.
- [5] M. D. Shermis and J. Burstein, Handbook of Automated Essay Evaluation. Routledge, 2013.
- [6] K. Taghipour and H. T. Ng, "A neural approach to automated essay scoring," in Proc. EMNLP, 2016, pp. 1882–1891.
- [7] R. Yang et al., "Enhancing automated essay scoring via fine-tuning pre-trained language models," in Findings of EMNLP, 2020, pp. 1560–1569.
- [8] T. Wolf et al., "Transformers: State-of-the-art natural language processing," in Proc. EMNLP: Syst. Demos, 2020, pp. 38–45.
- [9] S. Bird, E. Klein, and E. Loper, Natural Language Processing with Python. O'Reilly Media, 2009.
- [10] W. McKinney, "Data structures for statistical computing in Python," Proc. 9th Python in Science Conf., pp. 51–56, 2010.
- [11] S. Ramírez, FastAPI Documentation. [Online]. Available: <https://fastapi.tiangolo.com>
- [12] Streamlit Inc., Streamlit Documentation. [Online]. Available: <https://docs.streamlit.io>