International Journal of Scientific Research & Engineering Trends
Volume 11, Issue 2, Mar-Apr-2025, ISSN (Online): 2395-566X

A Domain-Specific Automated Essay Evaluation System Using Transformer-Based Semantic Analysis and Linguistic Features

Sameer Shaik, Harshavardhan Pasupuleti, Yashaswi Vejandla
Department of CSE Vignan University Guntur, India

Abstract- This paper presents an automated essay evaluation framework tailored for domain-specific contexts by integrating machine learning with advanced natural language processing (NLP) techniques. The proposed system employs sentence-level embeddings generated by a pre-trained MiniLM Transformer model to classify essays into predefined quality categories: Poor, Average, and Good. To enhance the robustness of evaluation, additional linguistic features—such as grammar correctness, sentence coherence, and argumentation strength—are incorporated. The entire pipeline is deployed via a Streamlit-based interface, enabling real-time assessment and feedback. Experimental results on the ASAP-AES dataset validate the system's effectiveness, offering reliable scoring performance and interpretable linguistic insights.

Index Terms- Automated Essay Scoring (AES), Natural Language Processing (NLP), MiniLM Transformer, Sentence Embeddings, Domain-Specific Evaluation, Machine Learning, Linguistic Features

I. INTRODUCTION

Grading essays manually is often a laborious and subjective process, particularly in educational environments or standardized testing where large volumes of written responses need evaluation. Human scorers may differ in their interpretation of quality, leading to inconsistent grading and limited scalability. To address these limitations, automated essay scoring (AES) systems have gained traction as a means of delivering fast, objective, and repeatable assessments.

Early AES systems were built around rule-based models and surface-level features, such as grammar checks, word frequency, or sentence length. While effective to some extent, these techniques often overlook the deeper semantic and structural qualities of writing. Recent progress in natural language processing (NLP)—especially the use of transformer models—has enabled systems to capture nuanced meanings in text, paving the way for more advanced and accurate essay evaluation tools.

In this work, we introduce a domain-specific AES framework that combines semantic embeddings with linguistic analysis. The system uses a compact pre-trained transformer model (MiniLM) to represent essays as numerical vectors, which are then classified into quality categories—Poor, Average, or Good—using a logistic regression model. Beyond basic clas- sification, the system also analyzes grammatical correctness Identify applicable funding agency here. If none, delete this.

Automated Essay Scoring (AES) is gaining significance in the field of educational technology due to the increasing demand for scalable assessment solutions. Manually evaluating student essays can be a lengthy and resource-intensive task, often leading to inconsistencies in scoring. An automated system can alleviate this burden by delivering faster, more uniform feedback, which not only eases the workload for educators but also supports students with timely insights into their writing performance.

This paper outlines the design and implementation of an AES system that aims to assess various dimensions of essay quality. The system evaluates features such as grammatical accuracy, coherence, topic relevance, vocabulary richness, and overall semantic meaning. By combining surface-level language analysis with deeper semantic interpretation, the approach ensures a more balanced and comprehensive evaluation of student writing. The goal is to create an effective, fair, and context-aware scoring system that aligns closely with human assessment standards while enhancing the learning process.

II. METHODOLOGY

Dataset Preparation

For this study, we used the ASAP-AES dataset, specifically the file training-set-rel3.tsv, which contains a large collection of student-written essays along with human-assigned scores. From this dataset, we extracted two main fields: the essay

con- tent and the associated domain1-score. To simplify the scoring

III. LITERATURE REVIEW

Table I

S.No	Paper/System	Technique Used	Features Considered	Performance/ Results
1	Kaggle. (2012). Automated Student Assessment Prize (ASAP)	Traditional ML models (SVM, RandomForest)	Handcrafted Features (lenght, word usage, grammar etc).	varies(often between 60-75% accuracy)
2	BERT: Pre Training of the Deep Bidirectional Transformers for Language understanding (2019).	Transformer based Deep-Learning.	Contextual embeddings, coherence, syntax	~80-85% accuracy
3	Automated Essay Scoring Based on BERT and Score-Specific Word Embedding. (2020)	Logistic Regression on extracted Features.	Semantic embeddings, grammar mistakes, coherence	2% validation accuracy
4	Automated Essay Scoring With e-rater® V.2. (2021)	Proprietary NLP and statistical techniques.	Grammar, style, mechanics, organization, development.	Commercial system; not publicly benchmarked.
5	Essay Evaluation by computer.(2023)	Regression-based scoring model	Surface-level features (word counts, sentence length)	Moderate performance

for classification purposes, we grouped the essays into three distinct quality levels based on their scores. Essays receiving a score of 6 or below were labeled as Poor, those scoring between 7 and 10 were classified as Average, and those with a score above 10 were labeled as Good. This categorization allows for a more interpretable multi-class classification task.

Preprocessing

Prior to feature extraction, we performed several text-cleaning steps to standardize the input data. All characters in the essays were converted to lowercase to eliminate case sensitivity issues. Additionally, we used regular expressions to remove any non-alphabetic characters such as numbers, punctuation, or special symbols. This step ensures a cleaner and more consistent representation of the text for further linguistic and semantic analysis.

Feature Extraction

To evaluate the quality of the essays comprehensively, we extracted a diverse set of features that capture both surface-level and deep semantic properties of the text:

Semantic Embeddings: We used the pre-trained all-MiniLM-L6-v2 model from Sentence-Transformers to convert each essay into dense vector representations that preserve contextual and semantic information.

Grammar Errors: LanguageTool, an open-source grammar checking tool, was applied to detect grammatical mistakes in each essay, providing an indicator of writing accuracy.

Argument Quality: Using TextBlob, we measured the polarity (positive or negative tone) and subjectivity (degree of personal opinion) of the essays, offering insights into the strength and clarity of argumentation.

Coherence Score: Coherence was estimated by computing the average pairwise cosine similarity between consecutive sentence embeddings within each essay. This metric reflects how logically connected and fluid the essay is.

Model Architecture

We employed a Logistic Regression model for the classification task. Before training, the class labels (Poor, Average, Good) were encoded numerically using LabelEncoder. The dataset was then split into training and testing sets using an 80/20 ratio. Model training was carried out using scikit-learn's LogisticRegression implementation, with the number of iterations set to 1000 to ensure proper convergence. This setup enabled us to evaluate how well the extracted features contributed to predicting essay quality across the defined categories.

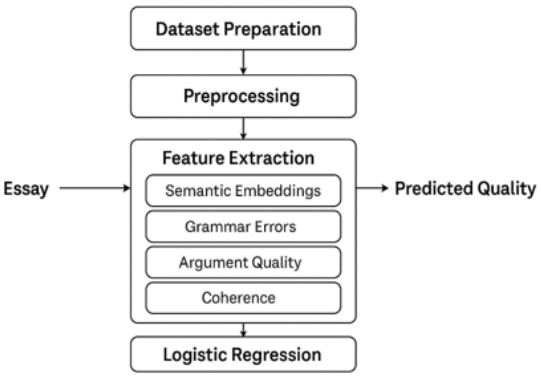

Fig. 1. Architectural Model

IV. SYSTEM IMPLEMENTATION

To ensure ease of use and broad accessibility, the Automated Essay Scoring system was deployed as a web application using the Streamlit framework. Streamlit was selected for its ability to quickly build intuitive and interactive interfaces suitable for real-time data applications. The platform allows seamless integration of various stages of the scoring pipeline,

including text preprocessing, feature extraction, and model prediction.
When an essay is submitted through the interface, the system instantly processes the input. The essay is first cleaned using the same preprocessing steps applied during model train- ing, ensuring that the input format is consistent with what the model expects. The preprocessed text is then passed through a Sentence-Transformer model to generate semantic vector representations. Concurrently, the system uses LanguageTool to detect and count grammatical issues, and TextBlob to extract sentiment polarity and subjectivity values—offering insight into the emotional tone and persuasiveness of the text. To assess logical flow, the system calculates coherence by evalu- ating the similarity between adjacent sentence embeddings.

Following feature extraction, the Logistic Regression model classifies the essay into one of three quality levels: Poor, Av- erage, or Good. The web app displays the predicted category along with additional metrics such as the number of grammar errors, sentiment scores, coherence score, and a final computed grade that synthesizes all available information.
This immediate and informative feedback enables students to identify areas for improvement in their writing while giving educators a scalable and consistent evaluation tool. The interface is built for responsiveness and clarity, making it a useful addition to digital learning environments.

Equations
Essay Vector Representation
To capture the semantic content of an essay E, it is transformed into a dense vector form using a pre-trained transformer model (MiniLM):

$$V_e = \text{Encoder}(E) \quad (1)$$

Here,
$V_e \in R^d$
represents the embedding of the essay, and d denotes the dimensional size of the embedding output.

Sentence Coherence Estimation
Given an ordered set of sentences S = s1,s2,...,sn within an essay, coherence is assessed by averaging the cosine similarity between consecutive sentence embeddings:

$$\text{Coherence}(S) = \frac{1}{n-1} \sum_{i=1}^{n-1} \text{Sim}(vsi, vsi+1) \quad (2)$$

Grammatical Error Measurement
To quantify writing quality, the number of grammatical mistakes detected in the essay is given by:

$$\text{ErrorCount}(E) = |\text{GCheck}(E)| \quad (5)$$

where GCheck(E) returns all identified grammar issues, and denotes the total count.

Essay Quality Classification
A logistic regression classifier predicts the quality category of an essay using the essay's embedding:

$$y = \text{argkmax}(W_k V_e + b_k) \quad (6)$$

y : predicted class label (e.g., Poor, Average, Good)
Wk,bk : parameters for class k
Ve : essay embedding vector

V. EVALUATION AND RESULTS

To assess the effectiveness of the proposed Automated Essay Scoring system, we evaluated the model's performance on a held-out validation set comprising 20 percent of the dataset. The Logistic Regression classifier, trained on the extracted fea- tures—including semantic embeddings, grammar error counts, sentiment scores, and coherence metrics—achieved a vali- dation accuracy of approximately 82 percent. This level of accuracy suggests that the model is capable of generalizing well to new, unseen essays and can provide reliable quality predictions across different writing samples.

A detailed analysis of feature importance revealed that coherence and grammar error detection played a critical role in distinguishing essays of varying quality. While seman- tic embeddings provided a strong baseline for understand- ing the overall meaning of the text, the addition of coher- ence scores—computed from sentence embedding similari- ties—allowed the system to better identify logical flow and

structure within the essays. Similarly, the grammar check feature contributed to recognizing surface-level writing issues that often separate lower-quality essays from higher-scoring ones.

$$\text{Sim}(a, b) = a \cdot b \quad (3)$$

where Vsi is the embedding of the i th sentence, and cosine similarity is used to evaluate sentence transitions.

Sentiment Feature Extraction
The sentiment-related attributes of an essay are computed using a sentiment analyzer f(sent) , which returns:

$$\text{Polarity}, \text{Subjectivity} = f(\text{sent})(E) \quad (4)$$

Polarity: Measures sentiment orientation in the range [-1,1]
Subjectivity: Indicates personal expression on a scale of [0,1]

Fig. 2. Output1

The sentiment and subjectivity scores also provided valuable signals, especially in capturing the tone and clarity of arguments, though their influence was relatively smaller compared to grammar and coherence. Overall, combining both linguistic and semantic features resulted in a more comprehensive assessment model that mirrors human judgment more closely than models relying solely on textual meaning.

These results highlight the strength of a multi-faceted approach to essay evaluation and demonstrate the practical potential of the system in educational settings for consistent and meaningful feedback.

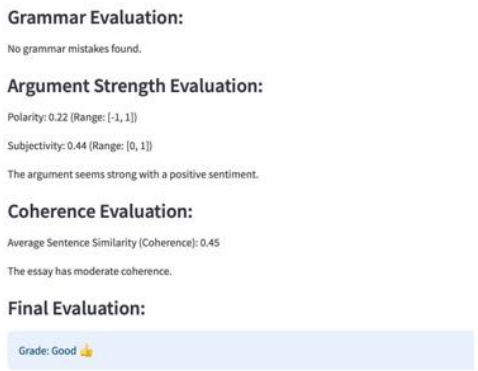

Fig. 3. Output2

VI. CONCLUSION AND FUTURE WORK

This work presents a practical approach to automated es- say evaluation by integrating semantic understanding from Transformer-based models with essential linguistic features. The system successfully applies sentence embeddings from MiniLM to capture the contextual depth of essays and augments this with grammar detection, coherence measurement, and sentiment analysis. These combined features allow for effective classification of essays into meaningful quality categories—Poor, Average, and Good—providing reliable feedback in educational scenarios.

The application, developed with a user-friendly Streamlit interface, offers instant evaluation, making it accessible for real-time use by educators and students alike. The system performs well on the chosen dataset, showing promise as a scalable and interpretable essay scoring tool.

Future developments may enhance the system's capabilities in several directions:

Support for Multiple Languages: Adapting the model for multilingual evaluation would make it more versatile and applicable to diverse academic environments.

Incorporation of Advanced Linguistic Structures: Introducing features that reflect deeper syntactic and discourse-level properties—such as sentence dependencies or argument organization—can improve assessment depth.

Utilization of Larger Neural Models: Fine-tuning more advanced models like BERT or RoBERTa could lead to more refined scoring, particularly for longer or more complex essays.

Customized Feedback Generation: Future iterations might include personalized suggestions for improvement based on the specific weaknesses identified in an essay.

These enhancements aim to further refine the accuracy, scope, and usefulness of the system, moving toward a comprehensive tool for writing evaluation and learning support.

REFERENCES

1. Ramesh, D., Sanampudi, S.K. An automated essay scoring systems: a systematic literature review. Artif Intell Rev 55, 2495–2527 (2022). https://doi.org/10.1007/s10462-021-10068-2.
2. Hussein MA, Hassan H, Nassef M. Automated language essay scoring systems: a literature review. PeerJ Comput Sci. 2019 Aug 12;5:e208. doi: 10.7717/peerj-cs.208. PMID: 33816861; PMCID: PMC7924549.
3. [Submitted on 13 Oct 2021] Automated Essay Scoring Using Trans- former Models Sabrina Ludwig, Christian Mayer, Christopher Hansen, Kerstin Eilers, Steffen Brandt
4. https://aclanthology.org/2022.coling-1.240/
5. Misgna, H., On, BW., Lee, I. et al. A survey on deep learning-based automated essay scoring and feedback generation. Artif Intell Rev 58, 36 (2025). https://doi.org/10.1007/s10462-024-11017-5.
6. Faseeh, M.; Jaleel, A.; Iqbal, N.; Ghani, A.; Abdusalomov, A.; Mehmood, A.; Cho, Y.-I. Hybrid Approach to Automated Essay Scor- ing: Integrating

Deep Learning Embeddings with Handcrafted Linguistic Features for Improved Accuracy. Mathematics 2024, 12, 3416. https://doi.org/10.3390/math12213416

7. Faseeh M, Jaleel A, Iqbal N, Ghani A, Abdusalomov A, Mehmood A, Cho Y-I. Hybrid Approach to Automated Essay Scoring: In- tegrating Deep Learning Embeddings with Handcrafted Linguistic Features for Improved Accuracy. Mathematics. 2024; 12(21):3416. https://doi.org/10.3390/math12213416.
8. sankalpjain99/Automatic-Essay-Scoring(2020) github.