AI based tool that instantly grades students' essays, provides constructive feedback for improvement, and detects plagiarism, helping teachers save time, reduce subjective bias in scoring, and promote fair and personalized learning.

Potnuru Jayant , Gavini Karthik , Kopalli Vijay Kumar, Yeluri Badri Nandan

23BEC7042, SENSE, VIT-AP UNIVERSITY
23BCE7905, SCOPE, VIT-AP UNIVERSITY
23MIC7015, SCOPE, VIT-AP UNIVERSITY
23BCE8900, SCOPE, VIT-AP UNIVERSITY

ABSTRACT: With the increasing role of technology in education, automatic essay grading systems have become essential for efficient and unbiased evaluations. This project aims to develop an advanced tool that instantly assesses essays, provides detailed feedback, and detects plagiarism. Using machine learning models, the system evaluates key writing aspects such as grammar, coherence, structure, and relevance, helping students improve their writing skills while reducing the workload on educators.To further enhance academic integrity, the integrated plagiarism detection system allows teachers to check entire folders in a single click, eliminating the need for manual submissions. This ensures a seamless and efficient way to identify similarities with existing content while maintaining fairness in assessments.By automating the grading and feedback process, this system minimizes human bias, enhances individualized learning, and makes assessments more transparent. Its adoption can significantly improve the efficiency of the education system, empowering both students and educators with real-time insights and a more interactive learning experience.

I. INTRODUCTION

The increasing reliance on technology in education has led to the development of automated tools that enhance learning and assessment. One such advancement is automatic essay grading, which significantly reduces the workload of educators while ensuring fairness and consistency in evaluation. Traditional essay grading is often time-consuming and subject to individual biases, making it difficult to provide timely and personalized feedback to students.This research presents an AI-powered essay evaluation tool that not only assigns scores based on predefined criteria but also provides constructive feedback and detects plagiarism. The system assesses key writing attributes—content, organization, and language quality—to generate meaningful insights that help students refine their writing skills. Our approach leverages the T5-small transformer model, a powerful NLP-based framework trained for text generation tasks, which has been fine-tuned on the DREsS dataset to enhance its performance in automated essay scoring.A key feature of this tool is its integrated plagiarism detection system, which allows teachers to efficiently check entire folders in a single click, eliminating the need for manual submissions. This ensures academic integrity while simplifying

the evaluation process. By automating grading, providing instant feedback, and maintaining transparency, this system transforms essay evaluation into a faster, more objective, and interactive process, ultimately enhancing the learning experience for students and streamlining assessment for educators.

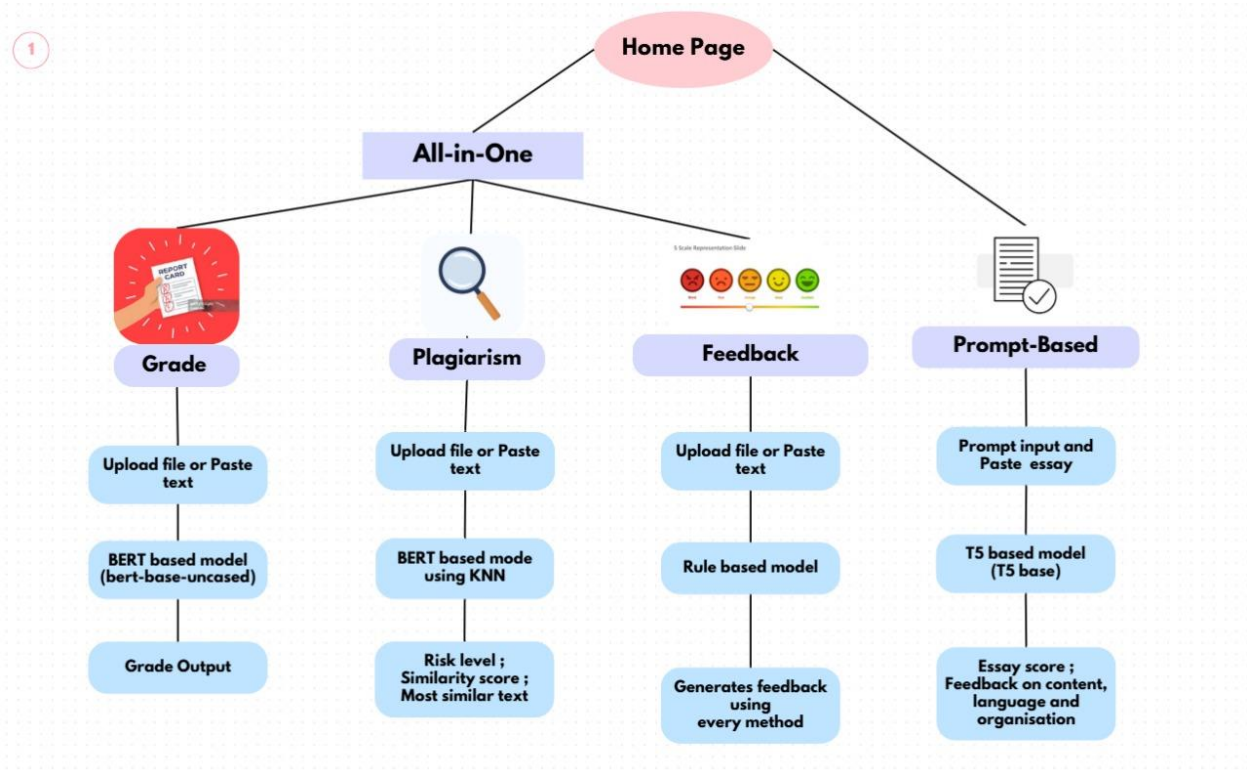
II. LITERATURE REVIEW

Research Papers on Automated Essay Scoring and Plagiarism Detection

Title	Publication	Year	Technology Used	Introduction
Automated Essay Scoring Using Machine Learning	IEEE	2024	Machine Learning (ML), NLP	This paper explores the use of ML techniques for scoring essays automatically, improving grading efficiency and reducing human bias.
Unveiling the Tapestry of Automated Essay Scoring	arXiv	2024	Support Vector Machines (SVM), Random Forests, Deep Learning (CNN, LSTM), Pretrained Language Models (BERT)	This study investigates the trade-offs between different AES models, comparing prompt-specific and cross-prompt approaches while highlighting fairness issues in essay scoring.
Review of Feedback in Automated Essay Scoring	arXiv	2023	AI, Machine Learning, Feedback Mechanisms	This paper reviews research on feedback in AES, discussing its importance in improving students' writing skills. It also examines case studies of AES systems that provide feedback.
An Automated Essay Scoring System: A Systematic Literature Review	Springer	2021	AI-based scoring, Feature Engineering	This paper systematically reviews automated essay scoring systems, highlighting key advancements and challenges.
Exact String Matching Algorithms: Issues, and Future Research Directions	IEEE	2019	String matching, Boyer-Moore, Rabin-Karp, Exact string matching, Pattern recognition	The work focuses on software-based pattern string matching algorithms and their applications.
An Improved Plagiarism Detection Scheme Based on Semantic Role Labeling	Elsevier	2012	Character-based methods, Cluster-based methods, Syntax-based methods, Cross-language-based methods	This paper introduces a plagiarism detection technique based on Semantic Role Labeling (SRL).

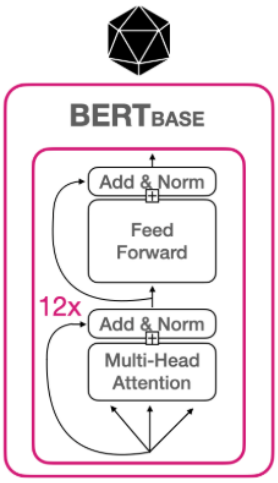
A New Approach for Cross-Language Plagiarism Analysis	Springer	2010	J48 Classification Algorithm	This paper presents a novel method for detecting plagiarism in multiple languages using classification techniques.
Attention-based Recurrent Convolutional Neural Network for Automated Essay Scoring	ACL Anthology	2017	Recurrent Neural Networks (RNN), Convolutional Neural Networks (CNN), Attention Mechanism	This paper explores the use of a hierarchical sentence-document model with attention mechanisms to improve essay scoring accuracy.
A Neural Approach to Automated Essay Scoring	ACL Anthology	2016	Recurrent Neural Networks (RNN), LSTM	This paper presents a recurrent neural network model for essay scoring, improving the fairness and accuracy of automated grading.

III. METHODOLOGY



In the given system architecture, users can upload or paste text for evaluation. The system processes essays using a **BERT-based model** for **grading** and **plagiarism detection**, while a **rule-based model** provides **detailed feedback** on writing quality. Additionally, a **T5-based model** is used for **prompt-based evaluation**, assessing content relevance and coherence. The system eliminates duplicate comparisons in plagiarism detection and generates a comprehensive report, including **essay scores, constructive feedback, plagiarism percentage, similarity analysis, and remarks**, ensuring an efficient, fair, and personalized assessment.

Grade:



BERT is powered by a powerful neural network architecture known as Transformers. This architecture incorporates a mechanism called self-attention, allowing BERT to weigh the significance of each word based on its context, both preceding and succeeding. This context-awareness imbues BERT with the ability to generate contextualized word embeddings, which are representations of words considering their meanings within sentences. It's akin to BERT reading and re-reading the sentence to gain a deep understanding of every word's role.

The grading process begins when a user uploads a file or pastes text into the system. The text undergoes tokenization and vectorization, converting it into numerical representations for processing. A BERT-based model then analyzes key aspects such as coherence, structure, grammar, and readability. By leveraging these embeddings, the model evaluates the overall quality of the essay and assigns a grade based on predefined criteria. This ensures an objective and standardized assessment. Additionally, the system provides detailed feedback, highlighting strengths and areas for improvement, enabling users to refine their writing effectively.

Plagiarism:

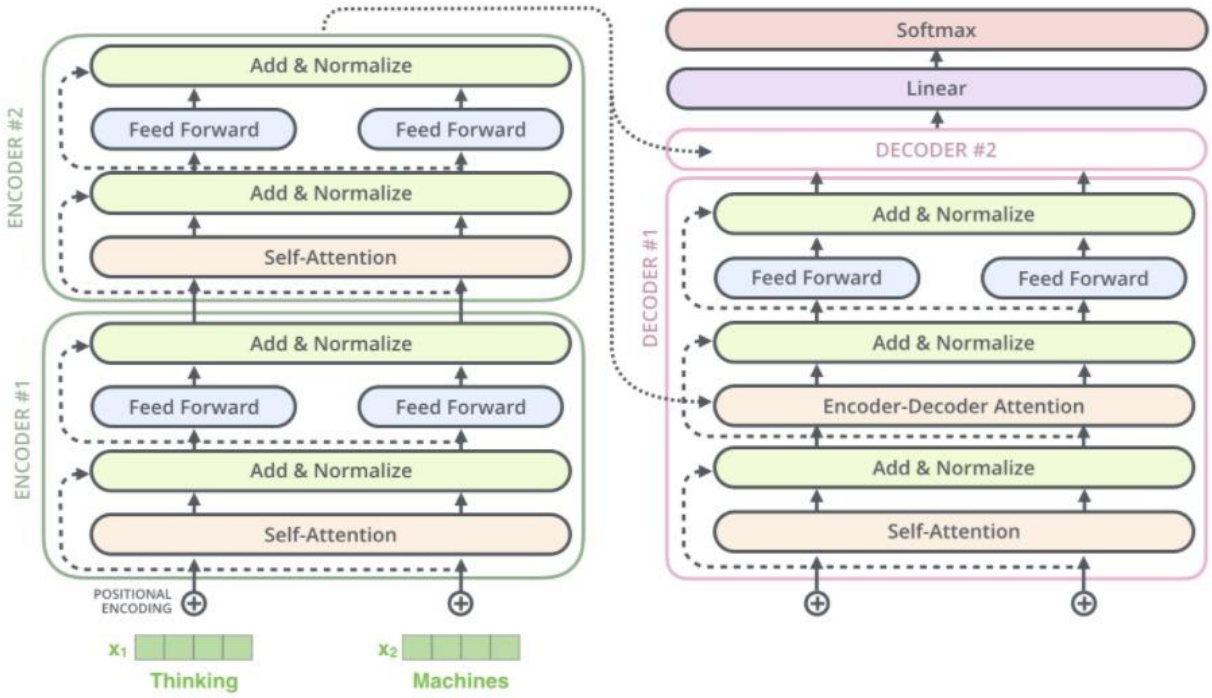
The proposed plagiarism detection system follows a structured approach consisting of input acquisition, preprocessing, feature extraction, similarity classification, and result generation. Initially, the system processes textual data, such as essays, research papers, or academic documents, for plagiarism detection. During the preprocessing stage, stopword removal eliminates common words that do not contribute significant meaning, while lemmatization converts words to their root forms to maintain consistency. In the feature extraction phase, BERT encoding transforms text into numerical vectors, capturing contextual meaning and semantic relationships. For similarity classification, k-Nearest Neighbors (KNN) is utilized to compare the BERT embeddings with reference documents using distance-based metrics. Finally, the system generates results in the form of a plagiarism score, representing the percentage similarity with existing texts, and a plagiarism status, which determines whether the text is plagiarized based on predefined thresholds.

Feedback:

The feedback generation system evaluates uploaded text using a rule-based model that applies multiple linguistic analysis methods. It first checks for grammar and spelling errors using predefined language rules. Next, it assesses readability by analyzing sentence complexity and structure. The system also evaluates sentence length, paragraph organization, and word count to ensure coherence and clarity. Finally, the model

compiles the results and generates detailed feedback, offering constructive suggestions for improving content quality, readability, and overall organization.

Prompt Based:



T5 converts all text processing problems into a "text-to-text" format (i.e., take text as input and produce text as output). This generic structure, which is also exploited by LLMs with zero/few-shot learning, allows us to model and solve a variety of different tasks with a shared approach.
After loading and preprocessing the dataset, the essays are tokenized with a tokenizer and transformed into PyTorch tensors for computational efficiency. Following batch grouping, these tensors are fed into a T5 model for training. In order to generate a score and feedback for an essay, the model first makes predictions through a forward pass. A loss function, which calculates the error, is then used to compare the expected outputs with the actual labels. Backpropagation is used to minimize this error, and automatic differentiation is used to calculate the gradients of the loss with respect to the model weights. The model's weights are then updated by the optimizer (AdamW) in a way that minimizes error. To avoid accumulation, gradients are reset prior to processing the subsequent batch. Until it learns to predict essay scores and feedback with accuracy, the model keeps training over several epochs.

IV. RESULT AND ANALYSIS

Training and Validation Results

Epoch	Training Loss	Validation Loss	Accuracy	F1	Precision	Recall
1	0.871000	0.859662	0.625796	0.622703	0.646228	0.625796
2	0.683000	0.776949	0.677813	0.675120	0.682303	0.677813
3	0.543700	0.874932	0.652070	0.652514	0.676209	0.652070

Validation Results:

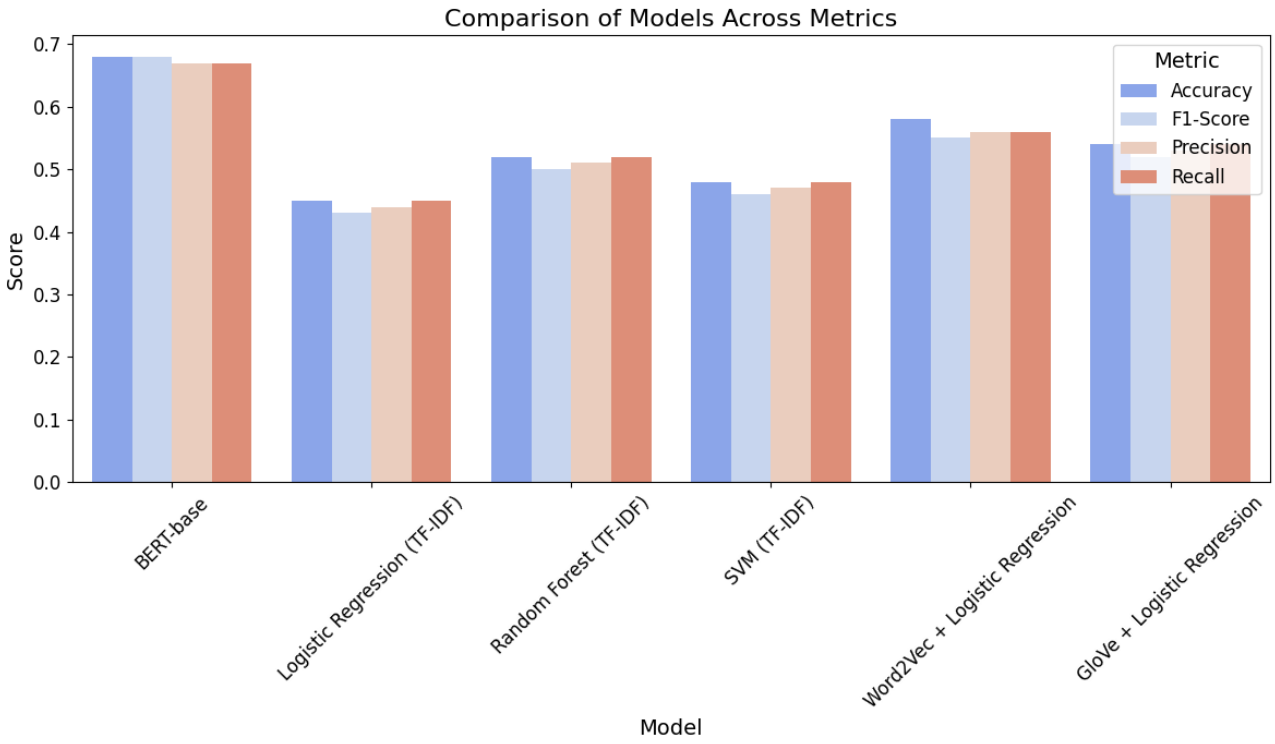
eval_loss: 0.7769498288330078
eval_accuracy: 0.6778131634819533
eval_f1: 0.6751199198386912
eval_precision: 0.682303302424202
eval_recall: 0.677813

Test Results:

eval_loss: 0.8109119862554658
eval_accuracy: 0.667462845016157
eval_f1: 0.6639458891394492
eval_precision: 0.6702594778194734
eval_recall: 0.667462

The BERT-based grading model was trained for three epochs, achieving a validation accuracy of **67.78%** with an F1-score of **67.51%**. Precision and recall values remained balanced, with the highest precision recorded at **68.23%**. Training loss consistently decreased, indicating effective learning, while validation loss fluctuated slightly.

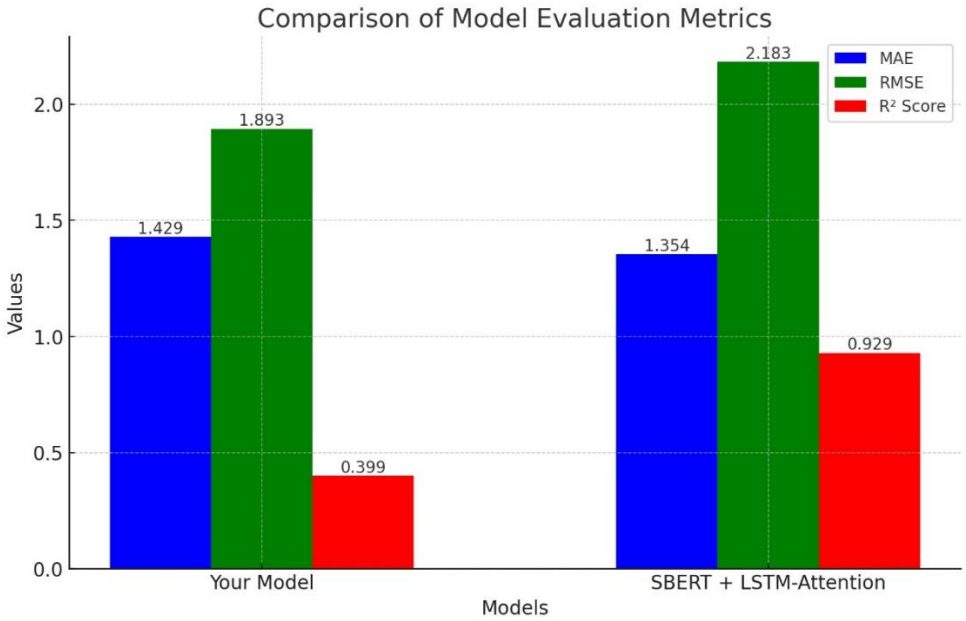
At 67.78% accuracy, our BERT-base model beats embedding models like Word2Vec and TF-IDF and more conventional models like Random Forest and Logistic Regression on important metrics like F1-score, precision, and recall. Word2Vec and TF-IDF provide some contextual information and semantic understanding, but they are not as good as BERT-base in terms of the richer context and depth of relations. When it comes to assessing essays, our grading system offers more reliable, accurate, and consistent scores thanks to the use of BERT-base's sophisticated contextual embeddings.



Evaluation Results:

Metric	Value
Mean Absolute Error (MAE)	1.4290
Root Mean Squared Error (RMSE)	1.8935
R^2 Score	0.3986

The prompt-based model's performance was assessed using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2 Score. The MAE was **1.429**, RMSE was **1.8935**, and the R^2 Score was **0.3986**, indicating moderate predictive accuracy. Further fine-tuning and data augmentation can enhance performance.



Our model has a lower RMSE (1.8935 vs. 2.1828), it makes more consistent and accurate predictions with fewer large deviations. This means our essay grading system produces results that are more reliable and stable than the SBERT + LSTM-Attention model

Unlike grading and prompt-based models, the performance of plagiarism detection and feedback generation cannot be directly measured using standard accuracy metrics. Their effectiveness depends on the quality and diversity of the input data.

V. CONCLUSION & FUTURE WORKS

The essay grading and plagiarism detection AI tool offers a great leap forward in automated marking, providing efficiency, equity, and individualized learning. With the use of BERT-based models for grading and plagiarism checking, and a T5-based system for prompt-based marking, the system guarantees objective marking, helpful feedback, and academic integrity. The application of sophisticated NLP methods ensures precision, with scope for additional fine-tuning to enhance performance. Ultimately, this system lightens the workload of educators, lessens human error, and empowers students with real-time, actionable feedback, making essay assessment more efficient and transparent.

In the future, we plan to improve the system's UI/UX for a smoother and more intuitive experience. By training on larger and more diverse datasets, we aim to enhance accuracy and adaptability. The feedback system will become more personalized, tailoring suggestions based on individual writing styles and progress. We also plan to commercialize the tool as a scalable SaaS platform for schools and universities. Our plagiarism detection will be refined for greater precision, while grading models will be fine-tuned to ensure fairness. Continuous learning from new data will make feedback even more insightful.

VI. REFERENCES

[1] S. Direct, "Measuring Text Similarity Based on Structure and Word Embedding," *ScienceDirect (Elsevier)*, 2020.

[2] Springer, "A New Hybrid Technique for Detection of Plagiarism from Text Documents," *Springer*, 2020.

[3] Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2020). Exploring the limits of transfer learning with a unified text-to-text transformer. *Journal of Machine Learning Research, 21*(140), 1-67. https://arxiv.org/abs/1910.10683

[4] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding.* arXiv. https://arxiv.org/abs/1810.04805

[5] Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Davison, J. (2020). *Transformers: State-of-the-art natural language processing.* arXiv. https://arxiv.org/abs/1910.03771

[6] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., & Desmaison, A. (2019). *PyTorch: An imperative style, high-performance deep learning library.* Advances in Neural Information Processing Systems, 32, 8024-8035. https://arxiv.org/abs/1912.01703

[7] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). *Scikit-learn: Machine learning in Python.* Journal of Machine Learning Research, 12, 2825-2830. https://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

[8] da Costa-Luis, C. (2019). *tqdm: A fast, extensible progress bar for Python and CLI.* Zenodo. https://doi.org/10.5281/zenodo.3479193

[9] LanguageTool contributors. (2024). *LanguageTool: Open-source proofreading software* [Software]. https://languagetool.org

[10] Shivam Bansal. (2021). *textstat: Calculate readability scores of text* [Software]. GitHub. https://github.com/shivam5992/textstat

[11] Nie Y. 2025. Automated essay scoring with SBERT embeddings and LSTM-Attention networks. *PeerJ Computer Science* 11:e2634

[12] Montani, I., & Honnibal, M. (2017). *spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing*. arXiv. https://arxiv.org/abs/1712.09405

[13] SciSpace, "Automated Essay Scoring in Argumentative Writing," *SciSpace*, [Online]. Available:https://scispace.com/pdf/automated-essay-scoring-in-argumentative-writing-2tuk1fl0.pdf.

[14] T. Foltynek, T. Ruas, P. Scharpf, N. Meuschke, M. Schubotz, W. Grosky, and B. Gipp, "Detecting Machine-obfuscated Plagiarism," *University of Michigan - Deep Blue Data*, 2019. [Online]. Available: https://doi.org/10.7302/bewj-qx93.

[15] P. Clough and M. Stevenson, "Developing a corpus of plagiarised short answers," *Language Resources & Evaluation*, vol. 45, pp. 5–24, 2011. [Online]. Available: https://doi.org/10.1007/s10579-009-9112-1.

[16] The Hewlett Foundation. (2012). *Automated Student Assessment Prize (ASAP) dataset* [Data set]. Kaggle. https://www.kaggle.com/c/asap-aes