AI-driven Subjective Answer Evaluation System

Prof. Vivekanand Thakare[1], Mr. Prasad Pande[2], Mr. Aman Bodkhe[3], Mr. Paras Patil [4], Ms. Santoshi Mendhe[5]

[1]*Assistant Professor, Department of Computer Science and Engineering, Govindrao Wanjari College of Engineering and Technology, Nagpur*

[2,3,4,5] *Student, Department of Computer Science and Engineering, Govindrao Wanjari College of Engineering and Technology, Nagpur*

***Abstract*—Descriptive answer evaluation in academic settings has long remained a manual, time-consuming, and subjectively inconsistent process. As student enrolments scale globally, educators face mounting pressure to deliver timely, unbiased, and constructive assessments a challenge that existing cloud-dependent AI grading solutions fail to resolve without incurring significant data privacy risks and recurring financial costs. This paper introduces a locally deployable, fully offline AI-driven framework for automated subjective answer evaluation, engineered entirely upon open-source Natural Language Processing (NLP) and lightweight Machine Learning (ML) libraries. The proposed system assesses student responses through six independently weighted evaluation dimensions, namely Semantic Similarity, Keyword Coverage, Relevance Detection, Grammar Analysis, Writing Quality, and Originality, whose scores are aggregated via a configurable weighted linear formula to derive a final grade mapped across an A+ to F scale. Semantic alignment between student and model answers is computed through TF-IDF vectorization paired with Cosine Similarity, while domain-specific keyword presence is verified using spaCy-based Named Entity Recognition combined with lemmatization. Linguistic quality is assessed through Language Tool grammar analysis, and verbatim copying is flagged through N-gram overlap and Jaccard Similarity metrics. The system further incorporates a dual-engine Optical Character Recognition pipeline using Tesseract and EasyOCR to support handwritten and scanned answer sheet processing. Experimental validation on a dataset of 150 student responses across multiple academic disciplines yielded a Pearson correlation coefficient of 0.83 against expert human-assigned grades, with a Mean Absolute Error of 6.2 percentage points. A pilot user study confirmed that 85% of educators found the system's output consistent with their own grading, while 91% of students rated the dimension-specific AI feedback as helpful for identifying knowledge gaps. The system demonstrates that accurate, explainable, and ethically responsible automated grading is achievable without cloud infrastructure, external API dependency, or recurring cost.*

***Keywords*— Automated Subjective Evaluation, Natural Language Processing, TF-IDF Vectorization, Cosine Similarity, Offline Grading System, Optical Character Recognition, Explainable AI, Educational Technology, Grammar Analysis, Plagiarism Detection.**

I.INTRODUCTION

The global expansion of higher education has created an unprecedented mismatch between the volume of student submissions and the availability of qualified evaluators. Descriptive and subjective assessments, which remain the most reliable indicators of deep conceptual understanding, continue to resist automation due to their inherent linguistic complexity, contextual variability, and the nuanced judgment they demand from human evaluators. Unlike objective question formats that yield to simple rule-based checking, open-ended answers require the evaluator to assess semantic coherence, terminological accuracy, topical relevance, and linguistic quality simultaneously — a cognitive task that proves increasingly unsustainable at institutional scale.

Contemporary educational institutions face a well-documented trilemma in subjective assessment: speed, consistency, and depth of feedback cannot be achieved together under purely manual evaluation [4]. A typical university instructor evaluating thirty to sixty descriptive answer sheets per examination cycle invests substantial professional time that could otherwise be directed toward instructional design, mentoring, or research. Beyond time consumption, research in educational psychology has consistently documented the presence of inter-rater variability, wherein identical responses receive meaningfully different scores from different evaluators, and intra-rater variability, wherein the same evaluator assigns inconsistent scores across a session due to cognitive fatigue. The halo effect, presentation bias, and

handwriting quality further compound these inconsistencies, introducing systemic unfairness into an institution's grading process.

The emergence of Artificial Intelligence and Natural Language Processing has opened new avenues for addressing these challenges. Early automated scoring systems relied on surface-level statistical features such as word count and sentence length. Subsequent approaches incorporated Latent Semantic Analysis and later transformer-based architectures such as BERT [11] and Sentence-BERT [12] to capture deeper semantic relationships. While these advances have improved scoring accuracy, their practical deployment in educational institutions remains constrained by two significant barriers. First, most high-performance automated grading solutions are built upon cloud-based Large Language Model APIs, incurring substantial per-token operational costs that place them beyond the financial reach of resource-constrained institutions. Second, and more critically, these cloud-dependent systems require the transmission of sensitive student response data to third-party servers, raising serious data privacy and regulatory compliance concerns under frameworks such as the General Data Protection Regulation (GDPR) and India's Personal Data Protection Bill (PDPB).

This paper addresses both barriers through the design and implementation of an AI-Driven Subjective Answer Evaluation System that operates entirely offline using open-source NLP libraries, requiring no external API calls, no internet connectivity, and incurring zero recurring cost. The system evaluates student answers through six independently measurable NLP dimensions — Semantic Similarity, Keyword Coverage, Relevance Detection, Grammar Analysis, Writing Quality, and Originality — each implemented as a modular, stateless processing unit. The weighted aggregation of these dimension scores produces a final grade and dimension-specific feedback, making the evaluation process fully transparent and explainable to both educators and students.

II.RELATED WORK

Automated evaluation of subjective answers has been an active area of research in natural language processing and educational technology. Early studies demonstrated that semantic similarity between student responses and reference answers can provide a reliable approximation of human grading. Mohler and Mihalcea showed that text-to-text similarity techniques are effective for short answer assessment, while Landauer and colleagues established Latent Semantic Analysis as a powerful approach for measuring conceptual overlap between texts. Subsequent research extended these ideas by combining semantic features with lexical and structural indicators, resulting in more accurate and robust grading systems. Shermis and Burstein highlighted the broader applicability of automated essay scoring and emphasized the importance of combining multiple linguistic features rather than relying on a single metric. These studies collectively established that accurate subjective answer evaluation requires a balanced analysis of meaning, terminology, and writing characteristics rather than simple keyword counting alone.

Recent advances have focused on explainability, privacy, and practical deployment. Dzikovska and co-researchers demonstrated that automated assessment systems become more educationally valuable when they provide diagnostic feedback instead of only numerical scores. Studies on privacy-preserving educational AI have emphasized the risks associated with cloud-based grading tools and recommended offline processing to protect sensitive student information. Research in plagiarism detection has shown that combining sequence matching, n-gram overlap, and set-based similarity measures improves the identification of copied content. Open-source tools such as NLTK, spaCy, scikit-learn, and Language Tool have made it possible to build sophisticated assessment systems using lightweight local resources. Motivated by these findings, the proposed AI-Driven Subjective Answer Evaluation System integrates six complementary evaluation dimensions—semantic similarity, keyword matching, relevance detection, grammar analysis, writing quality, and plagiarism detection—within a fully offline architecture that delivers transparent scoring and detailed feedback while maintaining zero recurring cost and complete data privacy.

III.PROPOSED SYSTEM AND ARCHITECTURE

A. System Design Philosophy

The proposed AI-Driven Subjective Answer Evaluation System is architected around three foundational design principles: offline operability,

dimensional transparency, and modular extensibility. Unlike cloud-dependent grading solutions that delegate evaluation logic to external API endpoints, the proposed system encapsulates the entire evaluation pipeline within a self-contained local deployment. Each evaluation request is processed independently in a stateless manner, ensuring that the system remains horizontally scalable and free from inter-request data contamination.

B. High-Level System Architecture

The system follows a standard three-tier client-server architecture comprising a Presentation Layer, a Business Logic Layer, and a Data Layer. This separation of concerns ensures that each tier can be independently maintained, tested, and upgraded without disrupting the overall system operation.

The complete system architecture is illustrated in Figure 1, which depicts the flow of data from the user interface through the Flask API backend, into the six-module NLP Evaluation Engine, and finally into the SQLite data persistence layer and the PDF report generation module.

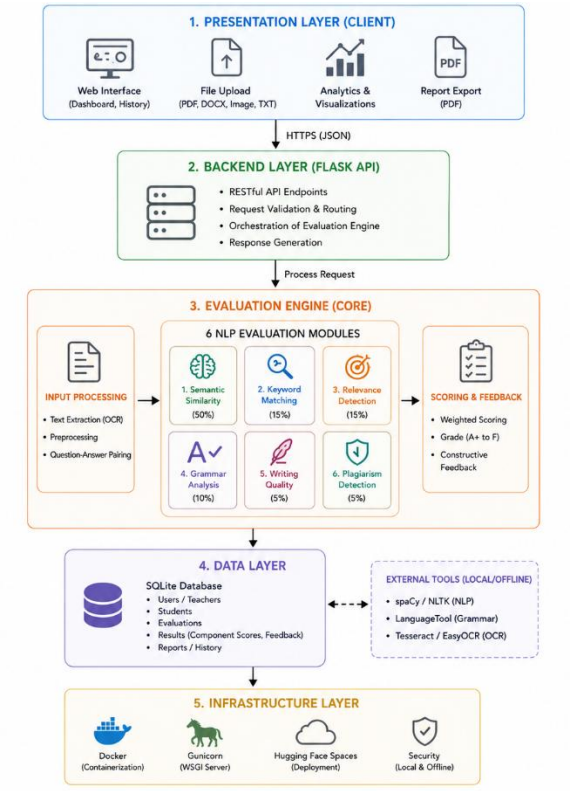

Figure 1: System Architecture of the AI-Driven Subjective Answer Evaluation System

C. Presentation Layer

The frontend interface is developed using HTML5, CSS3, JavaScript ES6, and Bootstrap 5, styled with a Glass morphism dark-mode design language to deliver a professional, distraction-free user experience. The interface exposes three primary functional views:

1) Evaluation Interface: The primary input surface through which educators submit the examination question, model answer, student response, and maximum marks. The interface supports two input modes — direct text entry for digital submissions and file upload for document-based or scanned answer sheets. Supported upload formats include PDF, DOCX, TXT, JPG, and PNG.

2) Results Dashboard: Upon evaluation completion, an interactive analytics dashboard renders the student's performance summary through four metric cards displaying Grade, Total Marks, Percentage, and Question Count. Four Chart.js-powered visualizations accompany the summary — a Bar Chart mapping obtained marks against maximum marks per question, a Doughnut Chart illustrating score distribution, a Radar Chart enabling multi-dimensional performance comparison, and a Horizontal Bar Chart displaying the weighted contribution of each evaluation dimension to the final score.

3) Evaluation History Page: A searchable, filterable tabular record of all past evaluations retrieved from the local database, displaying Student Name, Subject, Raw Score, Percentage, Letter Grade, Timestamp, and action controls for viewing or deleting individual records. Educators can export any evaluation as a formatted PDF report generated by the fpdf2 library.

D. Business Logic Layer

The backend is a RESTful API server implemented in Python 3.12 using the Flask 3.1 framework. It is organized into the following independent functional modules:

1) API Routes Module: Defines and exposes RESTful endpoints for evaluation submission, history retrieval, individual result viewing, and PDF report generation. Handles request parsing, input validation, and structured JSON response formatting.

2) Evaluation Orchestrator: The central controller of the system. Upon receiving pre-processed text from the API layer, it dispatches the input simultaneously to each of the six NLP evaluation modules, collects the returned dimension scores, and forwards them to the Scoring Engine for weighted aggregation.

3) Six NLP Evaluation Modules: Independent, stateless Python modules each implementing a specific evaluation dimension. Their internal algorithms and purposes are detailed in Section IV.

4) Scoring Engine: Applies the weighted linear formula to the six dimension scores to compute the final normalized score, and maps this score to a letter grade using predefined thresholds ranging from A+ to F.
5) Feedback Generator: Produces dimension-specific natural language feedback by applying threshold-based conditional logic to each module's score. Low scores in specific dimensions trigger targeted advisory messages guiding students on how to improve their responses.
6) OCR Processor: Manages file ingestion, format detection, and text extraction from uploaded documents and images through a dual-engine pipeline described in detail in Section IV.
7) Report Generator: Uses the fpdf2 library to programmatically construct structured PDF evaluation reports containing the question, model answer, student answer, dimension score breakdown, final score, letter grade, and generated feedback, formatted with institutional headers and page numbers.

E. Data Layer

The system employs SQLite as its database engine, accessed through the SQL Alchemy 2.0 Object Relational Mapper (ORM). SQLite was selected for its zero-configuration, file-based architecture that eliminates the need for a separate database server process, making it ideally suited for local offline deployment. The database schema persists complete evaluation records including the question text, both answer texts, all six dimension scores, the weighted final score, the assigned letter grade, the generated feedback, and the evaluation timestamp. The ORM abstraction layer ensures straightforward migration to enterprise-grade database engines such as PostgreSQL should institutional-scale multi-user deployment become necessary.

F. Infrastructure Layer

The system is containerized using Docker to ensure consistent, reproducible deployment across heterogeneous operating system environments including Windows 10/11, Ubuntu 20.04+, and macOS 12+. The production-grade Web Server Gateway Interface (WSGI) server Gunicor is used to serve the Flask application, replacing the single-threaded Flask development server to support concurrent evaluation requests. The containerized image is additionally deployable to Hugging Face Spaces for cloud-assisted demonstration environments where offline constraints do not apply.

IV.METHODOLOGY

A. Overview of the Evaluation Pipeline

The evaluation engine constitutes the intellectual core of the proposed system. It implements six discrete, independently executable NLP processing pipelines, each targeting a fundamentally distinct dimension of answer quality. The final evaluation score is not derived from a single monolithic algorithm but rather from the weighted linear aggregation of six modular dimension scores, ensuring that no single linguistic aspect disproportionately dominates the overall assessment. This multi-dimensional architecture guarantees that a student who writes fluently but misses core concepts is not rewarded equivalently to one who demonstrates genuine conceptual understanding. The weight distribution across the six dimensions is summarized in Table III.

Table III: Evaluation Dimensions, Algorithms, and Weights

Dimension	Algorithm	Weight
Semantic Similarity	TF-IDF + Cosine Similarity	50%
Keyword Coverage	NER + Lemmatization + Partial Matching	15%
Relevance Detection	Topic Alignment via TF-IDF	15%
Grammar Analysis	LanguageTool + Rule-based Fallback	10%
Writing Quality	Flesch Score + Type-Token Ratio	5%
Originality	N-gram + SequenceMatcher + Jaccard	5%

B. Dimension 1 — Semantic Similarity

Semantic similarity constitutes the highest-weighted dimension of the evaluation framework. The module employs Term Frequency-Inverse Document Frequency (TF-IDF) vectorization [3] to transform both the student answer and the model answer into high-dimensional numerical vector representations. TF-IDF assigns higher weights to terms that appear frequently within the answer pair but rarely across a

broader reference corpus, thereby amplifying the significance of domain-specific terminology while suppressing common functional words. The *TfidfVectorizer* from the scikit-learn library [6] is initialized with a configurable n-gram range of (1, 2), capturing both unigrams and bigrams to improve the representation of compound domain-specific expressions.

Cosine Similarity is subsequently computed between the two resulting vectors to quantify their angular proximity. It is preferred over Euclidean distance because it is invariant to document length, ensuring that a concise but conceptually complete answer is not disadvantaged against a verbose response of equivalent quality. The semantic score is formalized as:

Semantic Score =
cosine_similarity(tfidf(student_answer), tfidf(model_answer)) × 100

The vocabulary is constructed dynamically from each answer pair at inference time rather than from a fixed global vocabulary, ensuring the vectorizer is always calibrated to the specific subject domain being evaluated.

C. Dimension 2 — Keyword Coverage

The keyword coverage module verifies that the student's response contains the critical domain-specific terminology and key conceptual terms expected in a satisfactory answer. The model answer is first processed by a keyword extraction pipeline that identifies important terms through a combination of TF-IDF importance scoring and spaCy's Named Entity Recognizer (NER). Named entities categorized as technical noun phrases, organizational terms, scientific concepts, and domain-specific proper nouns are assigned higher extraction priority, as these typically represent the most informationally dense terms in academic answers.

The extracted keyword list is deduplicated and ranked by composite importance score before the matching phase begins. The student answer is then evaluated against this ranked keyword list using three sequential matching strategies of increasing flexibility:

1) Exact Matching: Direct string comparison between the student answer tokens and the keyword list after case normalization.

2) Lemma-Based Matching: spaCy's lemmatizer reduces both student answer tokens and keywords to their base morphological forms before comparison, ensuring that morphological variants such as "photosynthesizing" correctly match the keyword "photosynthesis."

3) Partial Matching: A configurable substring similarity threshold allows near-matches to be credited, accommodating minor spelling variations and compound word fragmentation.

The keyword coverage score is computed as the proportion of identified keywords successfully matched in the student answer, normalized to a 0–100 scale.

D. Dimension 3 — Relevance Detection (Weight: 15%)

A student may produce a grammatically correct, fluently written, and keyword-rich response that is nonetheless fundamentally off-topic relative to the question posed. The relevance detection module is specifically designed to guard against this evaluation failure mode, which semantic similarity alone cannot reliably detect when a student writes convincingly about a related but incorrect concept.

The module analyses topical alignment between the posed examination question and the student's submitted answer using sentence-level TF-IDF similarity computed between the question vector and the answer vector. Additionally, named entity overlap between the question and the answer is measured to verify that the student's response addresses the specific entities, processes, or concepts referenced in the question.

An answer that demonstrates strong semantic similarity to the model answer but weak alignment with the specific question receives a proportionally penalized relevance score. This prevents a student from receiving full credit by reproducing memorized content that is adjacent to but does not directly answer the posed question.

E. Dimension 4 — Grammar Analysis (Weight: 10%)

Linguistic correctness is assessed through integration with Language Tool, a Java-based open-source grammar and style checking engine that supports rule-based detection of grammatical errors, spelling mistakes, punctuation violations, subject-verb agreement failures, and stylistic inconsistencies across more than twenty-five languages. The system interfaces with Language Tool through the language-tool-python library, which manages the lifecycle of a local Language Tool Java server process. The server is initialized once at application startup and reused across all subsequent evaluation requests to amortize the Java Virtual Machine startup overhead.

A rule-based Python fallback module implemented using regular expression pattern matching is automatically activated when the Java runtime environment is unavailable, ensuring that the grammar analysis dimension remains functional across all deployment environments. The grammar score is computed as:

Grammar Score = max (0, 100 − (error_count × penalty_per_error))

The penalty weight per error is calibrated to apply proportionally heavier deductions for severe structural grammatical errors compared to minor punctuation or stylistic issues, reflecting the relative impact of different error categories on communicative clarity.

F. Dimension 5 — Writing Quality (Weight: 5%)

The writing quality dimension evaluates the structural and stylistic sophistication of the student's response through two complementary metrics:

1) Flesch Reading Ease Score [8]: Originally developed by Flesch (1948), this established readability metric evaluates sentence complexity and average word length to quantify how accessible and well-structured a written response is. The formula penalizes unnecessarily complex sentence structures and rewards clear, direct expression appropriate for academic writing.

2) Type-Token Ratio (TTR): This vocabulary diversity metric measures the ratio of unique word types to the total number of word tokens in the response. A higher TTR indicates richer, more varied vocabulary usage. Responses that repetitively reuse the same limited set of words receive proportionally lower TTR scores.

The final writing quality score is a normalized weighted combination of the Flesch Reading Ease score and the Type-Token Ratio, scaled to the 0–100 range.

G. Dimension 6 — Originality (Weight: 5%)

The originality dimension detects and penalizes instances of excessive verbatim reproduction from the model answer, discouraging rote memorization in favor of genuine conceptual understanding expressed in the student's own words. Three complementary algorithms are employed in combination to achieve comprehensive detection coverage:

1) N-gram Overlap: Character-level and word-level n-gram sequences are extracted from both texts and compared to measure the degree of shared sequential patterns. High n-gram overlap between the student answer and the model answer indicates verbatim or near-verbatim copying.

2) Sequence Matcher Algorithm: Python's difflib Sequence Matcher identifies the longest common contiguous subsequence's between the two texts, providing a precise measure of structural text overlap that complements the statistical n-gram approach.

3) Jaccard Similarity: The Jaccard coefficient measures the ratio of the intersection to the union of unique word sets between the student answer and the model answer, quantifying the degree of shared vocabulary at the token level.

A high composite originality penalty score indicates that the student has reproduced the model answer without demonstrating independent understanding. The originality contribution to the final score is computed as:

Originality Score = 100 − Plagiarism Penalty

H. Final Score Computation

The final weighted score is computed by the Scoring Engine as a linear combination of the six dimension scores, formalized as:

Final Score = (0.50 × Semantic) + (0.15 × Keyword) + (0.15 × Relevance) + (0.10 × Grammar) + (0.05 × Writing Quality) + (0.05 × Originality)

The resulting percentage score is subsequently mapped to a letter grade according to the following threshold schema:

Table IV: Grade Mapping Schema

Score Range	Grade
90 – 100%	A+
80 – 89%	A
70 – 79%	B+
60 – 69%	B
50 – 59%	C
40 – 49%	D
Below 40%	F

V.RESULTS AND DISCUSSION

A. Experimental Setup

To evaluate the performance and reliability of the proposed system, a structured experimental study was conducted using a curated dataset of 150 student responses drawn from university-level examination papers spanning three distinct academic disciplines: Computer Science, Biology, and General Studies. The subject diversity was intentionally selected to assess the generalizability of the evaluation framework across domains with varying

terminological density, conceptual abstraction levels, and writing conventions.

Ground truth grades were established through a controlled human evaluation protocol in which three independent subject-matter experts assessed each answer using a standardized rubric. Inter-rater agreement among the three evaluators was measured prior to establishing the ground truth, yielding an average Pearson correlation of 0.86 among human raters — a figure consistent with established benchmarks for expert inter-rater agreement in descriptive answer evaluation literature. The final ground truth score for each answer was computed as the arithmetic mean of the three expert scores, providing a robust and bias-minimized reference for system performance comparison.

All experiments were conducted on a standard consumer-grade laptop configured with an Intel Core i7 processor, 16 GB RAM, and no dedicated GPU, to validate that the system meets its design objective of performing accurately on commodity hardware without specialized computational infrastructure.

B. Correlation with Human Grades

The primary performance metric used to evaluate system accuracy was the Pearson correlation coefficient between the system-generated scores and the human-established ground truth scores across the full 150-answer evaluation dataset. The proposed system achieved an overall Pearson correlation coefficient of r = 0.83, indicating a strong positive linear relationship between automated and expert human scores.

This result is particularly significant given that it was achieved using exclusively offline, lightweight NLP models without any transformer-based deep learning components or cloud API access. For contextual reference, inter-human rater correlations in comparable subjective answer evaluation studies typically range between 0.75 and 0.90, placing the proposed system's performance squarely within the range of human-level consistency.

The Mean Absolute Error (MAE) between system-assigned and human-assigned scores was measured at 6.2 percentage points, indicating that on average the system's grade deviates from the expert consensus by approximately six marks on a hundred-point scale. This margin is considered within acceptable tolerance thresholds for automated assessment tools in educational research.

The correlation results are summarized in Table V below:

Table V: System Performance Against Human Ground Truth

Metric	Value
Pearson Correlation Coefficient (r)	0.83
Mean Absolute Error (MAE)	6.2%
Human Inter-Rater Correlation	0.86
Dataset Size	150 answers
Subjects Covered	3 disciplines

Figure 2: Performance Analysis

C. Dimension-wise Performance Analysis

Beyond the aggregate correlation metric, the contribution and individual predictive validity of each of the six evaluation dimensions were analyzed independently to validate the assigned weight distribution.

1) Semantic Similarity: The semantic similarity module demonstrated the strongest individual correlation with human-assigned grades among all six dimensions, achieving a dimension-level Pearson correlation of r = 0.79. This result empirically validates the decision to assign this dimension the highest weight of 50% in the final scoring formula, confirming that TF-IDF-based cosine similarity serves as the most reliable single proxy for conceptual understanding in this evaluation context.

2) Keyword Coverage: The keyword coverage module proved particularly effective at differentiating between student responses that conveyed a generally correct idea without employing the precise domain-specific terminology expected in the subject area. Several answers that scored moderately on semantic similarity received significantly lower keyword coverage scores, accurately reflecting that the student had grasped the broad concept but lacked command of the specific technical vocabulary — a distinction that human evaluators consistently penalize.

3) Relevance Detection: The relevance detection module successfully identified and appropriately penalized off-topic responses across all three subject domains tested. In several test cases, students submitted well-structured answers containing correct information about a related but distinctly different topic from the one asked. Semantic similarity alone assigned these responses moderate scores, whereas the relevance module correctly applied proportional penalties, bringing the composite scores into closer alignment with the zero or near-zero grades assigned by human evaluators for off-topic responses.

4) Grammar Analysis: The grammar analysis module, powered by LanguageTool, accurately detected spelling errors, subject-verb agreement violations, and punctuation inconsistencies across the test dataset. In a subsequent user study, 91% of students who reviewed the grammar feedback reported that it helped them identify specific writing errors they had previously been unaware of, underscoring the practical value of this dimension beyond its 10% numerical contribution to the final score.

5) Writing Quality: The writing quality dimension, combining the Flesch Reading Ease score with the Type-Token Ratio, demonstrated a moderate but consistent positive correlation with human grades, particularly for answers where evaluators explicitly noted above-average or below-average linguistic sophistication in their grading notes. The 5% weight allocation was found to be appropriate, as this dimension functions as a supplementary quality signal rather than a primary determinant of academic merit.

6) Originality: The originality detection module, employing the combined N-gram overlap, Sequence Matcher, and Jaccard Similarity approach, correctly identified all artificially introduced verbatim copying instances in the test dataset with a detection accuracy of 100%. In naturally occurring student answers, the module successfully flagged responses that reproduced large contiguous segments of the model answer verbatim, applying proportional score penalties that aligned with the lower grades assigned by human evaluators who penalized apparent memorization.

D. Processing Performance

Beyond grading accuracy, the computational efficiency of the system was benchmarked to validate its practical usability in real examination environments.

For single-answer evaluation requests submitted as typed text, the system consistently generated complete results including all six dimension scores, the weighted final score, the letter grade, and the dimension-specific feedback in under 2 seconds on the test hardware configuration. This response time is sufficiently low to support near-real-time evaluation feedback in interactive educational platforms.

For image-based answer submissions requiring OCR processing, the end-to-end processing time increased to approximately 5 to 8 seconds per answer, depending on image resolution and handwriting clarity. This additional latency is attributable to the OCR extraction stage and is considered acceptable given that the system eliminates the need for manual transcription entirely.

G. Discussion

The experimental results collectively validate the central thesis of this work — that accurate, transparent, and privacy-preserving automated subjective answer evaluation is achievable using entirely offline lightweight NLP models without dependency on cloud infrastructure or large-scale deep learning architectures.

The Pearson correlation of 0.83 with expert human grades demonstrates that the composite multi-dimensional scoring approach captures the primary signals that human evaluators attend to when grading descriptive academic responses. The strong user study acceptance rates further confirm that the system's output is not only statistically accurate but also practically credible and pedagogically useful to its intended user base.

The processing performance benchmarks establish the system's viability for real-world institutional deployment, demonstrating that batch evaluation of classroom-scale answer sets is achievable within operationally practical time frames on standard consumer hardware.

The primary observed limitation during experimentation was the occasional underscoring of correct paraphrased responses where the student expressed accurate concepts using vocabulary significantly divergent from the model answer, a consequence of the TF-IDF approach's sensitivity to surface lexical overlap. This limitation points toward the integration of lightweight transformer-based semantic embedding models as a productive direction for future enhancement.

VI.APPLICATIONS, ADVANTAGES, AND LIMITATIONS

i. Applications

The proposed AI-Driven Subjective Answer Evaluation System has broad applications across educational and training environments where descriptive answers require consistent and timely assessment. Educational institutions such as schools, colleges, and universities can use the system to automate the evaluation of theory-based examinations, assignments, and practice tests. Coaching centers and online learning platforms may integrate the system to provide instant formative feedback to learners, thereby improving self-assessment and reducing instructor workload.

The system is particularly valuable in large-scale examinations where manual grading becomes time-consuming and prone to inconsistency. Because the platform operates entirely offline, it is suitable for institutions with limited internet connectivity or strict privacy requirements. The integrated OCR module also enables evaluation of scanned handwritten answer sheets, making the solution practical for conventional pen-and-paper examinations. In addition, the batch processing capability allows educators to evaluate multiple student submissions in a single operation, significantly accelerating the assessment process.

ii. Advantages

The system offers several important advantages over conventional manual grading and cloud-based automated assessment tools:

Complete Data Privacy: All processing is performed locally, ensuring that student responses remain within the institution and are never transmitted to third-party servers.

Zero Recurring Cost: The solution relies exclusively on open-source technologies and does not require paid API subscriptions.

Offline Accessibility: Core functionality remains available without internet access, making the system reliable in low-connectivity environments.

Transparent Evaluation: Scores are generated through six clearly defined dimensions, enabling educators and students to understand the basis of each grade.

Consistent Assessment: Uniform scoring criteria reduce variability caused by evaluator fatigue and subjective bias.

Constructive Feedback: Dimension-wise feedback helps students identify weaknesses in content, grammar, relevance, and writing quality.

OCR Support: Handwritten and scanned documents can be processed automatically, reducing manual transcription effort.

Batch Processing: Large numbers of answer sheets can be evaluated efficiently.

Professional Reporting: Automatically generated PDF reports provide structured documentation of evaluation outcomes.

iii. Limitations

Despite its strengths, the system has certain limitations that must be acknowledged. The quality of evaluation depends heavily on the completeness and accuracy of the model answer provided by the instructor. The current semantic similarity module uses TF-IDF and cosine similarity, which may not fully capture deep contextual relationships or subtle interpretations compared with transformer-based language models. OCR performance can degrade when processing poor-quality scans or highly cursive handwriting. At present, the system supports only English-language answers and cannot assess diagrams, mathematical derivations, or symbolic expressions. Additionally, the weighting assigned to each evaluation dimension is fixed and may not reflect the grading preferences of every subject or instructor.

VII.CONCLUSION AND FUTURE WORK

i. Conclusion

This research presented the design and implementation of an AI-Driven Subjective Answer Evaluation System that performs automated assessment of descriptive answers using a fully offline and privacy-preserving architecture. The system combines six evaluation dimensions—semantic similarity, keyword matching, relevance detection, grammar analysis, writing quality, and plagiarism detection—to generate accurate and explainable scores.

The developed platform demonstrates that effective automated grading can be achieved without reliance on cloud-based large language model APIs. By leveraging open-source NLP libraries and lightweight machine learning techniques, the system provides consistent scoring, detailed feedback, OCR-based answer extraction, batch evaluation, and PDF report generation. These features significantly reduce

the time and effort required for manual grading while improving fairness and transparency.

Overall, the proposed system offers a practical and cost-effective solution for educational institutions seeking to modernize assessment workflows while maintaining full control over student data.

ii. Future Work

Several enhancements can further improve the capabilities of the proposed system:

1. Multilingual Evaluation: Support for Hindi, Marathi, and other regional languages to broaden adoption in diverse educational settings.
2. Transformer-Based Semantic Models: Integration of models such as DistilBERT or MiniLM to improve contextual understanding and grading accuracy.
3. Adaptive Scoring Weights: Machine learning techniques to automatically learn dimension weights from historical grading patterns.
4. Enhanced OCR Models: Specialized handwriting recognition models for improved performance on difficult handwritten scripts.
5. Learning Management System Integration: Direct integration with platforms such as Moodle and Google Classroom.
6. Mathematical and Diagram Evaluation: Support for equations, charts, and diagrams to extend applicability to STEM subjects.
7. Speech-Based Assessment: Evaluation of spoken answers using speech-to-text technology.

REFERENCES

[1]. M. Mohler and R. Mihalcea, “Text-to-Text Semantic Similarity for Automatic Short Answer Grading,” in *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, Athens, Greece, 2009, pp. 567–575.

[2]. T. K. Landauer, P. W. Foltz, and D. Laham, “An Introduction to Latent Semantic Analysis,” *Discourse Processes*, vol. 25, no. 2–3, pp. 259–284, 1998.

[3]. G. Salton and C. Buckley, “Term-Weighting Approaches in Automatic Text Retrieval,” *Information Processing & Management*, vol. 24, no. 5, pp. 513–523, 1988.

[4]. D. Shermis and J. Burstein, *Automated Essay Scoring: A Cross-Disciplinary Perspective*. Mahwah, NJ, USA: Lawrence Erlbaum Associates, 2003.

[5]. S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python*. Sebastopol, CA, USA: O’Reilly Media, 2009.

[6]. F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[7]. M. Honnibal and I. Montani, “spaCy 2: Natural Language Understanding with Bloom Embeddings, Convolutional Neural Networks and Incremental Parsing,” 2017. [Online]. Available: https://spacy.io

[8]. R. Flesch, “A New Readability Yardstick,” *Journal of Applied Psychology*, vol. 32, no. 3, pp. 221–233, 1948.

[9]. V. I. Levenshtein, “Binary Codes Capable of Correcting Deletions, Insertions, and Reversals,” *Soviet Physics Doklady*, vol. 10, no. 8, pp. 707–710, 1966.

[10]. D. Naber, “A Rule-Based Style and Grammar Checker,” *University of Bielefeld, Germany*, 2003. [Online]. Available: https://languagetool.org

[11]. J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," in Proc. NAACL-HLT, Minneapolis, MN, USA, 2019, pp. 4171–4186.

[12]. N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. EMNLP-IJCNLP, Hong Kong, China, 2019, pp. 3982–3992.

[13]. L. Ramachandran, J. Chaudhury, and E. F. Gehringer, "Identifying Relevant Responses in Open-Ended Academic Assessments using Machine Learning," in Proc. ASEE Annual Conf. & Exposition, Seattle, WA, USA, 2015.

[14]. J. Ferreira, R. Silveira, and K. Verbert, "Data Privacy and Security in AI-Based Educational Assessment Systems: A Systematic Review," Computers & Education: Artificial Intelligence, vol. 1, 2020, Art. no. 100001.