**Smart Evaluation of Handwritten Responses with RAG**

**Introduction**

**Question and Answer Evaluation**

The process of evaluating question-and-answer tasks has traditionally relied on manual effort, especially in educational settings. It involves assessing the accuracy, relevance, and completeness of the responses against predefined correct answers. While straightforward for objective questions, evaluating subjective and handwritten responses poses unique challenges, including understanding varied handwriting, gauging semantic relevance, and maintaining evaluation consistency. This project focuses on automating this process by integrating advanced technologies like Handwritten Text Recognition (HTR) and Retrieval-Augmented Generation (RAG). These technologies enable a robust pipeline that converts handwritten text into digital format, retrieves relevant answers, and evaluates responses based on semantic and syntactic criteria.

**Need for Automation in Question and Answer Evaluation**

Automation in evaluating handwritten question-and-answer scripts addresses critical challenges such as scalability, consistency, and fairness. Manual evaluation is labor-intensive, error-prone, and lacks uniformity, especially when evaluating large-scale assessments. Automated systems eliminate biases and expedite the process, ensuring accurate evaluation in diverse contexts, including remote and online learning environments. The advent of technologies like OCR, semantic similarity models, and natural language processing (NLP) has opened avenues to overcome the inefficiencies of traditional methods, paving the way for objective and reliable assessment frameworks.

**Role of AI in Question and Answer Evaluation**

Artificial Intelligence (AI) plays a transformative role in question-and-answer evaluation by mimicking human cognitive processes like comprehension, comparison, and decision-making. AI-powered OCR models, such as Microsoft TrOCR, excel in accurately converting handwritten text into digital formats. AI-driven frameworks like RAG use advanced NLP models, such as Google FLAN-T5, to retrieve and synthesize relevant information from textual sources. Additionally, semantic similarity models and transformer-based architectures ensure precise evaluation by comparing the context, syntax, and semantic meaning of responses. These advancements allow AI to handle complex tasks, such as evaluating subjective answers with high accuracy.

**Significance of AI in Question and Answer Evaluation**

AI enhances question-and-answer evaluation by introducing scalability, consistency, and context-aware scoring. It can analyze vast datasets, adapt to varied handwriting styles, and evaluate semantic content beyond surface-level correctness. By leveraging AI, educators and institutions can ensure fairness in grading, reduce human bias, and provide instant feedback to learners. AI's ability to measure deeper semantic relationships between answers and reference content strengthens its significance in modern education, especially in domains requiring large-scale and quick evaluations.

**Current State-of-the-Art AI Models in Question and Answer Evaluation**

Several state-of-the-art models and approaches have been proposed to address challenges in this domain:

* **TrOCR for Handwritten Text Recognition**  
  Microsoft’s TrOCR model utilizes Transformer-based architecture to accurately recognize handwritten texts, addressing spacing and handwriting style variations (Handwritten Text Processing).
* **Cosine Similarity for Semantic Comparison**  
  Cosine Similarity is widely used to measure semantic alignment between student answers and reference answers. It effectively quantifies text similarity, even with synonym usage (Semantic Evaluation Techniques).
* **Neural Networks for Text Scoring**  
  Deep learning techniques, such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs), have been successfully applied for automated scoring of textual answers based on semantic and syntactic features (Deep Learning in Answer Scoring).
* **Knowledge Augmentation for QA Models**  
  Retrieval-Augmented Generation (RAG) integrates external knowledge into transformer models, allowing them to provide accurate and contextually rich responses in subjective evaluations (Knowledge-Enhanced Answer Synthesis).
* **Challenges and Future Directions**  
  Despite advancements, challenges such as diverse answer formats, grammatical errors, and synonym usage persist. Future research aims to develop robust models that handle language variability and noisy data more effectively.

**Motivation**

The primary motivation for this project stems from the need for objective, scalable, and efficient evaluation systems in education. Manual evaluation often suffers from subjectivity, inconsistency, and inefficiency, especially in large-scale or remote learning setups. Automation offers the potential to address these challenges while also fostering inclusivity by enabling assessments in diverse settings. The incorporation of cutting-edge AI technologies further enhances the system's ability to deliver accurate, fair, and real-time evaluations.

**Contribution**

This project makes several notable contributions:

1. **Handwritten Text Conversion:** Implementation of a fine-tuned Microsoft TrOCR model for accurate handwriting-to-text conversion.
2. **Answer Retrieval and Synthesis:** Use of RAG architecture, combining embedding-based retrieval (using SentenceTransformers and ChromaDB) with contextual answer generation (using Google FLAN-T5).
3. **Automated Evaluation:** Development of a hybrid scoring mechanism combining semantic similarity metrics (cosine similarity, WM distance) and TS-GR-based term weighting for comprehensive answer evaluation.
4. **Innovation in Scoring:** Introduction of TS-GR and synonym mapping for context-aware scoring and enhanced semantic relevance detection.
5. **End-to-End Workflow:** Design of a seamless pipeline for digitizing handwritten scripts, generating answers, and performing automated evaluations with real-world applicability.

This project sets a foundation for future advancements in AI-driven education technologies and demonstrates the practical benefits of integrating state-of-the-art models into assessment workflows.

**Literature Analysis**

The paper “***Handwritten English Character Recognition Using Swarm Intelligence and Neural Network***”, proposes a hybrid methodology for handwritten English character recognition using Independent Component Analysis (ICA) for feature extraction, a combination of Particle Swarm Optimization (PSO) and Firefly Algorithm (FFA) for feature selection, and a Backpropagation Neural Network (BPNN) for classification. The approach achieves high sensitivity (98.72%) and recognition rates (98.25%) on the MNIST dataset. However, its applicability may be limited for datasets with more diverse or noisy real-world inputs.

| **Paper Title** | **Proposed Work** | **Technical Algorithm Used** | **Parameters and Evaluation** | **Data Set** | **Observed Drawback** |
| --- | --- | --- | --- | --- | --- |
| Handwritten English Character Recognition Using Swarm Intelligence and Neural Network | A hybrid system combining feature extraction and classification for handwritten English character recognition. | Hybrid PSO and Firefly Optimization; BPNN | Recall 98.72%; True Negative Rate: 96.74%; Recognition Rate: 98.25% | MNIST dataset | Limited focus on handling real-world variations such as noise or variations beyond MNIST scope. |
|  |  |  |  |  |  |

The paper “***End-to-End Page-Level Assessment of Handwritten Text Recognition***”, presents a novel two-fold evaluation framework for end-to-end handwritten text recognition at the page level, emphasizing transcription accuracy and reading order assessment. By integrating metrics like Word Error Rate (WER), Bag of Words WER (bWER), and the Hungarian Algorithm-based WER (hWER), the proposed approach efficiently identifies and quantifies layout analysis and ordering errors, enhancing system robustness. However, the hWER computation is computationally intensive, posing scalability challenges for larger datasets.

| **Paper Title** | **Proposed Work** | **Technical Algorithm Used** | **Parameters and Evaluation** | **Data Set** | **Observed Drawback** |
| --- | --- | --- | --- | --- | --- |
| End-to-End Page-Level Assessment of Handwritten Text Recognition | Two-fold evaluation for HTR systems separating transcription accuracy and reading order (RO) assessment | Regularized Hungarian Algorithm, Bag of Words | Metrics: WER, bWER, hWER, and NSFD. Evaluated across multiple datasets like ICFHR14 and IAMDB. | ICFHR14, IAMDB, ICDAR17, and others | Computational complexity of hWER is higher due to O(N³) cost in alignment. |

The paper *"****Enhancing Handwritten Text Recognition Accuracy with Gated Mechanisms****",* proposes a Gated-CNN-BGRU architecture for HTR, achieving superior recognition performance with fewer parameters (~830,000) than traditional models. Evaluated on IAM, RIMES, and other datasets, it reduces CER and WER significantly, addressing challenges in handling complex handwriting styles and variable noise levels.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Paper Title** | **Proposed Work** | **Technical Algorithm Used** | **Parameters and Evaluation** | **Data Set** | **Observed Drawback** |
| Enhancing Handwritten Text Recognition Accuracy with Gated Mechanisms | Improved HTR using Gated-CNN-BGRU with enhanced accuracy and reduced computational complexity. | Gated-CNN-BGRU | CER and WER values reduced across datasets, outperforming prior models | IAM, RIMES, Bentham, Washington, Saint Gall | Small datasets like Washington and Saint Gall are prone to overfitting; punctuation poses challenges. |

The paper **"*HCR-Net: A Deep Learning Based Script Independent Handwritten Character Recognition Network*",** presents HCR-Net, a deep learning framework leveraging partial transfer learning with VGG16 for script-independent handwritten character recognition. The network achieves state-of-the-art accuracy across 40 datasets from diverse scripts, establishing 26 new benchmarks. Despite achieving significant parameter efficiency and convergence speed, the reliance on clean datasets shows diminished augmentation benefits, indicating scope for optimization in noisy or augmented scenarios. This work sets a novel direction for robust, multilingual character recognition in constrained computational settings.

| **Paper Title** | **Proposed Work** | **Technical Algorithm Used** | **Parameters and Evaluation** | **Dataset** | **Observed Drawback** |
| --- | --- | --- | --- | --- | --- |
| HCR-Net: A deep learning-based script-independent handwritten character recognition network | A script-independent deep learning network (HCR-Net) for offline handwritten character recognition using transfer learning | CNN with VGG16-based transfer learning | Accuracy, precision, recall, F1-score; achieves up to 11% improvement, 34% parameter reduction, fast convergence up to 99% of performance in the first epoch | 40 datasets from diverse languages (e.g., Bangla, Hindi, Swedish, Tibetan, Arabic) | Computational efficiency is dataset-dependent; limited improvement with clean datasets using augmentation |

The paper *"****Advanced RAG Models with Graph Structures****"* introduces a novel RAG model leveraging Graph Neural Networks (GNNs) to process structured knowledge, enhancing reasoning and knowledge consistency. Experimental results on the Natural Questions dataset validate its superiority over baseline models in quality (0.90), consistency (0.85), and reasoning (0.91). However, performance gains plateau with excessive document retrieval, highlighting information redundancy limitations.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Paper Title** | **Proposed Work** | **Technical Algorithm Used** | **Parameters and Evaluation** | **Dataset** | **Observed Drawback** |
| Advanced RAG Models with Graph Structures | Proposes integrating Graph Neural Networks (GNNs) into Retrieval-Augmented Generation (RAG) models to enhance reasoning and text generation by leveraging graph structures. | Graph Neural Networks (GNN), Transformer architecture | Metrics: Quality (0.90), Knowledge Consistency (0.85), Reasoning Capability (0.91); Comparative analysis with other models | Natural Questions (NQ) dataset | Limited improvement beyond optimal document retrieval size due to information redundancy |

The paper *"****LongRAG: Enhancing Retrieval-Augmented Generation with Long-context LLMs"*** redefines the Retrieval-Augmented Generation (RAG) paradigm by incorporating long-context LLMs. It reduces retrieval unit granularity, significantly alleviating retriever burdens and preserving semantic integrity. Evaluations across four datasets demonstrate superior performance without additional training, though reliance on advanced LLMs poses challenges for open-source implementations.

| **Paper Title** | **Proposed Work** | **Technical Algorithm Used** | **Parameters and Evaluation** | **Dataset** | **Observed Drawback** |
| --- | --- | --- | --- | --- | --- |
| LongRAG: Enhancing Retrieval-Augmented Generation with Long-context LLMs | Introduces a framework with long retrievers and readers to process long-context retrieval units, reducing retriever workload and improving semantic completeness. | Retrieval-Augmented Generation (RAG) with long retrieval units | Achieved EM: 62.7% on NQ, 64.3% on HotpotQA; F1: 25.9% on Qasper, 57.5% on MultiFieldQA-en | NQ, HotpotQA (Wikipedia-based); Qasper, MultiFieldQA-en (non-Wikipedia-based) | Relies heavily on the capability of LLMs to handle long contexts, limiting performance on open-source LLMs. |

The paper *"****Implementation of Handwriting Recognition and Answer Evaluation with Recurrent Neural Network****"* proposes a handwriting recognition system using CNN and BLSTM for evaluating handwritten answers, achieving a validation accuracy of 50.54% on the IAM dataset. While the approach automates scoring with CTC decoding, it is limited by segmentation inaccuracies and struggles with complex essay inputs, highlighting the need for NLP enhancements and better UI for scalability.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Paper Title** | **Proposed Work** | **Technical Algorithm Used** | **Parameters and Evaluation** | **Data Set** | **Observed Drawback** |
| Handwritten English Character Recognition Using Swarm Intelligence and Neural Network | A hybrid system combining feature extraction and classification for handwritten English character recognition. | Hybrid PSO and Firefly Optimization; BPNN | Sensitivity: 98.72%; Specificity: 96.74%; Recognition Rate: 98.25% | MNIST dataset (60k train, 10k test) | Limited focus on handling real-world variations such as noise or variations beyond MNIST scope. |

The automated evaluation of subjective answers remains a challenge due to language variability and semantic nuances. The Paper ***"Subjective Answers Evaluation Using Machine Learning and Natural Language Processing****"*, proposed a method integrating Word2Vec embeddings with Word Mover’s Distance and Multinomial Naive Bayes for enhanced semantic similarity evaluation, achieving 88% accuracy. However, limitations include reliance on curated datasets and challenges in semantic generalization across diverse domains.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Paper Title** | **Proposed Work** | **Technical Algorithm Used** | **Parameters and Evaluation** | **Data Set** | **Observed Drawback** |
| Subjective Answers Evaluation Using Machine Learning and NLP | Automated scoring of subjective answers using semantic similarity and ML | Word2Vec, WMD, TF-IDF, Cosine Similarity, MNB | Accuracy (88%), F1-score, Recall, Precision | Custom corpus of 1000 questions and 20 answers each | Limited data, semantic generalization challenges |

The paper *"****Automated Evaluation of Handwritten Answer Script Using Deep Learning Approach****"* develops an automated evaluation system for handwritten answer scripts using a hybrid CNN and BiLSTM architecture. The system combines Optical Character Recognition (OCR) for text recognition with sequential grading based on vectorized answers. It achieves a training accuracy of ~90% and a test accuracy of ~80% on datasets from IAM and Hamdard University Bangladesh. Despite its promising performance, the model is limited to short answers and demonstrates reduced accuracy due to inadequate training data. Enhancements in dataset size and model generalization for complex answer formats remain critical for broader applicability.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Paper Title** | **Proposed Work** | **Technical Algorithm Used** | **Parameters and Evaluation** | **Data Set** | **Observed Drawback** |
| Automated Evaluation of Handwritten Answer Script Using Deep Learning Approach | A model for evaluating handwritten answer scripts using OCR and deep learning for grading. | Convolutional Neural Network (CNN) and Bidirectional LSTM (BiLSTM) | Training Accuracy: ~90%, Test Accuracy: ~80%; Confusion Matrix, Precision (max: 94%), Recall (max: 88%), F1-score (max: 0.89). | IAM dataset for OCR; student answers from Hamdard University Bangladesh (450 scripts, questions with 35–40 answers, grades assigned: 1–3). | Limited to short answers (40 words); insufficient training data affects accuracy; lacks generalization for longer text with figures and equations. |

The paper *"****Automatic Subjective Answer Evaluation****"* introduces an automated evaluation system for subjective answers using a combination of keyword matching, sentence similarity measures, and grammar checks. It effectively reduces manual effort while achieving grading consistency close to human evaluators. The model relies on NLP techniques like RAKE and BERT and shows promise for scalable online assessments. However, its limitation in handling non-textual content indicates room for further research in extending applicability to diverse response formats.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Paper Title** | **Proposed Work** | **Technical Algorithm Used** | **Parameters and Evaluation** | **Dataset** | **Observed Drawback** |
| Automatic Subjective Answer Evaluation | A model for automated evaluation of subjective answers to reduce human bias, save time, and match instructor grading closely. | Keyword extraction (RAKE, YAKE, TF-IDF), similarity measures (Cosine, Jaccard, BERT), grammar checks (LanguageTool), summarization methods (BM25, BLEU, ROUGE). | Evaluates answers based on keyword match, grammar correctness, similarity score; tested with responses to 20+ questions from 14 students. | Responses from students. | Does not handle non-textual data like graphs, equations, or tables; limited to textual answers; needs enhancement for complex vocabulary and grammar nuances. |

The paper *"****Adopting Computer-Assisted Assessment in Evaluation of Handwritten Answer Books****"* introduces a two-phase framework combining neural network-based handwriting recognition (CNN + LSTM) and semantic similarity evaluation (InferSent) for automatic grading of handwritten answer books. Tested on Grade VII Social Science answers, it achieved promising RMSE values (0.29–0.77). However, its limitations include handling only text-based content and requiring improved accuracy for practical deployment.

| **Paper Title** | **Proposed Work** | **Technical Algorithm Used** | **Parameters and Evaluation** | **Data Set** | **Observed Drawback** |
| --- | --- | --- | --- | --- | --- |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Adopting Computer-Assisted Assessment in Evaluation of Handwritten Answer Books | Two-phase framework: digitization of handwritten answers and evaluation using model answers | Neural network for handwriting recognition (CNN + LSTM), semantic similarity (InferSent), syntactic similarity (Tree Kernel) | RMSE for short answers (0.29), RMSE for long answers (0.77); evaluated on accuracy compared to human grading | Handwritten answer books (Grade VII Social Science, Ranchi, India) | Limited to text-based answers, does not handle diagrams or tables; requires improvement in accuracy and scalability |

The paper titled **"A Bidirectional LSTM Approach for Written Script Auto Evaluation Using Keywords-Based Pattern Matching"** proposes a model that leverages BiLSTM and CRNN to address the limitations of CNN and RNN in handwritten text recognition by overcoming the vanishing gradient issue. With optimized preprocessing and segmentation techniques, the model achieves a CER of 5.81% and WER of 23.55%, outperforming existing methods. However, scalability remains a challenge due to limited model depth and computational resource requirements, suggesting room for future enhancement.

| **Paper Title** | **Proposed Work** | **Technical Algorithm Used** | **Parameters and Evaluation** | **Data Set** | **Observed Drawback** |
| --- | --- | --- | --- | --- | --- |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| A Bidirectional LSTM approach for written script auto evaluation using keywords-based pattern matching | Automated evaluation of handwritten scripts using BiLSTM+CRNN for accuracy in keyword-based grading. | BiLSTM, CRNN, and CTC loss function | Achieved a Character Error Rate (CER) of 5.81%, Word Error Rate (WER) of 23.55%, and Sequence Error Rate (SER) of 35.44%. | IAM, Washington datasets | Limited depth in the model; requires higher memory and processing power for deeper models and diverse datasets. |

**Proposed Work**

The proposed architecture provides a streamlined solution for evaluating handwritten answer scripts by integrating multiple stages of processing and analysis. The system begins with image preprocessing, where the input is enhanced for clarity, followed by efficient line and word extraction techniques. A robust strike-word detection mechanism filters unnecessary content, ensuring clean text for conversion to digital format. The processed text is stored as semantic vectors in a vector database, enabling retrieval of related chunks for comparison with reference answers. Finally, the system employs semantic similarity analysis and weighted word scoring to evaluate answers and generate a final score, ensuring accuracy and fairness. Each module is designed to work cohesively for seamless and reliable automation.

**Architecture Diagram**

A diagram of a computer

Description automatically generated

**System Workflow Description**

**Student Input**

**Handwritten Answer Script Image**:  
This serves as the input layer where the student's handwritten answer script is captured as an image. The input is processed through various subsequent stages to extract, analyze, and evaluate the textual information.

**Task Creator**

**Image Grouping**:  
This module organizes scanned answer sheets into logical groups, ensuring that related images, such as multi-page answers or sections belonging to the same question, are processed collectively. It helps streamline the processing workflow, ensuring no misalignment or mismatch in subsequent stages.

**Reference PDFs:**This component holds the reference material in digital format. These are later compared with the student's answer using semantic similarity techniques. The references are preprocessed and stored for efficient retrieval during evaluation.

**Text Converter**

**Image Preprocessing:**

**Image Resizing**: The input images are resized to a standardized dimension of 2500x3525 pixels, which maintains the aspect ratio of A4 paper while preserving image clarity. This step is critical for normalizing images captured using different cameras, ensuring uniformity and compatibility with downstream processing algorithms. Resizing mitigates inconsistencies in input quality and enhances the reliability of subsequent processing stages.

**Grayscale Conversion**: Converts the image into grayscale to simplify processing while preserving key features.

**Binarization**: Converts the grayscale image into a binary format using a threshold value of 200 for enhanced text segmentation and improved processing accuracy.

**Binary Inversion**: Inverts the binary color scheme to ensure the text stands out clearly from the background.

**Noise Removal**: Eliminates small dots and distortions to enhance text clarity, ensuring accurate Handwritten Text Recognition (HTR).

**Line Extraction**

**Line Merging**: Uses a dilate operation with a kernel size of 1x250 to merge all characters and words in a horizontal line, ensuring continuity within text lines.

**Line Detection**: Detects the position of dilated lines using contour detection, accurately identifying the boundaries of each line.

**Line Extraction**: Applies a logical AND operation between the dilated and detected line images to isolate individual text lines for further word-level processing.

**Word Extraction**:

**Word Merging:** Applies a dilate operation with a kernel size of 8x8 to merge individual characters within each line, forming coherent words based on spatial alignment.

**Word Positioning**: Uses contour detection to accurately identify the position of each word in the line, ensuring the correct order of words.

**Word Extraction**: Extracts each word image by applying a logical AND operation between the dilated image and the word position image, isolating the individual words for text recognition.

**Strike Word Detection**

**Strike Word Model Training**:  
A model, fine-tuned on the google/vit-base-patch16-224 Vision Transformer (ViT), is trained to detect and discard irrelevant or struck-out words in handwritten text images.

**Strikethrough Dataset**:  
The Single-Writer Strikethrough Dataset, containing images of handwritten words with strikethrough annotations, is used to train and validate the model, enabling accurate identification of strikethrough content.

**HTR (Handwritten Text Recognition) Text Conversion**:  
This module uses advanced HTR models, like microsoft/trocr-large-handwritten, to convert extracted handwritten words into digital text. It outputs raw text that undergoes further processing.

**Post-Processing**:

**Whitespace Removal**: Eliminates unnecessary spaces between characters inside words.

**Word Concatenation**: Concatenates each word with white spaces in between to form complete words, resulting in the final digital text.

**Grammar & Spell Check**: Utilizes the SpellChecker library for correcting misspellings and the LanguageTool API for identifying and fixing grammatical errors, ensuring high-quality and readable digital text.

**Knowledge Retriever**

**Preprocessing**:

**Text Cleaning**: Removes unnecessary characters, multiple white spaces, and redundant new lines to produce a clean, structured text.

**Text Chunking**: Segments text into coherent paragraphs to ensure semantically similar content remains within the same chunk for accurate vectorization and analysis.

**Semantic Vector Storage**:

**Vectorization**: Transforms text chunks into vector representations using the Sentence-Transformers/all-MiniLM-L6-v2 model for efficient semantic analysis.

**Chunk Storage**: Stores these vectors in a database for efficient retrieval and comparison.

**Vector Database**: A high-performance vector database like ChromaDB is used to manage and retrieve vectorized chunks based on semantic similarity.

**Answer Synthesizer**:

**Related Chunk Retrieval**: Find the most relevant reference chunks by calculating cosine similarity between the Question vector and database vectors.

**Text Generation Model**: Generates or reformulates responses using the **Google FLAN-T5 Model** for balanced accuracy and fluency.

**Answer Evaluator**

**Semantic model Evaluation:**

**Convert Text to Vectors:** Textual data is transformed into numerical vectors using various sentence transformers (e.g., all-MiniLM-L6-v2) and word embedding models. These vectors capture semantic meaning, enabling precise comparison of sentence-level relationships.

**Question-Answer Repository:** A database of 10 questions was created, each with 4 answers: 2 correct and 2 incorrect. This repository provides a structured dataset for evaluating model performance in distinguishing correct from incorrect answers.

**Select Similarity Metrics:** Metrics like cosine similarity, Word Mover’s Distance (WMD), and soft cosine similarity were used to measure semantic similarity between vectors, ensuring diverse perspectives in evaluating text relationships.

**Compute Similarities**: Pairwise similarities were calculated between correct answer pairs and correct-incorrect pairs. A formula was applied to quantify model performance:

**Score** = ∑ (similarity (correct1, correct2) – average (similarity (correct, incorrect)))

**Evaluate Models:** Models were ranked based on their ability to maximize the separation between correct and incorrect answer similarities. Performance metrics and aggregated scores were used to compare their effectiveness.

**Optimal Model Identification:** The model with the highest score was identified as the optimal solution, demonstrating a superior ability to distinguish semantic nuances between correct and incorrect answers.

**Semantic Similarity Score**:

**Text Vectorization**: Converts both the student's and reference answers into vector embeddings.

**Semantic Similarity Analysis**: Calculates similarity between vectors to measure the semantic equivalence of answers.

**Weighted Word Scoring**:

**Stop Word Removal**: Removes filler words that do not contribute to the meaning or relevance of the answer.

**Word Scoring Model**: Assigns relevance scores to words by calculating their importance in the context of the question and the provided answers. The model uses the following formulas:

**Term Significance (TS):** TS(W) = Number of times word W appears in Answer A /

Total number of words in document D

**Global Relevance (GR):** GR(W) = Number of answers containing word W /

Total number of answers

**Answer Scoring**:

**Score Filtering**: Identifies and eliminates anomalous or outlier scores that deviate significantly from the expected range, ensuring robustness in evaluation metrics and preventing skewed results.

**Score Normalization**: Applies consistent scaling across different scoring methods to maintain fairness and comparability. Techniques such as min-max normalization adjust values within predefined ranges, enhancing uniformity across various metrics.

**Output**

**Final Score**:

The final evaluation result is computed using a weighted aggregation of semantic similarity scores and word-level scoring. This ensures an accurate, fair, and comprehensive assessment of the student's handwritten answer.

**Results:**

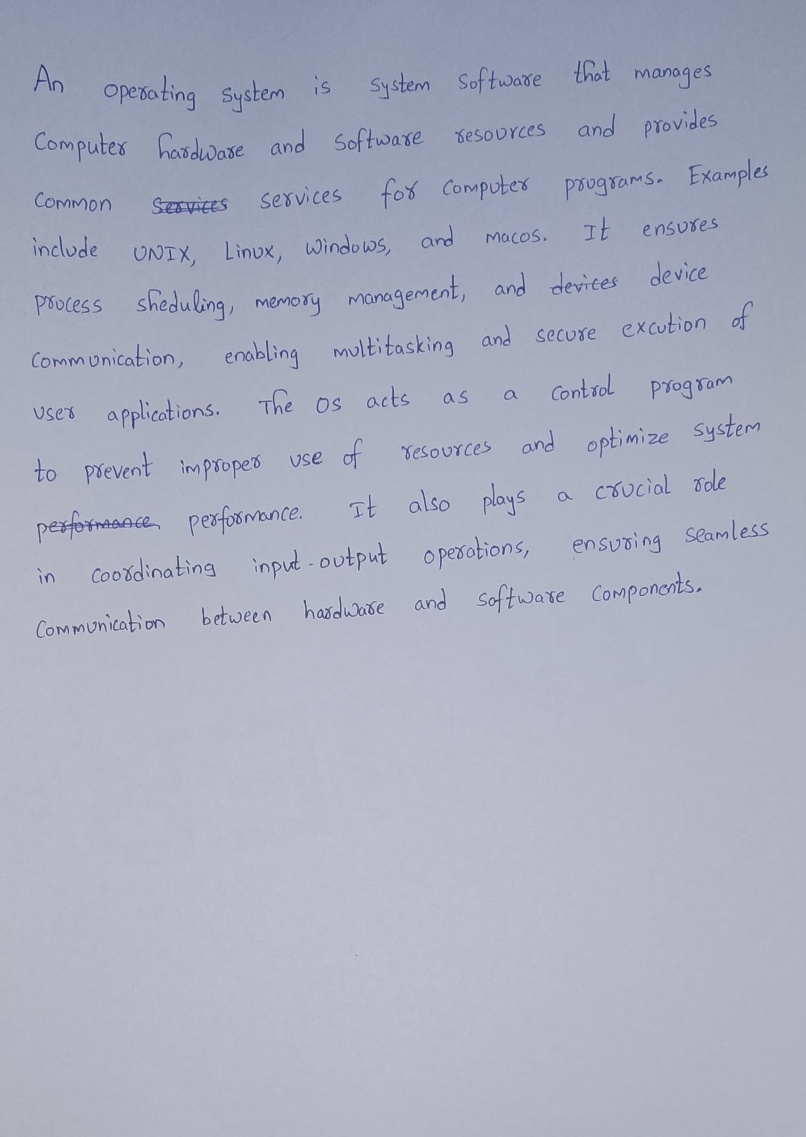
**Strikethrough Detection Using Fine-Tuned Image Classification Model**

Strikethrough detection was performed using the fine-tuned **google/vit-base-patch16-224** model on the **Single-Writer Strikethrough Dataset**. The training process utilized:

* **Loss Function:** CrossEntropy
* **Optimizer:** AdamW
* **Learning Rate:** 2e-4

The model achieved a **validation accuracy of 99.60%**, demonstrating exceptional reliability in identifying strikethrough patterns and ensuring accurate preprocessing for downstream tasks.

**Handwritten Text Recognition**

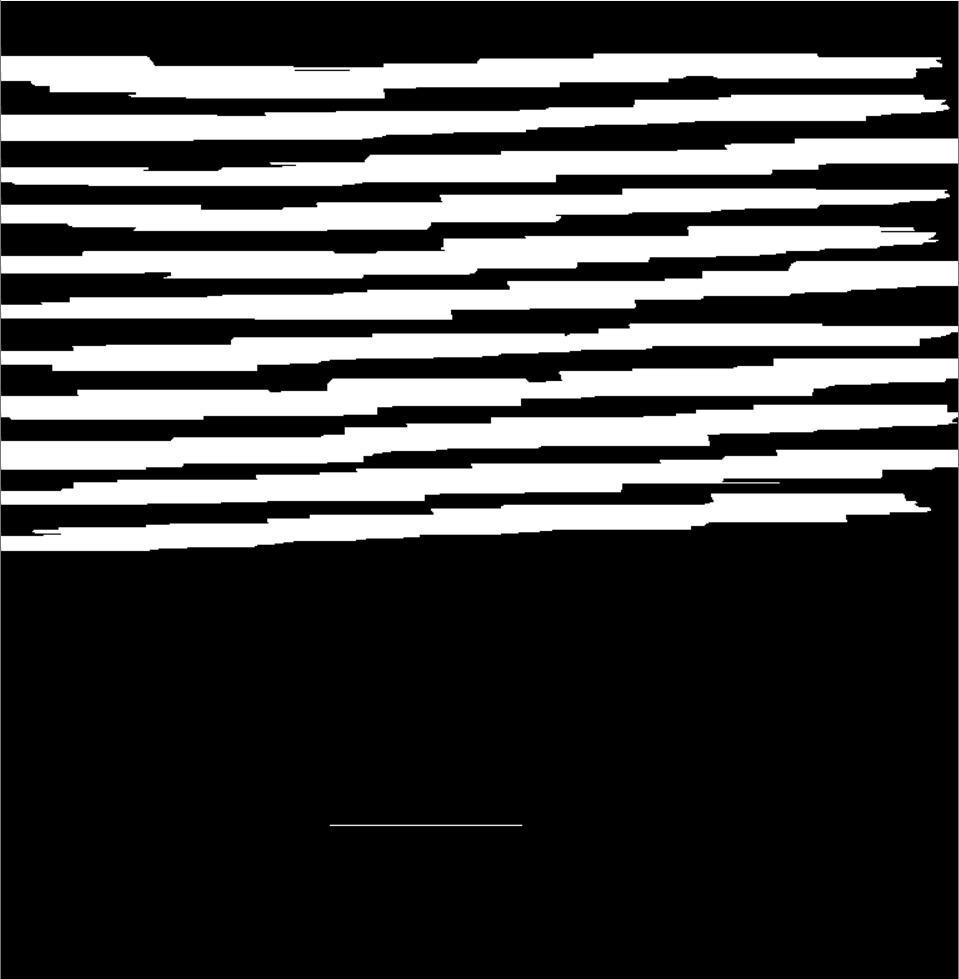
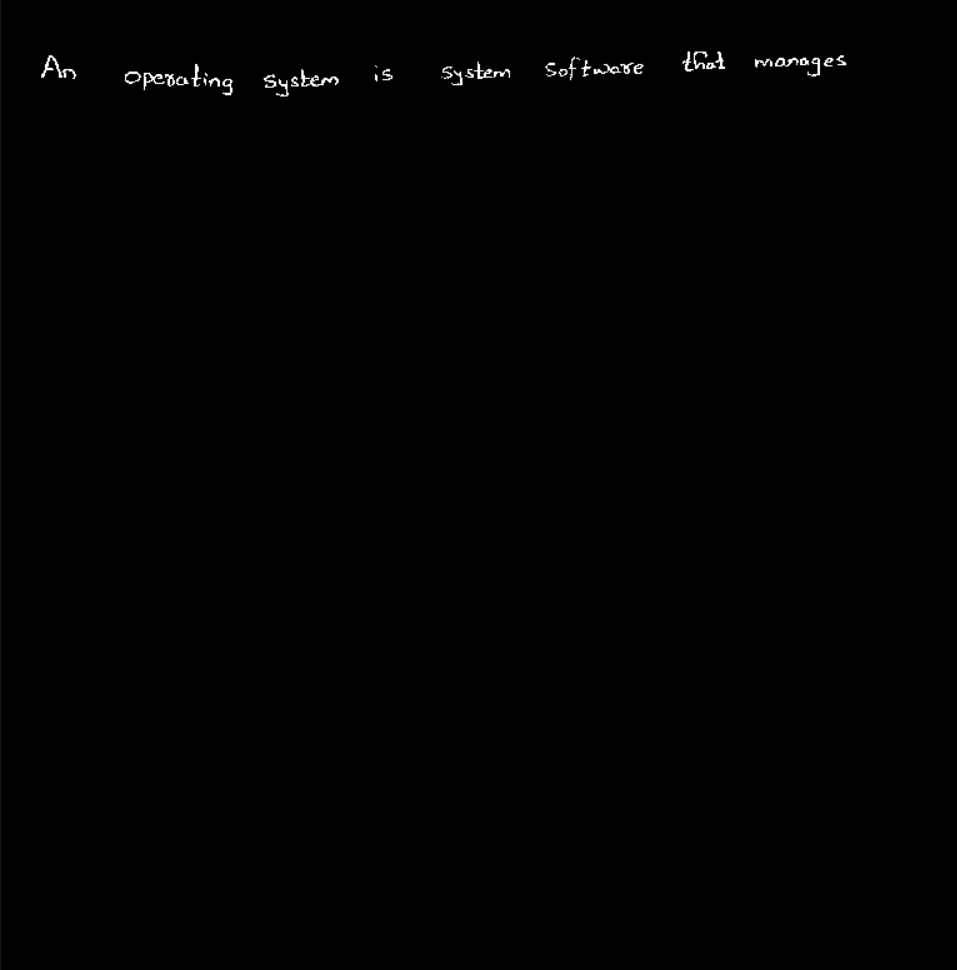
****

Input Image

**Line-Level Handwritten Text Identification**

When the lines in a handwritten image are very close together or overlapping, there is a risk of multiple lines being erroneously identified as a single contour, thus misrepresenting the text structure. To address this, continuous erosion and dilation operations can be applied to remove intersecting pixels. However, these operations can significantly degrade the image's clarity and quality. This issue can be mitigated by mapping the processed image back to the original, preserving the quality while separating the lines.

Nevertheless, the primary challenge lies in determining the optimal threshold for these operations, as it varies depending on the image. Furthermore, this process is computationally expensive, especially on a large scale. An alternative approach is to address cases where lines are close but not touching by vertically scaling the image. Doubling the height of the image while maintaining the original width effectively increases the separation between lines. This approach creates a larger gap between contours, making it easier to distinguish individual lines without compromising image quality.

** **

Line Position Identification in Handwritten Text Using Dilation with 1x250 Kernel Detection of the First Handwritten Line

**Word-Level Handwritten Text Identification**

****

Word-Level Detection in Handwritten Text Following an Organized Reading Order

**Digital Conversion of Handwritten Text**

An operating system. is system. software that manages Computer hardware and software. resources and provides common services for computer. programs. Examples include UNIX, Linux, windows, and macos. It ensures. process sheduling, management, and device Communication, enabling multitasking and secure excution usex applications. the 0s acts as a control program to prevent improper use of resources and optimize system performance. also plays a crucial in coordinating input output operations, ensuring seamless Communication between. hardware. and software. components.

Output of Handwritten Text Recognition Before Post-Processing:

**Post-Processing and Refinement**

An operating system is system software that manages Computer hardware and software resources and provides common services for computer programs Examples include UNIX, Linux, windows and tacos It ensures process scheduling management and device Communication, enabling multitasking and secure execution used applications the is acts as a control program to prevent improper use of resources and optimize system performance also plays a crucial in coordinating input output operations ensuring seamless Communication between hardware and software components

Refined Handwritten Text After Post-Processing

**Semantic Similarity Model Comparison**

Two types of models were utilized for evaluating text similarity:

1. **Sentence Transformer Models:** Converts entire sentences into single vectors, enabling holistic semantic comparison.
2. **Word Embedding Models:** Represents each word as a vector, facilitating granular word-level similarity analysis.

**Similarity Metrics**

The following techniques were employed for vector comparisons:

* **Cosine Similarity:** Computes the cosine of the angle between two vectors, indicating their directional similarity.
* **Word Mover's Distance (WM Distance):** Measures the semantic distance between two word embeddings by calculating the minimal "cost" required to transform one into the other.
* **Soft Cosine Similarity:** Extends cosine similarity by considering feature-level relationships, such as inter-word semantic similarities.

**Implementation**

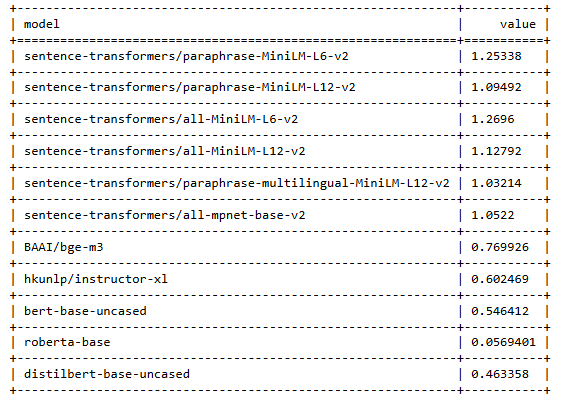
* **Sentence Transformers:** Cosine similarity was used for direct sentence-to-sentence vector comparison.
* **Word Embedding Models:** All three metrics—Cosine Similarity, WM Distance, and Soft Cosine Similarity—were applied, leveraging the word-level vector representations.

**Evaluation Setup**

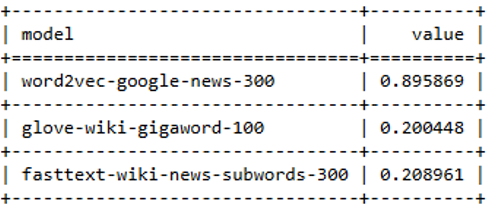
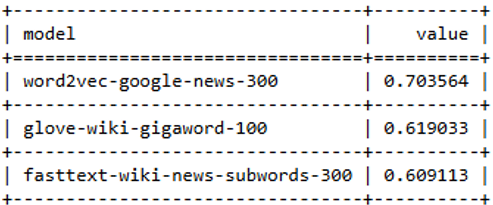
* **Dataset:** Ten questions, each with four answers: two correct and two incorrect.
* **Scoring Formula:** ∑ (similarity (correct1​,correct2​) – average (similarity (correct, incorrect)) )

This scoring approach quantifies the distinction between correct and incorrect answers by emphasizing intra-correct similarity while penalizing overlap with incorrect answers.

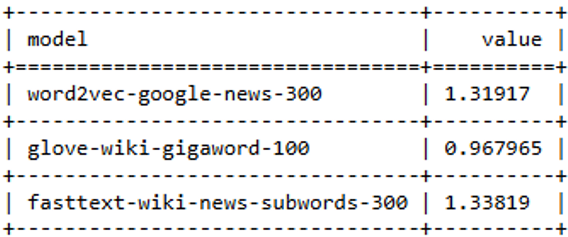
**Results:**



**Similarity score** using **Sentence Transformers** models and **cosine similarity**

**Similarity score** using **word embedding** models and **cosine similarity Similarity score** using **word embedding** models and **WM distance**



**Similarity score** using **word embedding** models and **Soft Cosine Similarity**

**Evaluation (Hybrid Scoring)**

The total score was evaluated out of 10 marks by combining results from multiple scoring mechanisms: TS-GR scoring, Sentence-Transformer scoring, and Word Embedding Model scoring. The evaluation framework is detailed as follows:

1. **TS-GR Score:**  
   The TS-GR score was calculated as:

TS−GR Score = (TS-GR value of answer minimum) / (TS-GR of correct answers) × 5

To ensure uniformity, the TS-GR score was capped at a maximum value of 5.

1. **Sentence-Transformer Score:**  
   Using the sentence-transformers/all-MiniLM-L6-v2 model, cosine similarity was computed for each answer. The scoring mechanism applied was:

If cosine similarity > 0.95: Score = 3

If cosine similarity < 0.5: Score = 0

Otherwise, scores were normalized using Min-Max Normalization with parameters: old min=0.5, old max=0.95, new min=0, new max=3

1. **Word Embedding Model Score:**  
   The fasttext-wiki-news-subwords-300 model was employed for its ability to handle out-of-vocabulary words. The scoring was based on Soft Cosine Similarity as follows:

If similarity > 0.9: Score = 3

If similarity < 0.4: Score = 0

Otherwise, scores were normalized using Min-Max Normalization with parameters: old min=0.4, old max=0.9, new min=0, new max=2

**Final Score Calculation:**  
The hybrid scoring system effectively integrates normalized values from all three methods, ensuring a balanced evaluation of semantic, contextual, and structural aspects of answers. This approach provides a robust and reliable measure of the answer quality, with a total score capped at 10.  
  
**output**

“An operating system is a collection of system programs that together control the operations of a computer system. Some examples of operating systems are UNIX, Mach, MS-DOS, MS-Windows, WindowsNT, Chicago, OS2, MacOS, VMS, MVS, and VM System View: From the computer's point of view, an operating system is a control program that manages the execution of user programs to prevent errors and improper use of the computer. It is concerned with the operation and control of IO devices.”

Knowledge Retriever: Answer 1

"The operating system sits between the user and the hardware of the computer providing an operational environment to the users and application programs. For a user, therefore, a computer is nothing but the operating system running on it. It is an extended machine Operating System (or briefly OS) provides services to both the users and to the programs 1. It provides programs, an environment to execute OS is a resource allocate Manages all resources Decides between conflicting requests for efficient and fair resource use OS is a control program Controls execution of programs to prevent errors and improper use of the computer."

Knowledge Retriever: Answer 2

"An operating system is system software that manages Computer hardware and software resources and provides common services for computer programs Examples include UNIX, Linux, windows and tacos It ensures process scheduling management and device Communication, enabling multitasking and secure execution used applications the is acts as a control program to prevent improper use of resources and optimize system performance also plays a crucial in coordinating input output operations ensuring seamless Communication between hardware and software components"

Answer-1

Marks are calculated using the TS-GR Score = 4.4420 marked out of 5

Marks are calculated using the Sentence-Transformer Score 2.7562 marked out of 3

Marks are calculated using the Word Embedding Model Score 2 marked out of 2

**Total marks = 9.1983**

"An operating system is a collection of software that manages hardware and software resources in a computer. It provides a platform for applications to run and coordinates processes, memory, and storage. Examples include UNIX, Windows, and macOS, which enable multitasking and secure user operations. By controlling hardware like I/O devices and ensuring the efficient execution of programs, the OS acts as an intermediary between users and the machine. It plays a crucial role in error prevention and system reliability, allowing for smooth and efficient use of computing resources."

Answer-2

Marks are calculated using the TS-GR Score = 4.5260 marked out of 5

Marks are calculated using the Sentence-Transformer Score 2.7244 marked out of 3

Marks are calculated using the Word Embedding Model Score 2 marked out of 2

**Total marks = 9.2504**

"An operating system is a program designed to load applications and manage the appearance of the desktop environment. It works primarily to enhance user experience by controlling graphical features like icons, themes, and window arrangements. For instance, operating systems such as iOS and Android mainly function as user interfaces for smartphones. They simplify accessing apps but do not handle complex operations like process management or security. The OS focuses on ensuring applications run visually appealingly, leaving performance optimization to hardware."

Answer-3

Marks are calculated using the TS-GR Score = 3.7577 marked out of 5

Marks are calculated using the Sentence-Transformer Score 2.3749 marked out of 3

Marks are calculated using the Word Embedding Model Score 1.2524 marked out of 2

**Total marks = 7.3851**

"An operating system is mainly used to open and close applications on a device. It helps users interact with the graphical interface and ensures that programs can load. While it is often confused with hardware, the OS’s primary role is to make computers visually appealing and user-friendly. For example, mobile operating systems like iOS focus on apps and user convenience but don’t manage lower-level tasks like memory or processor allocation. Without an OS, applications wouldn’t look the same or respond as quickly, but hardware would still operate independently."

Answer-4

Marks are calculated using the TS-GR Score = 3.0699 marked out of 5

Marks are calculated using the Sentence-Transformer Score 1.9311 marked out of 3

Marks are calculated using the Word Embedding Model Score 0.9950 marked out of 2

**Total marks = 5.9961**

"An operating system is a hardware component inside the computer that generates electricity to power other parts. It functions like a battery, storing energy and distributing it when needed. Examples of operating systems include the RAM, the power supply unit, and the motherboard, all of which are critical for maintaining a computer's energy flow. Without an OS, the computer would not turn on because it would lack the electrical charge necessary to power the system."

Answer-5

Marks are calculated using the TS-GR Score = 1.1577 marked out of 5

Marks are calculated using the Sentence-Transformer Score 2.1725 marked out of 3

Marks are calculated using the Word Embedding Model Score 1.2120 marked out of 2

**Total marks = 4.5423**

"An operating system is a physical object that helps connect computers to the internet by capturing Wi-Fi signals. It is stored inside the monitor and works by converting light signals into data that the computer can process. Examples of operating systems include routers, Bluetooth devices, and USB drives. Without an operating system, computers wouldn’t be able to browse websites or use social media. The OS also powers the screen by generating electricity and controls how bright or dim the display appears."

Answer-6

Marks are calculated using the TS-GR Score = 1.5343 marked out of 5

Marks are calculated using the Sentence-Transformer Score 2.1053 marked out of 3

Marks are calculated using the Word Embedding Model Score 1.1673 marked out of 2

**Total marks = 4.8070**

**Conclusion**

This paper represents a significant milestone in the automation of handwritten script evaluation, addressing critical challenges in educational assessment through an innovative integration of advanced technologies. The proposed system seamlessly combines Handwritten Text Recognition (HTR), natural language processing (NLP), and retrieval-augmented generation (RAG) techniques to deliver a comprehensive solution for evaluating handwritten responses. The system effectively measures content relevance, accuracy, and semantic similarity by digitizing handwritten text, retrieving contextually relevant reference materials from study content, and employing a hybrid scoring mechanism.

The project demonstrated the efficacy of combining traditional methods such as TS-GR with modern transformer-based language models like Sentence-Transformers and FLAN-T5. These models provide nuanced assessments of semantic meaning, ensuring that the system accurately captures exact matches and contextual alignments between student responses and reference answers. Furthermore, the innovative preprocessing pipeline, including strikethrough detection and correction, enhances the system's robustness against common handwriting artifacts.

Key contributions of this work include its scalability for large-scale evaluations, suitability for remote learning scenarios, and ability to maintain objectivity and fairness in assessments. These features address pressing issues faced by educators, particularly the increasing demand for efficient, consistent, and unbiased evaluation methods in digital and hybrid learning environments.

Despite its success, the project also opens several avenues for future research and improvement. Enhancing the semantic understanding capabilities of the scoring algorithm can further refine the evaluation process, particularly for complex or multi-part questions. Expanding the system to support multilingual and multimodal input, such as diagrams or mathematical notations, will broaden its applicability to diverse educational contexts. Additionally, optimizing computational efficiency will enable real-time processing, making the system more accessible for large-scale deployment in resource-constrained settings.

This work not only lays the groundwork for a robust AI-driven evaluation framework but also underscores the potential of leveraging machine learning and AI to transform traditional educational paradigms. By addressing current limitations and advancing the system’s capabilities, this project aims to contribute significantly to the future of automated educational assessments, setting a benchmark for innovation in the field.

**References**

1. S. R. Zanwar, et al., "Handwritten English character recognition using swarm intelligence and neural network," *Intelligent Systems, Technologies and Applications: Proceedings of Fifth ISTA 2019, India*, Singapore: Springer Singapore, 2020, pp. 93-102.
2. E. Vidal, et al., "End-to-end page-level assessment of handwritten text recognition," *Pattern Recognition*, vol. 142, 2023, Art. no. 109695.
3. R. Chinthaginjala, et al., "Enhancing handwritten text recognition accuracy with gated mechanisms," *Scientific Reports*, vol. 14, no. 1, 2024, Art. no. 16800.
4. V. K. Chauhan, S. Singh, and A. Sharma, "HCR-Net: A deep learning based script independent handwritten character recognition network," *Multimedia Tools and Applications*, 2024, pp. 1-35.
5. Y. Dong, et al., "Advanced RAG Models with Graph Structures: Optimizing Complex Knowledge Reasoning and Text Generation," *arXiv preprint arXiv:2411.03572*, 2024.
6. Z. Jiang, X. Ma, and W. Chen, "Longrag: Enhancing retrieval-augmented generation with long-context LLMs," *arXiv preprint arXiv:2406.15319*, 2024.
7. J. S. Pangestu, et al., "Implementation of Handwriting Recognition and Answer Evaluation with Recurrent Neural Network," *Procedia Computer Science*, vol. 227, 2023, pp. 779-784.
8. M. F. Bashir, et al., "Subjective answers evaluation using machine learning and natural language processing," *IEEE Access*, vol. 9, 2021, pp. 158972-158983.
9. M. A. Rahaman and H. Mahmud, "Automated evaluation of handwritten answer script using deep learning approach," *Transactions on Machine Learning and Artificial Intelligence*, vol. 10, no. 4, 2022.
10. V. Kumari, P. Godbole, and Y. Sharma, "Automatic Subjective Answer Evaluation," *ICPRAM*, 2023.
11. S. K. Saha and R. Gupta, "Adopting computer-assisted assessment in evaluation of handwritten answer books: An experimental study," *Education and Information Technologies*, vol. 25, no. 6, 2020, pp. 4845-4860.
12. N. Prabakaran, et al., "A Bidirectional LSTM approach for written script auto evaluation using keywords-based pattern matching," *Natural Language Processing Journal*, vol. 5, 2023, Art. no. 100033.
13. P. Bhagyasree, A. James, and S. Chandran, "A proposed framework for recognition of handwritten cursive English characters using DAG-CNN," pp. 1–4, Apr. 2019.
14. D. Alikaniotis, H. Yannakoudakis, and M. Rei, "Automatic text scoring using neural networks," *arXiv preprint* arXiv:1606.04289, 2016.
15. V. Rowtula, S. R. Oota, and C. V. Jawahar, "Towards automated evaluation of handwritten assessments," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, 2019, pp. 426–433.