Generative Artificial Intelligence Project– CS787 "Automated Subjective Answer Grader"

Sahib J. Parmar (251110062)
Boda Meekumar M. (251110050)
Department of Computer Science, IIT Kanpur

Abstract

Automated assessment of subjective answers remains a challenging problem due to the diversity of writing styles, ambiguity in student expressions, and the tendency of large language models (LLMs) to hallucinate or overlook rubric-specific details. To address these challenges, we develop a hybrid multi-model grading framework that combines the strengths of generative LLMs, extractive QA models, and semantic embedding models within a unified teacher-in-the-loop workflow. The system first generates a structured grading rubric using an LLM and then employs three alternative engines: GPT-OSS-120B, DeBERTa-v3 QA, and MPNet sentence embeddings, to extract rubric-aligned answer segments from the student's response. These extracted segments are subsequently evaluated by the LLM to produce tentative score suggestions, which the instructor can review, edit, or override through an interactive Streamlit-based interface. The framework also supports automated rubric refinement and dynamic highlighting of relevant answer spans, enabling transparent and explainable grading. Experiments further compare the behavior of the different extraction engines, while fine-tuning attempts on DeBERTa-like models highlight the limitations of purely model-centric approaches. Overall, the proposed system balances automation with pedagogical reliability, offering a practical and extensible solution for subjective answer grading.

1 Introduction

Subjective answer grading is difficult because students express ideas with varied structure, vocabulary, and reasoning depth. Although modern large language models (LLMs) appear capable of directly scoring long answers, prompting them with "Grade this out of X marks" produces evaluations that are opaque and often unreliable. The model may hallucinate reasoning, miss essential rubric points, or exhibit length bias, making one-shot grading unsuitable for academic use.

To address these limitations, we design a rubric-driven grading framework that breaks evaluation into interpretable steps: rubric generation, answer segmentation, AI scoring, and instructor verification. Instead of forcing a single model to judge the entire answer, the system allows the user to select one of several specialized extraction engines: GPT-OSS 120B, DeBERTa-v3 QA, or MPNet embeddings, each acting as an independent endpoint for mapping the student's answer to rubric criteria. Only the chosen model processes the segmentation stage, ensuring modularity and clear behavior at the prototype level.

This approach removes hidden reasoning, reduces hallucination, and provides explicit evidence for each awarded mark. A teacher-in-the-loop interface further ensures transparency by allowing instructors to inspect extracted segments, adjust scores, and revise the rubric when needed. Overall, this design offers more reliable and explainable grading compared to direct LLM scoring while remaining flexible across diverse question types.

2 System Overview and Architecture

The proposed system follows a modular, rubric-driven pipeline designed to make subjective answer grading transparent, explainable, and controllable. Instead of asking a single model to grade an entire answer directly, the system decomposes the task into distinct stages, each handled by a specialized component. This design also allows the instructor to select which model performs the answer-segmentation step, enabling flexible experimentation and controlled comparison across multiple model behaviors.

2.1 Architecture Diagram

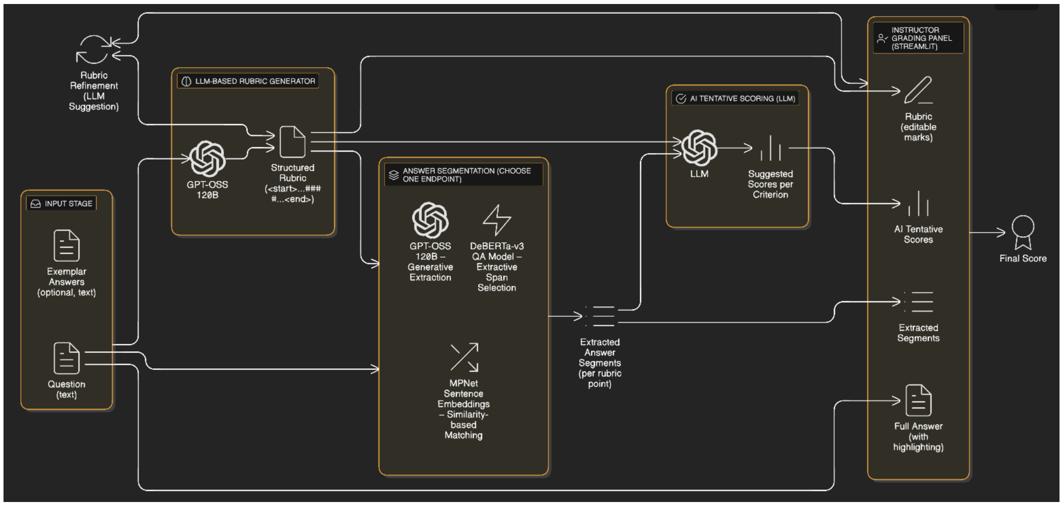
Rubric Refinement (LLM Suggestion)
LLM-BASED RUBRIC GENERATOR
GPT-OSS 120B
Structured Rubric (<start>...### #...<end>)
INPUT STAGE
Exemplar Answers (optional, text)
Question (text)
ANSWER SEGMENTATION (CHOOSE ONE ENDPOINT)
GPT-OSS 120B – Generative Extraction
DeBERTa-v3 QA Model – Extractive Span Selection
MPNet Sentence Embeddings – Similarity-based Matching
Extracted Answer Segments (per rubric point)
AI TENTATIVE SCORING (LLM)
LLM
Suggested Scores per Criterion
INSTRUCTOR GRADING PANEL (STREAMLIT)
Rubric (editable marks)
AI Tentative Scores
Extracted Segments
Full Answer (with highlighting)
Final Score

Figure 1: Overall architecture of the hybrid multi-model subjective answer grading system.

2.2 Rubric Generation

Given the question (and optional exemplar answers), a large language model generates a structured rubric in a strict `<start> ...` `...` `<end>` format. Each rubric point includes a criterion name and its allocated marks, forming a consistent scoring template for all subsequent stages.

2.3 Answer Segmentation (Selectable Endpoint)

The instructor selects one of three independent engines to map the student's answer to rubric criteria:

- **GPT-OSS 120B (Groq):** generative extraction via prompt-based reasoning.
- **DeBERTa-v3 QA:** extractive span selection using a question-answering model.
- **MPNet Sentence Embeddings:** similarity-based assignment of sentences to rubric points.

Only the chosen endpoint executes for a given session. This modularity allows clear observation of how different models behave on the same task without complicating the pipeline.

2.4 AI Tentative Scoring

Using the rubric and extracted answer segments, an LLM assigns tentative marks for each criterion. The score for each point is paired with the exact evidence extracted in the previous step, ensuring interpretability and reducing hidden reasoning.

2.5 Human-in-the-Loop Grading Interface

A Streamlit-based interface displays the full student answer, the rubric, extracted segments, and suggested scores. The instructor can:

- inspect the extracted evidence,
- modify marks,
- edit rubric points,
- toggle highlighting on the student's answer,

ensuring that the final grade remains fully transparent and under instructor control.

2.6 Rubric Refinement

The system optionally suggests minimal modifications to the rubric based on the student's answer, such as introducing a missing criterion or adjusting an existing one. The instructor retains complete authority over whether to apply these changes.

2.7 Workflow Summary

1. Input question and exemplar answers (optional) → *Generate rubric*
2. Provide student answer → Select endpoint → *Extract relevant segments*
3. Extracted segments + rubric → *LLM tentative scoring*
4. Instructor reviews and edits scores / rubric
5. Final score is produced.

3 Methodology

The system decomposes subjective answer grading into five interpretable stages: rubric generation, answer segmentation, AI scoring, instructor-led validation, and optional rubric refinement. Each component is modular and can be independently replaced or extended. This section summarizes the core methodology and briefly describes the models used at each stage.

3.1 Rubric Generation Module

Given a question and optional exemplar answers, a large language model (GPT-OSS 120B running on Groq) generates a structured rubric using a strict `<start> ... ... <end>` format. GPT-OSS 120B is an instruction-following generative model optimized for high-throughput inference, making it suitable for consistent rubric production. The generated rubric defines the evaluation criteria and maximum marks per point, serving as the foundation for all downstream steps.

3.2 Answer Segmentation Module (Selectable Endpoint)

The instructor selects one of three alternative engines to map student answer content to rubric criteria. Only the chosen endpoint runs for that session.

3.2.1 GPT-OSS 120B [3](Generative Extraction)

A prompt-based approach where the LLM identifies verbatim excerpts that correspond to each rubric point. This method is flexible and context-sensitive but prone to occasional hallucination.

3.2.2 DeBERTa-v3[1] QA Model (Extractive Span Selection)

A pretrained question-answering model, `deepset/deberta-v3-large-squad2`, is used to extract spans by treating each rubric point as a "question" and the student answer as "context". DeBERTa's disentangled attention mechanism enables precise token-level extraction, reducing paraphrasing and ensuring that only text present in the answer is returned.

3.2.3 MPNet Sentence Embeddings (Similarity-Based Matching)

The student's answer is sentence-tokenized and embedded using `all-mpnet-base-v2`[4]. For each rubric point, the top-k most semantically similar sentences are selected using cosine similarity, with an optional constraint ensuring each sentence is used at most once. This approach is deterministic, lightweight, and fast, making it suitable for longer answers.

3.3 AI Tentative Scoring

The rubric and extracted answer segments are passed to GPT-OSS 120B, which assigns tentative marks for each rubric point. Scores are produced in a structured format and paired with the exact text evidence extracted earlier, maintaining transparency and preventing hidden reasoning.

3.4 Human-in-the-Loop Grading Interface

A Streamlit-based interface integrates all components into an instructor-friendly workflow. It displays:

- the full student answer with optional highlighting,
- the rubric and its allocated marks,
- extracted segments returned by the selected endpoint,
- AI-suggested tentative scores.

Instructors may override any score, edit rubric points, or reject the AI's initial suggestions. This step ensures the final grade remains aligned with academic judgement rather than fully automated decisions.

3.5 Rubric Refinement Module

To handle unexpected but correct reasoning from students, the system can analyze the answer and suggest minimal rubric updates (e.g., adding a missing criterion). The instructor can choose whether to accept these modifications.

3.6 Highlighting Engine

For interpretability, the system highlights the extracted answer segment inside the full student response. This provides an evidence-based justification for each awarded mark and enables quick verification by instructors. Instructor can also make a look for nearby sentences in case AI missed something, again ensuring fairness.

4 Experiments

4.1 Initial Experiments and Motivation for Pivot

Before developing the hybrid system, we first attempted to reproduce the results shown in paper-[2] by fine-tuning Qwen-2.5[5] 3B (4-bit quantized) on subsets of the ASAP–SAS datasets. Although the model could be fine-tuned, the results were unstable and inconsistent. The quantized 3B model frequently generated irrelevant or low-quality outputs during inference. Even after fine-tuning, the model tended to collapse to mid-range scores(for example, in one experimentation, model assigned average score of training data to all the answers during inference) and failed to generalize across question sets. Also smaller models like Qwen-2.5 3B have less intelligence to assign transperanr grades to the "Subjective" answers. More importantly, this approach could only grade essays it had been trained on and provided no transparency or rubric alignment. These limitations made the SAS-style fine-tuning path unsuitable for subjective short-answer grading, motivating the shift toward a transparent, rubric-driven hybrid system.

4.2 Experiments on Answer Segmentation Models

A significant portion of our experimentation focused on the answer-segmentation stage, as this component determines how reliably the system can map portions of a student's answer to individual rubric criteria. To reduce dependence on large LLMs for this step and explore cost-efficient alternatives, we evaluated several smaller models capable of running inference entirely on CPU. These included RoBERTa-based QA models, encoder-only architectures, and multiple variants of DeBERTa.

(Teammate's detailed observations will be added here.)
This space is reserved for summarizing the fine-tuning attempts on DeBERTa, RoBERTa, and related models, including performance behaviour, comparative results, and any quantitative or qualitative findings obtained during those experiments.

Despite multiple fine-tuning attempts, the best overall performance among compact extractive models came from the pretrained `deberta-v3-large-squad2` model (approximately 400M parameters). Without any additional tuning, it consistently produced the most accurate span extractions in our tests and ran efficiently on CPU, making it a strong low-cost endpoint for answer segmentation.

However, DeBERTa's QA-based formulation inherently restricts it to extracting a single contiguous span per rubric point. When the relevant content is distributed across different parts of the student's answer, the model typically retrieves only a partial match. This limitation motivated the inclusion of two complementary endpoints:

- **GPT-OSS 120B**, which can synthesize evidence drawn from multiple regions of the answer through generative reasoning, and
- **MPNet sentence embeddings**, which compare each rubric point against every sentence in the answer, enabling multi-sentence evidence retrieval.

These observations guided the final design choice of exposing all three models, GPT-OSS, DeBERTa-v3, and MPNet as selectable segmentation endpoints. This provides instructors with the flexibility to choose the most appropriate extraction mechanism depending on the question type, answer length, and available computational resources.

Model	Pros	Cons
GPT-OSS 120B	High semantic understanding; can combine scattered information.	API cost; may paraphrase instead of copying exactly.
DeBERTa-v3 QA	Precise extractive spans; runs on CPU; no hallucination.	Returns only one contiguous span; struggles with distributed evidence.
MPNet Embeddings	Local, fast, and consistent; captures multiple relevant sentences.	Can include loosely related sentences; Retrieval of same sentence in more than one rubric part

Table 1: comparison of segmentation endpoints.

5 Datasets

In order to build a robust Automated Subjective Answer Grader (ASAG), we fine-tuned our models on three large-scale Question Answering datasets: **SQuAD 2.0**, **HotpotQA**, and **MASH-QA**. These datasets were chosen strategically because each of them strengthens a different capability required for reliable and interpretable subjective answer scoring.

5.1 SQuAD 2.0

SQuAD 2.0 is a widely-used extractive question answering dataset consisting of over 100,000 answerable questions along with an additional 50,000 unanswerable ones. The central objective of

SQuAD 2.0 is to train models to accurately identify answer spans within a reference passage, while also learning to abstain when the answer does not exist in the provided context. This dual nature—span prediction and "no-answer" identification—is particularly valuable for our grading pipeline, as subjective answers often require detecting the presence or absence of specific key ideas. Fine-tuning on SQuAD 2.0 enables the model to precisely locate rubric-relevant concepts within a student's response and to identify missing information, which is essential for evidence-based and phrase-level scoring.

5.2 HotpotQA

HotpotQA is a multi-hop reasoning dataset containing over 112,000 questions that require reasoning over multiple supporting paragraphs. Unlike single-span extractive tasks, HotpotQA demands that the model integrate information from different portions of the context, construct logical connections, and identify the supporting facts necessary to justify an answer. These skills directly align with the nature of subjective answers, which typically span several sentences and require coherent reasoning, explanation, and the integration of multiple concepts. Training on HotpotQA enhances the model's ability to evaluate the structure and depth of a student's explanation while also improving performance on long-context inputs, especially when using architectures such as BigBird.

5.3 MASH-QA

MASH-QA is a multi-span, multi-evidence dataset originally developed for medical question answering. Unlike datasets that focus on single contiguous answer spans, MASH-QA frequently requires the model to extract several non-adjacent pieces of information from long documents. This property is highly beneficial for rubric-based grading, where a student's score often depends on their inclusion of multiple key points that may be scattered throughout the response. Fine-tuning on MASH-QA trains the model to detect multiple relevant fragments, thereby supporting partial-credit scoring and multi-point rubric alignment.

5.4 Rationale for Combining the Three

These datasets together provide a complementary set of capabilities crucial for subjective answer grading. SQuAD 2.0 contributes precise span extraction and missing-information detection; HotpotQA injects strong multi-hop reasoning and long-context understanding; and MASH-QA enables multi-span extraction required for evaluating answers against multi-point rubrics.

6 Fine-tuning and Results

Once the initial pipeline of the system was ready, we decided to fine tune **DeBERTa-v3-large** (used in system) and also **BigBird-RoBERTa-large** on above SQuAD2.0, HotpotQA and MASHQA to improve the performance of these models for obtaining best answer spans possible.

6.1 DeBERTa-v3-large

DeBERTa-v3-large is part of the "Decoding-enhanced BERT with disentangled attention" family of models, representing one of the most advanced encoder-only Transformer architectures. Unlike BERT or RoBERTa, DeBERTa introduces two key innovations: *disentangled attention* and *enhanced mask decoder*. Disentangled attention separates the content and positional information of tokens, allowing the model to attend to semantic meaning and positional structure independently. This leads to more expressive attention patterns and stronger contextual understanding. The v3 version incorporates the *ELECTRA-style replaced token detection* pretraining objective, making it more sample-efficient and enabling significantly better generalization with fewer pretraining steps. With 24 layers and roughly 435 million parameters, DeBERTa-v3-large achieves state-of-the-art performance on many natural language understanding tasks. For our grading pipeline, its ability to capture fine-grained semantics and precise span-level details makes it particularly effective for rubric alignment, keyphrase detection, and evidence-based scoring.

6.2 BigBird-RoBERTa-large

BigBird-RoBERTa-large is a long-sequence Transformer model derived from RoBERTa but equipped with BigBird's sparse attention mechanism. Traditional Transformers exhibit quadratic complexity in sequence length, limiting practical usage to sequences of around 512 tokens. BigBird overcomes this by combining three forms of attention—global, windowed (local), and random—thereby achieving linear complexity while preserving theoretical guarantees of universal approximability and Turing completeness. BigBird-RoBERTa-large can process inputs up to 4,096 tokens efficiently, making it highly suitable for tasks that involve long documents, multi-paragraph reasoning, or full-length subjective answers. With approximately 355 million parameters, it retains the strong language understanding capabilities of RoBERTa while enabling long-context modeling. In our ASAG system, BigBird-RoBERTa-large is especially valuable for evaluating long student responses, performing multi-hop reasoning, and integrating information across several sentences to assess overall explanation quality.

References

[1] Pengcheng He, Jianfeng Gao, and Weizhu Chen. Debertav3: Improving deberta using electra-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021.

[2] Ahmed Karim, Qiao Wang, and Zheng Yuan. Beyond the score: Uncertainty-calibrated llms for automated essay assessment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 19642–19647, 2025.

[3] OpenAI. gpt-oss-120b gpt-oss-20b model card, 2025.

[4] Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. Mpnet: Masked and permuted pre-training for language understanding. *arXiv preprint arXiv:2004.09297*, 2020.

[5] Qwen Team. Qwen2.5: A party of foundation models, September 2024.