EduGrade: Retrieval-Augmented Short-Answer Grading System

Sai Sri Harsha Chunduri
(G01514410)
schundu2@gmu.edu

Rutuja More
(G01511407)
rmore@gmu.edu

Shreyas Ajgaonkar
(G01520908)
sajgaon@gmu.edu

1 Recap: Brief Project Description

The project **EduGrade** is a retrieval-augmented pipeline for automated short-answer grading. Traditional supervised grading systems depend on fixed human-written reference answers, which limits generalization and prevents adaptation to unseen questions. EduGrade addresses this limitation by integrating retrieval-augmented generation (RAG) with a RoBERTa-based classifier, enabling the system to dynamically construct teacher-like reference answers before grading student responses.

The full pipeline consists of three components:
(1) A retrieval module using MPNet embeddings and a FAISS index to locate relevant scientific passages.
(2) A GPT-4o-mini generator that produces context-grounded reference answers.
(3) A RoBERTa-large classifier trained on SciEntsBank to label student answers as *correct*, *contradictory*, or *incorrect*.

The project evaluates the system under three complementary settings: the baseline grader, the full RAG pipeline, and an intrinsic quality analysis of generated answers.

EduGrade reframes short-answer grading by grounding evaluation in domain knowledge rather than surface text. It constructs instructional reference answers from retrieved science evidence and uses these to support RoBERTa-based judgment of conceptual correctness. This approach strengthens generalization to unseen questions and improves semantic alignment during grading.

2 Detailed System or Methodology Description

EduGrade is a modular system composed of independent retrieval, generation, and grading components. This modularity allows each part to be evaluated separately and jointly, as illustrated in the Figure 1 pipeline diagram.

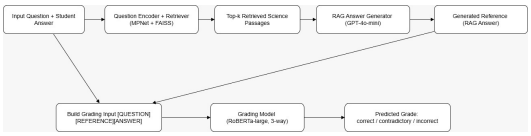

Figure 1: EduGrade system architecture illustrating retrieval, reference generation, and grading flow.

The pipeline diagram illustrates EduGrade's end-to-end grading workflow, beginning with a student question–answer pair and progressing through retrieval, reference generation, and semantic evaluation stages. First, the question is encoded using MPNet and passed to a FAISS index to retrieve the top-k science passages most relevant to the concept being assessed. These retrieved passages supply factual grounding for the next stage, where a GPT-4o-mini model synthesizes a teacher-style explanatory reference answer. The original question, generated reference, and student response are then concatenated into a structured input template; [QUESTION][REFERENCE][ANSWER], enabling comparative reasoning rather than surface-form matching. This enriched input is evaluated by a RoBERTa-large classifier fine-tuned on SciEntsBank labels, which predicts one of three outcomes: correct, incorrect, or contradictory. Structuring the system this way mirrors authentic instructional practice, where teachers first recall domain knowledge, articulate an ideal explanation, and then compare student reasoning against it. The modular nature of this architecture supports interpretability, facilitates error diagnosis across individual components, and allows each stage, retrieval, generation, and grading to be independently optimized or replaced without redesigning the entire system. This design

enables EduGrade to generalize beyond keyword matching and evaluate conceptual understanding based on explanation quality and scientific alignment.

2.1 Knowledge Base and Retrieval

Our system is built upon the SciEntsBank dataset, which comprises science assessment questions and student answers covering topics for grades 3 through 6. Because these questions require specific domain knowledge, off-the-shelf retrieval corpora were insufficient. To address this, we manually analyzed the dataset to identify core scientific concepts inherent in the SciEntsBank dataset. Based on these identified concepts, we constructed a custom knowledge base of 10,000 passages using GPT-4o-mini. This manual curation ensures that the retrieval system is strictly aligned with the educational content expected in the student answers.

To mirror the kinds of knowledge teachers rely on when evaluating explanations, the knowledge base was organized into three distinct categories. Roughly 3,400 passages provide concise factual explanations of core concepts, 3,300 passages describe underlying scientific processes and causal mechanisms, and 3,300 passages correct common student misconceptions. This balanced distribution ensures that retrieval surfaces definitional content, causal reasoning patterns and misconception-focused clarifications that are central to grade-level science understanding.

We generated MPNet embeddings for all 10,000 passages and utilized a FAISS index to enable efficient top-k similarity search. At inference time, a student question is encoded using MPNet and used to query this index. This ensures that the generator receives content-grounded information that is not just semantically similar, but pedagogically relevant to the specific grade-level science topics.

Separating retrieval from downstream generation and grading provides flexibility and analytic transparency. The knowledge base can be expanded or refined without retraining the entire system, and retrieval outputs can be inspected to diagnose whether grading errors stem from missing facts or irrelevant passages.

2.2 Reference Answer Generation

To overcome limitations of static gold references, we employ a **Retrieval-Augmented Generation** step. Using GPT-4o-mini, the system generates a teacher-style reference answer conditioned on the retrieved passages. Unlike short, exam-style gold references, RAG answers are often longer and richer in explanation, mimicking realistic instructional feedback. This stylistic difference is a key factor for the evaluation results.

Beyond producing richer explanations, this generation step plays a crucial alignment role in EduGrade's reasoning pipeline. Because the generator constructs reference answers dynamically rather than relying on static labels, it allows the system to adapt to paraphrased questions and unseen prompts. The generative output also makes grading more interpretable: instructors can inspect the generated reference to understand why the classifier judged a response as correct or incorrect. Importantly, this stage introduces a distributional shift: human gold references in SciEntsBank are brief and targeted, whereas GPT-derived explanations are multi-sentence, pedagogically styled narratives. That mismatch affects classifier behavior and motivates retraining the grader on RAG-style outputs.

2.3 Grading Model

EduGrade uses a fine-tuned **RoBERTa-large** classifier trained on the SciEntsBank dataset. The model receives an input template:

```
[QUESTION] ...
[REFERENCE] ...  [ANSWER]
```

The classifier predicts one of three labels: *correct*, *contradictory*, or *incorrect*. The baseline version of the system uses the original gold reference answers, while the full pipeline uses dynamically generated RAG references.

In addition to assigning labels, this stage plays a interpretive role in the overall pipeline. Since the classifier evaluates a structured input that includes the question, the generated reference, and the student's response, it effectively compares the student's reasoning to an idealized explanation rather than judging the answer in isolation.

This comparative format mirrors how teachers assess understanding in real classrooms, where the quality of justification often matters as much as factual correctness. Training on short teacher-written references but evaluating on longer generated ones changes how the model draws its decision boundaries. Ensuring that the grader is trained on inputs matching those it will encounter at inference time is critical for stable performance.

Source Code. The source code for EduGrade is available at: https://github.com/Shrey3009/NLP_Final-Project. The repository includes the full pipeline notebook, retrieval artifacts (.json, .npy, .faiss), and evaluation scripts. The grading model directory is excluded due to size limits but is required for reproducing classifier-level outputs.

3 System/Methodology Evaluation

We evaluated EduGrade using three complementary experiments. This approach was designed to isolate the performance of specific components within the pipeline, allowing us to determine if errors were coming from the retrieval process, the text generation, or the final grading model.

3.1 Experimental Setup

To ensure the results were reliable, we established a specific dataset partition, three distinct evaluation settings, and a set of metrics that cover both grading accuracy and text generation quality.

Dataset: We used the SciEntsBank benchmark for short-answer grading. All evaluations were performed on the official unseen partitions (test-u). This set contains questions that are completely absent from the training data. Using the unseen partition is necessary to test whether the model has actually learned scientific reasoning that can apply to new topics, rather than simply memorizing specific question-answer pairs it saw during training.

We designed three experiments to separate the contribution of each part of the system:

1. **Baseline Grader**: Gold reference answers passed directly into the RoBERTa classifier. This establishes a baseline for performance, showing us how well the grader works when given perfect input.

2. **RAG → Grader**: RAG-generated references replace gold references. This tests the actual performance of the full automated pipeline.

3. **RAG Answer Quality**: Intrinsic evaluation of RAG answers using BERTScore, cosine similarity, BLEU, ROUGE, and retrieval relevance. We compared the RAG-generated answers against the gold references to see how accurate and natural the generated text was.

Since the system performs two different tasks which is text generation and answer classification, we used different metrics for each task:

1. **For Grading Performance**
   (a) **Accuracy**: This measures the overall percentage of correct grades assigned by the system.
   (b) **Macro-F1**: This metric addresses class imbalance by weighting "Correct," "Incorrect," and "Contradictory" classes equally, ensuring performance is evaluated effectively across all categories rather than being skewed by frequent labels.

2. **For Generation Quality**
   (a) **BERTScore**: Utilizes contextual embeddings to assess semantic equivalence, determining if the generated answer conveys the same meaning as the reference regardless of lexical variation.
   (b) **Cosine Similarity**: Quantifies the mathematical alignment between the vector representations of the generated answer and the gold reference.
   (c) **BLEU & ROUGE**: These evaluate lexical overlap; BLEU focuses on precision (generated text present in reference), while ROUGE measures recall (reference text captured in generation).
   (d) **Retrieval Relevance**: Assesses the pertinence of retrieved documents to the input query, helping isolate whether inaccuracies stem from the generation component or incorrect source material.

3.2 Experimental Results

The results show that while the system is effective at retrieving and generating relevant information, integrating that information with the classifier introduces challenges related to input style.

Table 1 summarizes all three evaluations.

Evaluation 1: Baseline Grader. The RoBERTa classifier, when fed gold references, achieves an accuracy of 0.6016 and a macro-F1 of 0.4584. The gap between accuracy and macro-F1 indicates a problem with the contradictory class. Because contradictory cases are rare in SciEntsBank, the

Evaluation	What It Tests	Key Result
Baseline Grader	Gold reference → Grader	Acc = 0.6016
RAG → Grader	RAG reference → Grader	Acc = 0.6126
RAG Quality	RAG vs Gold reference	BERTScore = 0.878

Table 1: Summary of evaluation settings used in EduGrade. Results match values reported in the project presentation.

model sees fewer examples of them during training. Consequently, it tends to predict the dominant classes (Correct/Incorrect) more often. This boosts accuracy but lowers the macro-F1 score, confirming that the classifier relies on majority-class patterns and struggles with nuanced refutations.

Evaluation 2: Full RAG Pipeline. Replacing gold references with RAG-generated answers increased accuracy to 0.6126, but the macro-F1 score decreased. This happens because of a shift in data distribution: the RAG answers are generally longer and more explanatory than the short, exam-style gold references the classifier was trained on. The increase in accuracy suggests that the extra context helps the model make better decisions in ambiguous cases. However, the drop in macro-F1 indicates that the classifier's calibration is thrown off by the change in length and style, causing it to misclassify the difficult minority labels even more frequently.

Evaluation 3: Intrinsic RAG Quality. The generation metrics were strong. The BERTScore of 0.878 indicates high semantic fidelity, meaning the RAG system successfully preserves the core scientific meaning expressed in the gold reference. However, Retrieval Relevance was moderate. This suggests that the system occasionally fails to provide the necessary evidence to support the answer. This is likely a combination of retrieval errors and gaps in the underlying knowledge base.

Across these evaluations, several implications for system design become clear. Overall, the findings suggest that retrieval-augmented reasoning provides genuine value, but its full gains are unlocked only when downstream models are tuned to its output distribution. This creates an important design insight for educational NLP: generative supervision should be co-trained with the grader to align style, length, and semantic density. Without such adaptation, the benefits of improved knowledge grounding remain only partially realized.

3.3 Analysis

Analyzing these three experiments together allows us to draw several conclusions about the system's current state and necessary improvements.

- **Semantic correctness of RAG answers is high**. The high BERTScore confirms that the "Generation" component is functioning well. The generated answers capture the core scientific meaning effectively, even when they are lexically different from the gold references. This explains why exact-match metrics like BLEU might be lower while BERTScore remains high. The model is paraphrasing correctly rather than just copying.

- **The Classifier is the Bottleneck**. The main issue is the mismatch between the generated text and the classifier's expectations. The RoBERTa model was trained on concise gold references. When we feed it detailed explanations from the RAG system, we change the input distribution. The fact that accuracy improved despite this mismatch is a positive sign, because it proves the richer information is valuable. However, the instability in macro-F1 shows that the classifier needs to be retrained on RAG-style data to fully utilize this new information without getting confused by the length.

- **Knowledge Base Completeness and Retrieval Quality**. The moderate retrieval relevance scores point to a deeper issue regarding the completeness of our knowledge base. The system's ability to answer correctly is strictly limited by the information available in its index. Since we constructed the knowledge base by manually defining concepts and then using GPT-4o-mini to generate the relevant facts, it is highly probable that the knowledge base is not fully comprehensive. There are two likely sources of error here:

- **Manual Oversights**: During the manual identification of concepts, we may have missed specific topics or sub-topics required for certain test questions.
- **Generation Gaps**: Even if the concept was identified, GPT-4o-mini might not have generated every single relevant fact or nuance associated with that concept.

If a specific fact required to grade a student's answer was never generated or indexed, the RAG system cannot retrieve it, no matter how good the search algorithm is. This "missing data" problem leads to the generation of generic or hallucinated references, which in turn lowers the grading accuracy.

These results indicate that RAG adds value only when downstream models are aligned with its output style and the knowledge base is reliable. EduGrade demonstrates that adding more context helps, but it also reveals that reliable automated assessment requires two things: aligning the evidence construction (generation) with the evaluation (grading), and ensuring the underlying knowledge base is exhaustive. The most promising path forward is to co-train the grader with the generator and to perform a gap analysis on the knowledge base to ensure all relevant scientific facts are present.

4 Limitations and Future Work

While EduGrade demonstrates the clear potential of retrieval-augmented generation for automated grading, our analysis has identified specific limitations. These constraints provide a clear roadmap for the development needed to transition the system from a prototype to a robust educational tool.

4.1 Limitations

Distributional Mismatch Between Training and Inference. The primary bottleneck is a distributional mismatch between the training and inference phases. The RoBERTa classifier was originally fine-tuned on concise, human-written references but is expected to evaluate students using long, generated explanations. This fundamental shift in input distribution disrupts the model's calibration and directly causes the observed instability in macro-F1 scores, as the classifier struggles to process the additional context provided by the RAG system.

Knowledge Base Completeness and Retrieval Dependence. System performance is directly limited by the comprehensiveness of the knowledge base. The reliance on manual concept identification and GPT-4o-mini generation likely introduced coverage gaps that improved searching cannot resolve. Furthermore, the pipeline is highly sensitive to retrieval errors; if the retriever selects noisy or irrelevant evidence, the generator produces generic answers or hallucinations. This propagates errors downstream and deprives the grader of context needed to evaluate student answers accurately.

Interpretability and Trust. While generated reference answers expose partial reasoning, the final grading decision remains a black box. The classifier outputs a label without pointing to the specific part of the student's answer that caused the deduction. This lack of granular feedback limits the system's utility in real-world classrooms, where teachers and students require transparency to understand why an answer was marked incorrect or contradictory.

4.2 Future Work

To address these limitations, future iterations of EduGrade can focus on alignment, robustness, and explainability.

Classifier Re-training for Alignment. The immediate priority is to retrain the grading classifier on RAG-style data. Instead of training solely on original gold references, we can generate a training set using the RAG pipeline's own outputs. This will allow the classifier to learn the distribution, length, and style of generated references, resolving the mismatch issue and improving consistency across minority classes.

Enhancing Retrieval and Knowledge Coverage. To fix knowledge gaps, we can conduct a systematic review of the knowledge base against standard textbooks to identify missing concepts. We also propose implementing a cross-encoder reranking step to filter noisy search results and incorporating structured science ontologies to improve coverage. This hybrid approach will ensure the generator receives only the most relevant, conceptually accurate evidence.

Model Exploration and Explainability. We can evaluate using alternative open-weights models, such as Llama-3.2 or Qwen-2.5, to fine-tune the generator specifically for educational explana-

tions. Finally, to improve trust, we can integrate interpretability techniques like rationale extraction or attention visualization. This will allow the system to highlight specific sentences in a student's answer that contradict the reference, transforming EduGrade into a more actionable instructional support system.

5 Team Work Clarification

All members contributed equally to the development of EduGrade, with specific responsibilities divided as follows:

Sai Sri Harsha Chunduri was responsible for the retrieval module. He constructed the knowledge base, generated the MPNet embeddings, and implemented the FAISS index for vector search.

Rutuja More was responsible for the generation module and the grading model. She developed the RAG logic, managed the prompt engineering to synthesize references using GPT models, and trained and configured the grader used for final label prediction.

Shreyas Ajgaonkar was responsible for system integration and evaluation. He connected the modules into a single pipeline, ran the experimental trials, and compiled the performance metrics for the report.

The final presentation, analysis of results, and report writing were completed collaboratively by the entire team.

References

[1] Yucheng Chu, Peng He, Hang Li, Haoyu Han, Kaiqi Yang, Yu Xue, Tingting Li, Joseph Krajcik, and Jiliang Tang. Enhancing LLM-based short answer grading with retrieval-augmented generation, 2025.

[2] Myroslava Dzikovska, Rodney Nielsen, and Chris Brew. SemEval-2013 task 7: The SciEntsBank corpus for evaluation of textual entailment in student answers. In *Proceedings of SemEval*, 2013.

[3] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Saar Kuzi, Saeid Momtazi, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *NeurIPS*, 2020.

[4] Chul Sung, Tejas Indulal Dhamecha, and Nirmal Mukhi. Improving short answer grading using transformer-based pre-training. In *Artificial Intelligence in Education*, pages 469-481, 2019.