RATAS v2 (From Rubric to Feedback): Towards Rubric-Aligned and Subject-Agnostic Automated Text Assessment with Explainable Scoring

In this chapter, we introduce **RATAS v2**, a second-stage framework that extends RATAS to address a key practical limitation observed in previous Chapter: **reduced scoring stability as response length increases**. RATAS v2 strengthens the grading pipeline by coupling the **Rubric Knowledge Tree (RKT)** [27] with a **fine-tuned foundation model** that performs response analysis prior to scoring. The role of this model is not to replace rubric-driven reasoning, but to **improve evidence localization**, i.e., to identify the most influential parts of a response for each criterion, so that rubric alignment is preserved even when relevant evidence is scattered across longer answers. In parallel, the RKT representation is leveraged not only as a hierarchical encoding of rubric criteria, but also as a contextual scaffold that supports richer rubric interpretation through explicit rubric structure and inferred semantic relations. Together, these two components improve the accuracy and consistency of criterion-level judgments and yield more reliable aggregated grades for semi-long responses while preserving rubric-indexed traceability.

Similarly, RATAS v2 retains a **task-decomposition** strategy and operationalizes scoring through a set of focused *Downstream NLP Tasks (DNTs)*. However, its DNT design is refined to reduce the risk of local decisions drifting from global grading objectives (e.g., fairness, rubric fidelity, and explainability). This is achieved through an orchestrated integration strategy that defines DNT boundaries around rubric-conditioned evidence selection and criterion evaluation, applies tailored methods for each task, and aggregates outputs in a controlled way that preserves end-to-end scoring coherence. Our evaluation

shows that these changes improve **length robustness** relative to RATAS and outperform direct GPT-4o prompting on the same dataset, particularly for responses where content is distributed across multiple sections and where verbosity can otherwise bias criterion-level assessments.

Finally, we designed and implemented a multi-layer software system in which the AI layer implements RATAS v2 and the surrounding layers (front end, back end, and integration services) support an end-to-end assessment workflow. The platform enables instructors to design exams using authentic, complex rubrics and to operationalize flexible scoring logic through an interface that supports rule definition, rule combination, and rubric reuse. It supports automatic scoring of submitted responses and produces structured, rubric-aligned feedback and answer analysis in a form suitable for instructional review.

6.1 Workflows and Processes

Figure 6.1 summarizes the end-to-end workflow of **RATAS v2**. While RATAS v2 inherits the rubric-centered philosophy established earlier, its workflow introduces a **distinct response-side stage** that explicitly targets the key weakness observed in RATAS: when answers become longer, relevant evidence for a criterion may be distributed across different parts, and naive criterion-answer matching becomes less stable. RATAS v2 addresses this by inserting an **evidence localization step** between rubric contextualization and scoring. As a result, the workflow can be understood as three tightly coupled phases: (i) **rubric contextualization** (RKT construction), (ii) **answer evidence extraction** (effective content selection), and (iii) **criterion-conditioned scoring and aggregation**.

Step 1 (a): Constructing the RKT (Rubric-side contextualization). Similar to RATAS, the workflow begins by constructing a **Rubric Knowledge Tree (RKT)** (Step 1 in Fig. 6.1). The RKT provides a hierarchical representation of rubric criteria and associated metadata required for rubric-aligned scoring. In RATAS v2, the role of the RKT is not merely structural; it serves as the **indexing backbone** that connects rubric elements, extracted evidence from the answer, and scoring and feedback outputs. This explicit indexing is essential for generating criterion-level feedback that remains auditable under long-form responses.

Similarly, at each iteration, a rubric criterion is decomposed into smaller, semantically coherent components using **Criteria Topic Modeling (CTM)**. The output of CTM is a set of derived rule fragments that become child nodes. The leaf nodes produced by this process are referred to as **Simplest Rules (SRs)**, and they form the atomic units for subsequent rubric-answer alignment and scoring.

Step 1(b): Aligning SRs with achievement levels (CSC). After CTM produces SR nodes, RATAS v2 applies **Criteria Sub-condition Classification (CSC)** to associate each SR with the appropriate rubric performance descriptors (i.e., the rubric's achievement-level statements). This step is crucial for two reasons. First, it ensures that each SR is evaluated using the **correct notion of quality** intended by the rubric (rather than using

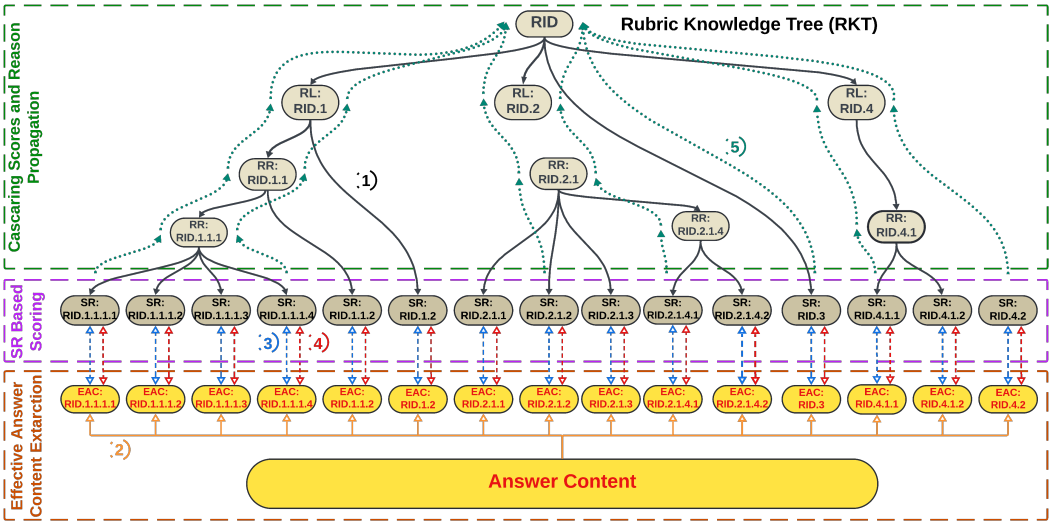

FIGURE 6.1: RATAS v2 end-to-end grading process and data flow. The rubric is first expanded into a RKT, where rubric nodes are decomposed into SRs. For each SR, the *Effective Answer Extracting* step selects the most relevant evidence spans from the student answer (EACs) to minimize noise and improve traceability. The *SR-based scoring* then computes an SR-level score and justification from the extracted evidence, and the resulting scores/reasons are propagated upward through the RKT hierarchy.

a generic scoring heuristic). Second, it makes later explanations more faithful: when RATAS v2 generates a rationale, it can explicitly reference the relevant achievement-level description that the student's response matches (or fails to match).

Step 2: Effective Answer Extracting (Answer-side contextualization). The defining workflow addition in RATAS v2 is the **Effective Answer Extracting (EAE)** step (Step 2 in Fig. 6.1). EAE is applied *per SR* to identify the most relevant part(s) of the student response for evaluating that SR. This step directly targets a common failure mode in rubric-based scoring for semi-long answers: the answer may include correct evidence, but it may be scattered among unrelated content, making direct rubric-answer matching noisy and length-sensitive.

Operationally, EAE takes the SR (and its rubric context) as a query and produces an **effective segment**, a reduced, focused subset of the response that is maximally informative for that SR. This yields two practical benefits that are central to RATAS v2: (i) it reduces the risk that verbosity or irrelevant narrative dominates the scoring decision, (ii) it provides a clear evidence object that can be surfaced in feedback (e.g., "this excerpt supports criterion X").

Step 3: SR-based Scoring and Reasoning (SSR) over extracted evidence. After EAE identifies an effective response segment for an SR, RATAS v2 applies **SR-based Scoring and Reasoning (SSR)** (Step 2 in Fig. 6.1) to determine the SR-level outcome. SSR evaluates how well the extracted segment satisfies the SR requirement by jointly considering: (i) the SR text itself, and (ii) the effective content selected by EAE,

(iii) the relevant achievement-level descriptor(s) selected by CSC. This design makes SR-level decisions more stable for longer answers, because the model is asked to score a **targeted evidence segment** rather than the full response. It also improves feedback quality, because the scoring rationale can explicitly cite the extracted segment as the basis for the decision.

Step 4: Cascading aggregation of scores and reasons. In the final phase (Step 3 in Fig. 6.1), RATAS v2 aggregates SR-level scoring results into **partial and final scores**, and composes SR-level explanations into rubric-indexed reasons at higher levels of the RKT. This aggregation propagates from SR nodes upward to internal nodes and rubric-row nodes, producing a structured assessment artifact that aligns with the rubric's hierarchy. Importantly, because each SR is backed by an extracted evidence segment (from EAE) and an achievement-level alignment (from CSC), the aggregated reasoning can remain **grounded** even when the original answer is long and heterogeneous.

6.2 Model Architecture and Implementation

Figure 6.2 presents the high-level architecture of RATAS v2. The framework is organized around three primary engines: **RKT construction**, **answer analysis**, and **scoring and aggregation**. This separation reflects the staged workflow shown in Fig. 6.1: rubric-side contextualization (CTM+CSC) produces stable criterion units (SRs with aligned achievement descriptions), answer-side analysis (EAE) produces SR-specific evidence segments, and scoring integrates the two through SSR and cascading aggregation.

Across engines, RATAS v2 maintains an orchestration layer that coordinates module execution, prepares well-scoped inputs for each DNT, and integrates intermediate outputs into final grading artifacts. This orchestration is necessary because task decomposition, while beneficial for model reliability and input-length control, can otherwise introduce local inconsistencies (e.g., extracting evidence for one SR that implicitly depends on content needed for another SR). RATAS v2 mitigates this by enforcing consistent indexing through the RKT, tracking SR-to-evidence links, and aggregating results in a controlled manner that preserves rubric-level coherence [27].

Addressing DNTs is a critical component of the RATAS v2 framework because the reliability of each downstream output directly affects the quality of evidence selection, criterion-level decisions, and the final aggregated score. Although DNTs are narrower in scope than the full RAGT problem, they remain non-trivial NLP tasks and require careful implementation to ensure stable, rubric-aligned behavior. Consistent with RATAS and our comparative analysis of alternative strategies in previous chapter, RATAS v2 employs the **GPT-4o Assistant API** to support **subject-agnostic** operation without requiring

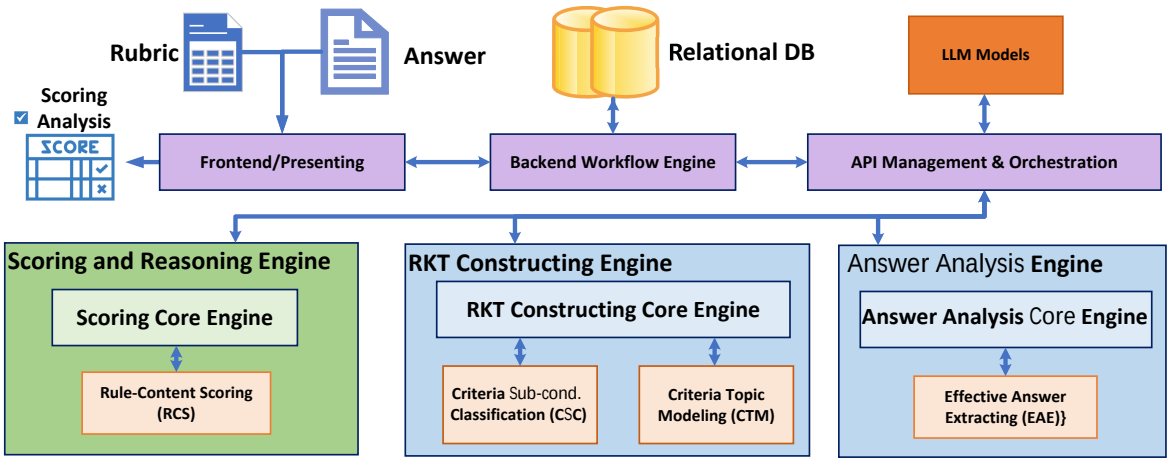

FIGURE 6.2: High-level architecture (HLA) of RATAS v2 as an API-driven automated grading service. A *frontend/presenting* layer supports rubric authoring and score analysis, while a *backend workflow engine* coordinates the grading pipeline and persists artifacts in a relational database. The core AI functionality is modularized into three engines: an *RKT constructing engine* (including CTM and CSC), an *answer analysis engine* (EAE for evidence extraction), and a *scoring and reasoning engine* (RCS for rubric-aligned scoring). An *API management and orchestration* layer mediates calls to LLM services and integrates system components for reliable execution.

domain-specific fine-tuning. To improve accuracy and reduce output variance across DNT executions, we use a structured prompting strategy that combines instruction prompting with few-shot exemplars and (where appropriate) Chain-of-Thought-style guidance. The prompts for the newly introduced DNTs in RATAS v2 are documented in the project repository (datalab912/RATASv1). For completeness and reproducibility, the core prompts and templates are also provided in the Appendix.

6.3 Evaluation and Results

We evaluated **RATAS v2** against the original **RATAS** and direct **GPT-4o prompting** on the same university-level dataset used throughout this thesis. The evaluation set contains **418** rubric-scored responses spanning **14 to 1,285 words** (average length: **366**), paired with authentic structured rubrics containing **34** detailed scoring criteria . To study length robustness explicitly, we report results on the full dataset as well as two subsets: **shorter** responses (14 to 600 words; N=37) and **longer** responses (600 to 1,285 words; N=379). Performance is measured using **MAE** (lower is better), **Pearson correlation** r (higher is better), and R^2 (higher is better).

Overall performance (14 to 1,285 words). Across all responses, RATAS v2 achieves **MAE = 0.0295**, $r = 0.9837$, and $R^2 = 0.9677$ (Table 7.2). These results indicate that RATAS v2 produces scores that are both **highly accurate** (low absolute error) and **strongly aligned** with human grading (high correlation and high explained variance).

In comparison, direct GPT-4o prompting yields substantially weaker performance on the same data (**MAE = 0.2355**, $r = 0.3743$, $R^2 = -0.6262$), showing that generic prompting does not reliably reproduce rubric-conditioned scoring on this dataset.

Shorter responses (14 to 600 words). For shorter answers (N=37), RATAS v2 maintains strong accuracy (**MAE = 0.0288**) and very high alignment with human scores ($r = 0.9842$, $R^2 = 0.9686$). RATAS performs similarly well on this subset (**MAE = 0.0280**, $r = 0.9848$, $R^2 = 0.9696$), indicating that both frameworks are highly effective when responses are compact and relevant evidence is relatively concentrated. By contrast, GPT-4o remains far less reliable even on shorter inputs (**MAE = 0.2313**, $r = 0.4227$, $R^2 = -0.6793$), reinforcing that rubric alignment and structured reasoning are necessary even when answers are not long.

Longer responses (600 to 1,285 words) and length robustness. The most important difference emerges on the longer subset (N=379), which dominates the dataset and reflects semi-long answers where relevant content may be distributed across the response. Here, RATAS v2 achieves **MAE = 0.0364**, $r = 0.9777$, and $R^2 = 0.9558$, while RATAS drops to **MAE = 0.0603**, $r = 0.9519$, and $R^2 = 0.8874$. This demonstrates that RATAS v2 substantially improves **length robustness**: compared to RATAS on long answers, RATAS v2 reduces absolute error by **0.0239** (from 0.0603 to 0.0364) and improves explained variance by **0.0684** (from 0.8874 to 0.9558). These gains are consistent with the design goal of RATAS v2: **localizing influential evidence** in longer responses before scoring, which reduces sensitivity to verbosity and irrelevant content while preserving rubric-aligned decisions.

Comparison to direct GPT-4o prompting. Across all splits, GPT-4o prompting exhibits weak correlation and negative R^2, with performance degrading further on longer answers ($r = -0.2582$, $R^2 = -0.9181$). A negative R^2 indicates that, under this evaluation protocol, the prompted model performs worse than a simple baseline that predicts the mean score. This behavior highlights the practical risk of using direct prompting for rubric-based scoring: without explicit rubric contextualization and criterion-conditioned evidence handling, the model can become unstable as response length increases and as rubric logic becomes more structured.

Output consistency. In terms of reliability and output consistency, both RATAS and RATAS v2 achieved high agreement scores (approximately 0.95), substantially higher than GPT-4o (0.6).

Summary of findings. Overall, Table 7.2 shows that RATAS v2: (i) matches RATAS on shorter answers, (ii) substantially improves scoring accuracy and consistency on longer answers, and (iii) decisively outperforms direct GPT-4o prompting across all metrics.

These results support the central claim of RATAS v2: incorporating rubric-contextual knowledge together with targeted evidence extraction yields a more reliable rubric-aligned

TABLE 6.1: Evaluation results of RATAS v2 compared to RATAS and GPT-4o.

Approach	Data Set	N	MAE	Pearson r	R^2
RATAS v2	All Data(14-1285)	418	0.0295	0.9837	0.9677
RATAS v2	Shorter (14-600)	379	0.0288	0.9842	0.9686
RATAS v2	Longer(600-1285)	39	0.0364	0.9777	0.9558
RATAS	All Data(14-1285)	418	0.0309	0.9813	0.9627
RATAS	Shorter (14-600)	379	0.0280	0.9848	0.9696
RATAS	Longer(600-1285)	39	0.0603	0.9519	0.8874
GPT 4o	All Data(14-1285)	418	0.2355	0.3743	-0.6262
GPT 4o	Shorter (14-600)	379	0.2313	0.4227	-0.6793
GPT 4o	Longer(600-1285)	39	0.2768	-0.2582	-0.9181

scoring pipeline for semi-long responses, while preserving the structured feedback and traceability required in real educational assessment.

6.4 Database Structure of the RATAS v2 Software

Figure 6.3 presents the relational database design used in the RATAS v2 software system. The schema is organized around three connected layers: (i) **academic context** (courses and semesters), (ii) **assessment design** (exams, questions, rubrics, and rubric criteria), and (iii) **assessment delivery and student data** (enrolment, exam participation, and student answers).

This structure enables RATAS v2 to (a) reuse rubric definitions across offerings, (b) maintain traceability between answers and the exact rubric criteria used for scoring, and (c) support end-to-end workflows from exam creation to response submission and analysis. Table 6.2 summarizes the core database entities and their primary keys used in the RATAS v2 system.

6.4.1 Relationships and Cardinalities

Academic context. A **Course** can be offered in multiple **Semesters**, and each **Semester** can include multiple **Courses**. This many-to-many relationship is implemented through **CourseInstance**:

$$\text{Course} \longleftrightarrow \text{CourseInstance}(\text{Courseid}, \text{Semesterid}) \longleftrightarrow \text{Semester}.$$

This design allows the system to manage repeated offerings of the same course across different terms without duplicating course-level definitions.

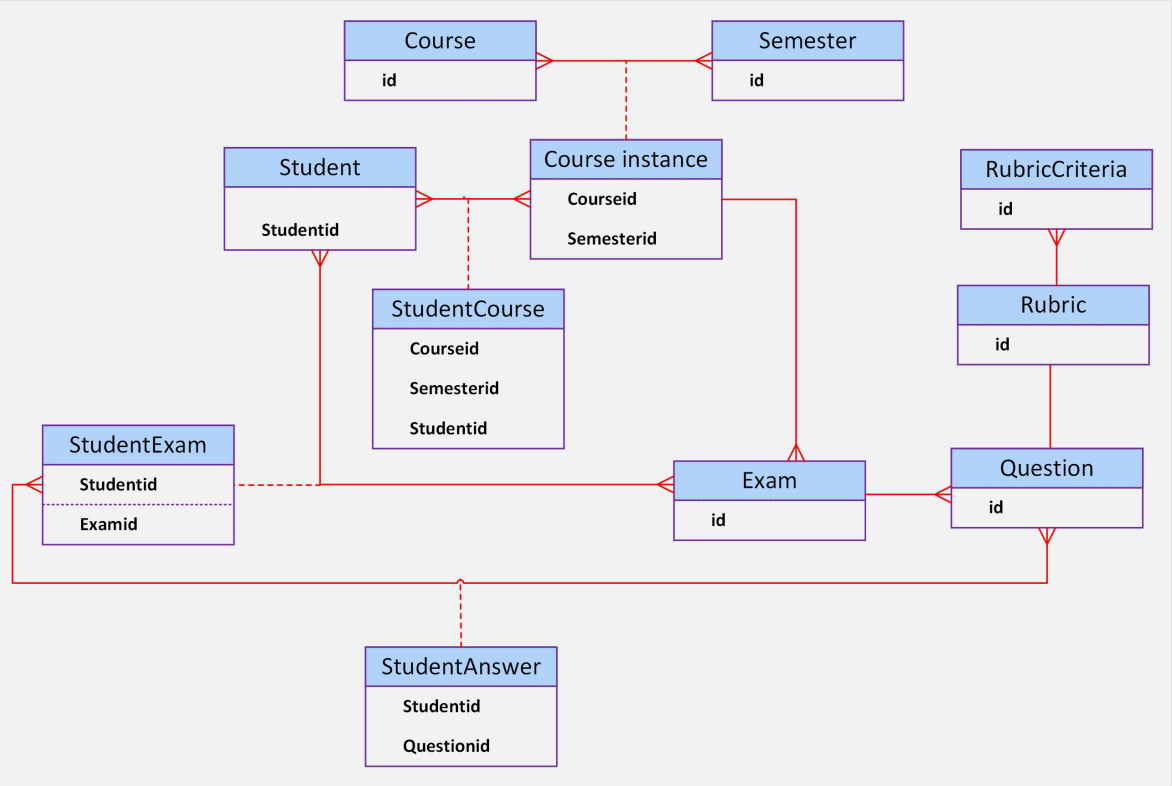

Figure 6.3: Relational database schema supporting RATAS v2 data management and grading workflows. The schema links academic context (*Course*, *Semester*, *Course instance*) to assessments (*Exam*, *Question*) and rubric artifacts (*Rubric*, *RubricCriteria*). Student participation is represented through junction tables (*StudentCourse*, *StudentExam*), while *StudentAnswer* stores per-student responses keyed by student and question to enable rubric-aligned scoring, aggregation, and downstream analysis.

Student enrollment. Students enroll in a specific course offering (not just a course in general). This is modeled by **StudentCourse**, which links **Student** to **CourseInstance**:

$$\text{Student} \longleftrightarrow \text{StudentCourse}(\text{Courseid}, \text{Semesterid}, \text{Studentid}) \longleftrightarrow \text{CourseInstance}.$$

This supports enrollment tracking per semester and enables filtering of exams and questions for the correct cohort.

Assessment design. Each **CourseInstance** can have multiple **Exams**. Each **Exam** can have multiple **Questions**. For rubric-based scoring, each **Question** is associated with a **Rubric**, and each **Rubric** contains multiple **RubricCriteria**:

$$\text{CourseInstance} \rightarrow \text{Exam} \rightarrow \text{Question} \rightarrow \text{Rubric} \rightarrow \text{RubricCriteria}.$$

This chain is important for RATAS v2 because it establishes a deterministic mapping from a submitted answer to the exact rubric criteria used to generate SR-level scoring decisions

TABLE 6.2: Core Database Entities and Primary Keys

Entity (Primary Key)	Description
Course (**id**)	Defines a course as an academic unit.
Semester (**id**)	Defines an academic term (e.g., Fall 2024).
CourseInstance (**Courseid**, **Semesterid**)	Represents a specific offering of a course in a given semester; associative entity between Course and Semester.
Student (**Studentid**)	Stores student identity information.
StudentCourse (**Courseid**, **Semesterid**, **Studentid**)	Enrollment table linking a student to a specific course instance.
Exam (**id**)	Represents an assessment event (e.g., midterm or final) associated with a course instance.
Question (**id**)	Represents a question within an exam.
Rubric (**id**)	Stores the rubric definition associated with a question.
RubricCriteria (**id**)	Stores individual rubric criteria (rows/items) associated with a rubric.
StudentExam (**Studentid**, **Examid**)	Participation table linking a student to a specific exam instance.
StudentAnswer (**Studentid**, **Questionid**)	Stores a student's submitted answer for a specific question.

and rubric-indexed feedback.

Exam participation and answers. A student may sit multiple exams, and each exam may be taken by many students; this many-to-many association is represented by **StudentExam**(**Studentid**, **Examid**). For each question in an exam, the student submission is stored in **StudentAnswer**(**Studentid**, **Questionid**), linking the answer to both the student and the question (and therefore to its rubric through the Question → Rubric relationship).

6.4.2 Why this schema supports RATAS v2 workflows

This schema provides the minimum structural requirements to support RATAS v2's operational pipeline:

- **Rubric traceability:** every `StudentAnswer` links to `Question` and therefore to a unique `Rubric` and its `RubricCriteria`.
- **Cohort correctness:** `StudentCourse` ensures that only enrolled students are associated with a given course offering and its exams.

- **Exam integrity:** `StudentExam` separates exam participation from per-question answers, enabling consistent exam-level bookkeeping while storing answers at question granularity.
- **Reuse across semesters:** `CourseInstance` allows the same course structure to be offered across terms, while keeping exam and submission data term-specific.

6.5 User Experience, Interface Design, and System Capabilities

To operationalize RATAS v2 in real educational workflows, we implemented a web-based assessment platform that exposes the framework through an instructor-facing user experience. The platform is designed around two UX goals: (i) lowering the effort of authoring and maintaining authentic rubrics, and (ii) making automated scoring transparent and inspectable through rubric-indexed, evidence-grounded feedback.

At the system level, the solution follows a service-oriented design (Fig. 6.2): requests pass through an API management and orchestration layer for authentication, routing, and traffic control, while long-running operations are executed asynchronously using job IDs. Lightweight FastAPI services validate requests and persist core entities, RabbitMQ supports asynchronous task execution, Redis tracks job state and coordination, and PostgreSQL acts as the system of record for courses, exams, rubrics, student answers, and structured reasoning outputs.

6.5.1 Rubric Authoring and Flexible Rule Composition

Figure 6.4 shows the rubric authoring interface, which is intentionally aligned with how instructors write rubrics in practice: as **plain-text rules** and performance descriptions rather than rigid schemas. The UI supports creating rubric rows with **Score-Source** allocations and writing **Basic-Rule** and **Level-of-Achievement** statements directly in natural language. This design enables instructors to express both simple criteria and more realistic, multi-part requirements without needing to manually encode formal logic. Instructors can also **combine multiple rule statements sequentially** within a row, leveraging the flexible scoring logic introduced earlier in the thesis. The system then uses these authored rules as the basis for rubric contextualization and scoring, ensuring that the grading pipeline remains anchored to instructor intent rather than implicit model heuristics.

6.5.2 Rubric Contextualization and RKT Exploration in the UI

After rubric creation, the platform analyzes the rubric and constructs the **Rubric Knowledge Tree (RKT)**. From a UX perspective, the RKT is treated as the system's **explainability spine**: every score and feedback element is traceable back to a specific rubric component indexed by the RKT. To support rubric review and validation, the interface provides

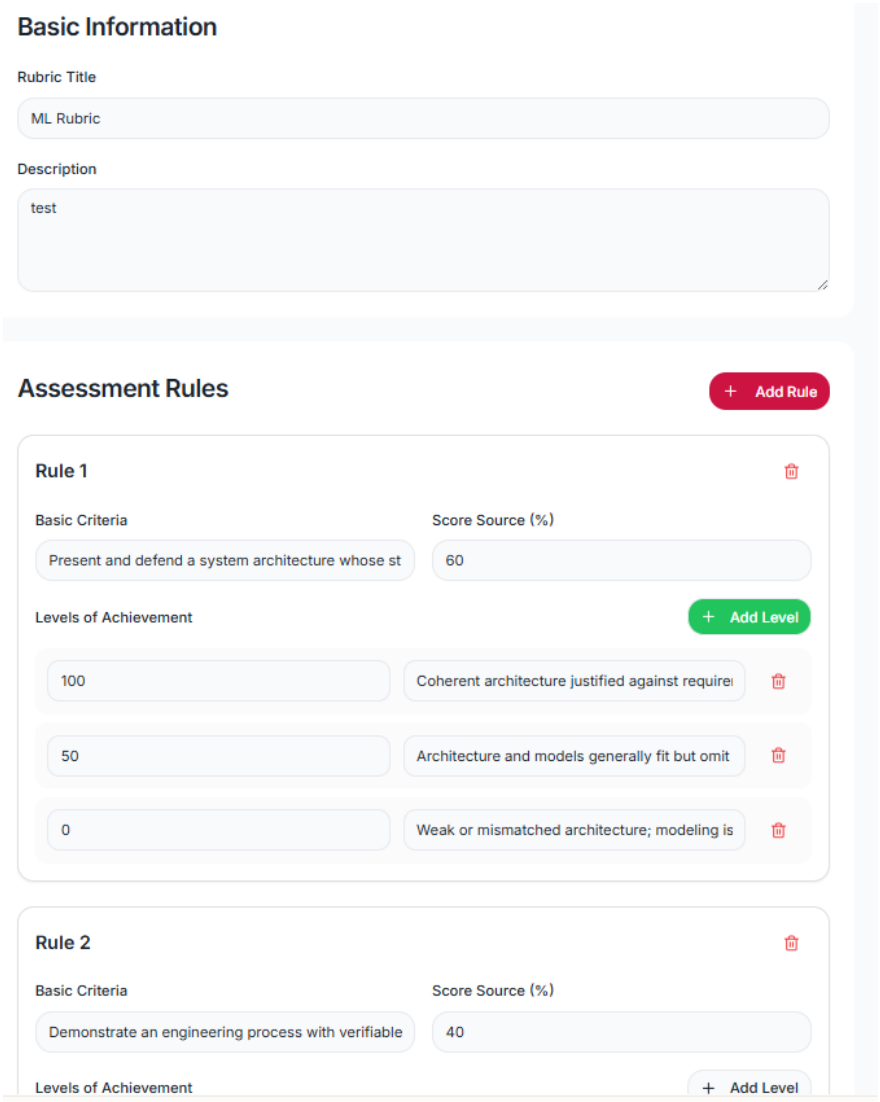

FIGURE 6.4: Rubric authoring interface for RATAS v2, supporting plain-text rule composition and achievement-level descriptions.

multiple ways to explore the constructed RKT (e.g., tree and list views). Selecting a node reveals the contextualized rubric information associated with that node, allowing instructors to inspect how rubric rules were decomposed into SR-level units and how achievement-level descriptors were associated with them. This interaction is particularly important for adoption in real settings, where instructors often need to verify rubric interpretation before trusting automated scoring outputs.

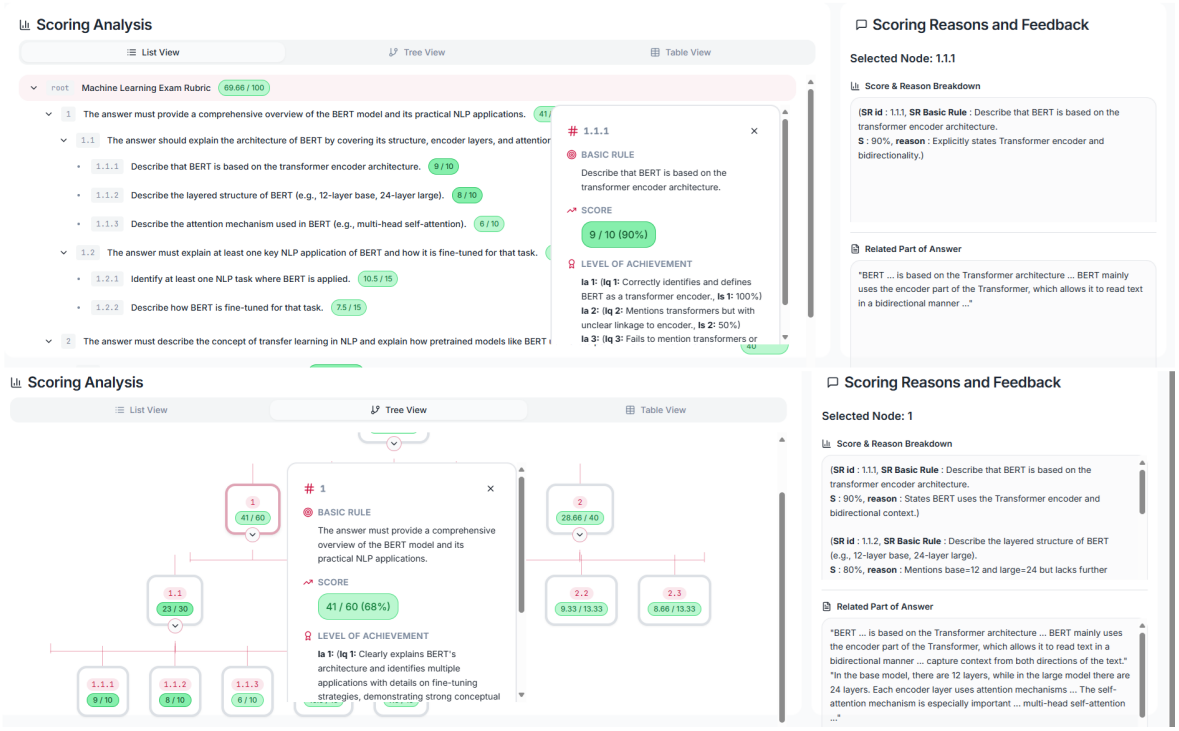

FIGURE 6.5: RATAS v2 scoring and reasoning interface, enabling rubric-indexed selection, partial-score inspection, and structured feedback grounded in the student response.

6.5.3 Structured Feedback, Evidence Traceability, and Interactive Answer Analysis

Figure 6.5 presents the scoring and reasoning analysis page, which is the primary interface for interpreting RATAS v2 outputs. The core capability of this page is **interactive, rubric-indexed inspection**: instructors can click individual criteria (or combinations of criteria) and immediately view three aligned elements:

1. **The relevant portion of the student response** selected for the chosen rubric element(s), enabling evidence-based review.

2. **The computed score** for the selected rubric scope (criterion-level or aggregated across multiple nodes), supporting partial-score inspection in addition to the final grade.

3. **Structured reasoning and feedback** that explains why the score was awarded, explicitly tied to the rubric requirements and achievement descriptions.

This design turns the scoring output into a navigable explanation object rather than a single opaque number. It also supports practical instructor workflows such as: auditing borderline cases, identifying missing rubric components, and using rubric-aligned feedback as formative guidance for students.

6.5.4 Operational Features Supporting Real Assessment Workflows

Beyond scoring and feedback visualization, the platform includes supporting features required in real assessment settings, including management of students, exams, rubrics, and submitted responses. These capabilities ensure that RATAS v2 functions as part of an end-to-end grading workflow rather than a standalone research prototype: instructors can create and reuse rubrics, associate rubrics with questions, submit or upload student responses, and retrieve structured scoring outputs consistently across multiple assessment instances.

6.6 Conclusion

This chapter introduced **RATAS v2**, an enhanced rubric-based scoring framework designed to improve the reliability of automated grading for **semi-long** open-ended responses while preserving **rubric fidelity** and **explainability**. Building on the rubric contextualization foundations of the RKT, RATAS v2 strengthens response interpretation by adding an explicit **answer evidence localization stage** through **Effective Answer Extracting (EAE)** and a criterion-conditioned scoring stage through **SR-based Scoring and Reasoning Scoring (SSR)**. Together, these additions reduce the impact of irrelevant verbosity and dispersed evidence, yielding more stable criterion-level decisions and more consistent aggregation into partial and final scores.

Empirically, RATAS v2 demonstrated strong performance across the full evaluation set and, most importantly, delivered **substantial gains on longer responses** (600 to 1,285 words) compared to RATAS, while maintaining the high accuracy achieved on shorter responses. In addition to accuracy improvements, RATAS v2 operationalizes rubric-aligned feedback in a form that is inspectable through the platform UI: scores and explanations are indexed by rubric components, enabling instructors to review evidence, audit scoring decisions, and communicate actionable improvement points in a rubric-consistent manner. This combination of length-robust scoring and structured, navigable feedback advances the practical viability of rubric-based automated grading in real educational workflows.

Despite these improvements, two practical limitations remain. First, RATAS v2 is designed primarily around rubrics that can be decomposed into SR-level criteria and assessed through localized evidence extraction; however, **real-world rubrics frequently contain richer and more interdependent rule types**, including **structural requirements** (e.g., section presence and organization), **essential prerequisites** (e.g., mandatory criteria that gate downstream scoring), and **explicit logical relations** among criteria (e.g., **AND/OR** combinations and conditional dependencies). Such rubrics introduce scoring behaviors that cannot always be expressed or executed as independent SR evaluations followed by simple aggregation. Second, while EAE improves evidence localization for semi-long answers, the framework still assumes that criterion-level evidence can be adequately captured through limited excerpts; this assumption becomes increasingly fragile for **multi-page reports** where rubric compliance depends on discourse structure, cross-section consistency, and long-range dependencies