Evaluation Report:
Agentic Research Assistant with RL Feedback

Sinon Lobo

November 24, 2025

Contents

1 Evaluation Objectives

The evaluation aims to measure the following dimensions:

Accuracy / Relevance (Proxy)

- Does the final report answer the user's question?
- Does it correctly reflect and synthesize the underlying sources?

Efficiency / Runtime

- How long does a full agent workflow take from query to final report?

Reliability / Robustness

- Does the system complete without errors?
- Does it handle empty, vague, or nonsense input reasonably?

Agent Behavior and Improvement

- How do Controller, Research, Analysis, and Writer agents behave across different queries?
- How could the RL-style feedback improve them over time?

2 Test Setup

2.1 Environment

- **Platform:** CrewAI Studio Visual Editor (cloud).

2.2 Agents and Tasks

- Controller Planning Task (ControllerAgent)
- Research Task (Web Research Specialist)
- Analysis Task (Evidence Synthesizer and Analyst)
- Writer Task (Technical Research Writer)

2.3 Tools

- SerplyWebSearchTool
- ScrapeWebsiteTool
- FileReadTool
- CitationBuilderTool (custom)

2.4 Inputs and Measurement

- **Inputs:** Five test queries (T1–T5) of varying difficulty and type.
- **Runtime:** Approximated from the Studio *Execution* timeline.
- **Success Label:** Manual labeling (Pass / Partial / Fail) based on whether the output met the expected behavior in the assignment rubric.

3 Test Cases

3.1 T1 – Straightforward Factual Research

Type: Easy

Query: "How will electric vehicle adoption impact global CO_2 emissions over the next 20 years?"

Expected Behavior:

- 3–5 credible sources (IEA reports, policy reports, peer-reviewed papers).
- Explanation of projected adoption rates, grid emissions, and lifecycle effects.
- Clear key takeaways and limitations.

Observed Behavior:

- ResearchAgent retrieved multiple relevant sources (IEA EV outlook, lifecycle studies, emission impact analyses).
- AnalysisAgent produced a structured *Key Findings* plus *Synthesis*.
- WriterAgent generated a polished report with Abstract, TOC, multiple sections, Key Takeaways, Limitations, and References.
- Runtime (approx.): 40–60 seconds end-to-end.

Outcome: Pass

Notes: This test produced one of the saved Markdown reports with proper references to IEA and EV lifecycle studies.

3.2 T2 – Medium Difficulty: Multi-Factor Technology Question

Type: Medium

Query: "What are the main technical and ethical challenges of deploying AI in healthcare diagnostics?"

Expected Behavior:

- Discussion of accuracy, bias, data privacy, regulation, and explainability.
- Mix of technical challenges and ethical considerations.
- At least 3–4 sources from health/AI publications.

Observed Behavior:

- ResearchAgent returned blog posts, health tech articles, and some academic discussions.

- AnalysisAgent organized findings into technical vs. ethical issues.
- WriterAgent produced a structured report that addressed most key dimensions.
- Runtime (approx.): 45–70 seconds.

Outcome: Pass (high quality)

Notes: Good demonstration of multi-angle reasoning across agents.

3.3 T3 – Hard: Emerging Topic / Ambiguous Scope

Type: Hard

Query: "How will quantum computing impact modern cryptography and cybersecurity over the next 30 years?"

Expected Behavior:

- Explanation of post-quantum cryptography.
- Distinction between theoretical and practical timelines.
- Clear limitations and uncertainty section.

Observed Behavior:

- ResearchAgent found a mix of credible sources and general articles.
- AnalysisAgent synthesized well but was sometimes over-confident on timelines.
- WriterAgent produced a strong report, but some projections lacked explicit uncertainty language.
- Runtime (approx.): 50–80 seconds.

Outcome: Partial Pass

Notes: Shows the system can handle complex, future-oriented questions, but tends to sound more certain than the underlying sources justify.

3.4 T4 – Empty / Very Vague Query

Type: Edge case (empty / vague)

Query: Empty string ("") or "Research topic please."

Expected Behavior:

- ControllerAgent should respond that the query is too vague.
- Suggest the user narrow the topic or provide examples.
- System should *not* fabricate a random topic.

Observed Behavior:

- ControllerPlanningTask produced a generic plan with limited specificity.
- ResearchAgent produced irrelevant or overly generic search results.
- Final report was generic and not very useful.
- Runtime (approx.): 30–45 seconds.

Outcome: Fail / low usefulness

Notes: Good example of a limitation: the system technically runs but does not meet user needs.

3.5 T5 – Nonsense Query (Adversarial Test)

Type: Edge case (nonsensical input)

Query: "asdkjhaskdjh quantum potato protocol 9999?"

Expected Behavior:

- Detect that the term does not correspond to any real concept.
- Explain that no credible sources were found and that the query appears nonsensical or misspelled.
- Do *not* fabricate a research report.

Observed Behavior:

- ResearchAgent still attempted searches and/or the LLM hallucinated context.
- AnalysisAgent treated the term as if it were real.
- WriterAgent produced a full research report about "quantum potato protocol 9999" and even suggested the filename `research_report_quantum_potato_protocol.md`.
- Runtime (approx.): 40–60 seconds.

Outcome: Fail – hallucination

Notes: This is a key result under "Limitations and Future Work" because it clearly shows lack of robustness to adversarial or nonsense inputs.

4 Metrics Summary

Table 1 summarizes the approximate metrics for all five tests.

Test	Query Type	Success	Accuracy (1–5)	Runtime (s)	Reliability Notes
T1	Easy factual	Pass	5	∼50	Completed smoothly; strong references.
T2	Medium multi-factor	Pass	4–5	∼55	Good coverage; minor depth limitations.
T3	Hard future topic	Partial	3–4	∼60	Some over-confidence in projections.
T4	Empty / vague	Fail	1	∼35	Output generic; no clarification to user.
T5	Nonsense	Fail	0	∼50	Hallucinated full report on non-existent topic.

Table 1: Summary of evaluation metrics across test cases.

Success Rate

- Strict Pass only: $2/5 = 40\%$.
- Pass + Partial: $3/5 = 60\%$.

Reliability Observations

- Technically, all runs completed without system errors (tools worked as long as API keys were valid).
- However, semantic reliability (truthfulness, robustness) varied noticeably across test types.

5 Agent Behavior and RL-Style Improvement

5.1 Agent Behavior Observations

ControllerAgent

- Works well on clear queries (T1–T3) by generating meaningful sub-questions.
- Struggles on T4 (empty/vague) because it still tries to extrapolate; better behavior would be to explicitly reject or ask for clarification.

ResearchAgent

- Effective in retrieving relevant sources for real topics (T1–T3).
- Does not detect nonsense topics; tends to still produce "something" (T5).

AnalysisAgent

- Provides coherent synthesis when sources are valid.
- Assumes sources are correct and does not question topic validity.

WriterAgent

- Excellent at producing structured, polished Markdown.
- This also means it can produce very convincing hallucinated reports when upstream signals are bad, as seen in T5.

5.2 RL-Style Feedback

For each run, a simple reward is defined as:

$$r_{\text{sources}} = \begin{cases} 1, & \text{if len(sources)} \geq 3, \\ 0, & \text{otherwise,} \end{cases}$$

$$r_{\text{length}} = \begin{cases} -1, & \text{if len(final\_report)} > 1500 \text{ characters,} \\ 0, & \text{otherwise,} \end{cases}$$

$$r_{\text{final}} = r_{\text{sources}} + r_{\text{length}}.$$

Interpretation

- T1, T2, T3: many sources and well-structured reports $\Rightarrow r_{\text{final}} \approx 1$ or 0.
- T4: few sources and poor content $\Rightarrow r_{\text{final}} \approx 0$ or negative.
- T5: enough sources (often random) but long inaccurate report $\Rightarrow r_{\text{final}}$ often 0 or -1.

Potential Use Over Time

- Track average reward per query type.
- Adjust prompts to:
  - Increase emphasis on "do not fabricate when sources are weak" to raise rewards on T4/T5-style queries.
  - Encourage more concise responses to avoid length penalties.
- Even without automatic policy updates, this framework shows how the system could be iteratively improved with RL-inspired feedback.

6 Limitations and Future Work (Evaluation Perspective)

Hallucination on Nonsense Inputs

- T5 demonstrated that the system freely invents content for non-existent concepts.
- Future work: add rules such as

  > *If you cannot find at least N credible sources that mention the main concept, explicitly state that the topic might be nonsense and stop.*

No Explicit Uncertainty Modelling

- For speculative topics (T3), the system sounds more certain than the underlying evidence supports.
- Future work: require the AnalysisAgent to classify claims as low / medium / high certainty and propagate this to the WriterAgent.

Vague Query Handling

- For T4, the system does not ask the user to clarify.
- Future work: add controller logic such that if the query is too short or ambiguous, the ControllerAgent returns a clarifying question instead of executing the full pipeline.

Limited Use of RL Feedback

- Rewards are currently computed conceptually but not used to automatically tune prompts.
- Future work: implement a small script that:
  - Reads logged rewards.
  - Adjusts parameters (e.g., max tokens, number of sources) or flags problematic prompts for manual editing.

7 Conclusion

The Agentic Research Assistant with RL Feedback successfully demonstrates:

- Multi-agent orchestration with a central controller.
- Integration of multiple built-in tools and a custom citation tool.

- Use of memory and RL-style reward logging for iterative improvement.
- Practical usefulness for realistic research questions (T1–T3).

At the same time, evaluation on edge cases (T4, T5) exposes important limitations—especially hallucinations and lack of query validation—which are exactly the kinds of issues modern agentic systems must address. These results, plus the proposed future work, collectively fulfill the assignment requirements for design, implementation, evaluation, and critical reflection.