Error Audit

Error 429: Too Many Requests	Users: Students, researchers, professionals conducting extensive literature searches
Error type: System limitation - The system can't provide answers due to external API constraints (NewsAPI, arXiv rate limits)	User stakes: High
Error:Irrelevant Research Results	Users: All user types (students, researchers, professionals)
Error type: Context Limitation - System is "working as intended" but keyword extraction failed to disambiguate context - Poor assumptions about user intent	User Stakes : High
Error: Hallucinated Citations	Users: All user types (students, researchers, professionals)
Error type: Background - Severe credibility damage - Academic integrity violations if citations are used	User Stakes : High
Error: Outdated Information Presented as Current	Users: Researchers, professionals requiring current information
Error type: Background - System isn't working correctly but doesn't register an error - Data pipeline hasn't updated recently enough	User stakes: High
Error: Ambiguous Query Interpretation	Users: Users with domain-specific terminology that has multiple meanings

Error type: Context Limitation - System lacks context to disambiguate query intent	User stakes: Medium
Error: Vector Database Retrieval Failure	Users: All users during peak usage or system maintenance
Error type: System limitation - Infrastructure failure in vector database layer	User stakes: High
Error: Slow Response Time (>5 seconds)	Users: All users, especially those on slower connections
Error type: System limitation - LLM inference latency, multiple retrieval calls, or network issues	User stakes: Low

Error Sources Analysis

Error 1: API Rate Limit Exceeded

Failure state:

- Is your feature unusable as the result of multiple errors?
  If multiple APIs fail simultaneously, our system could become unusable. We currently query two API's, NewsAPI and arXiv API, to retrieve data. While a failure in one does not necessarily impact the other, if both APIs fail at the same time, the system will be unable to function.

Error 2: Irrelevant Research Results

Relevance error signals:

- Is the model lacking available data or requirements for prediction accuracy?
  Yes, there are two possible reasons this could occur:
  1. The model may not have enough data available to answer the question.
  2. The retrieval component of the RAG model may not be correctly fetching all the relevant document chunks.

Error 3: Outdated Information

Relevance error signals:

- Is the model lacking available data or requirements for prediction accuracy?
  Yes, there are two possible reasons this could occur:
  1. The model may not have enough data available to answer the question.
  2. The retrieval component of the RAG model may not be correctly fetching all the relevant document chunks.

Error 4: Hallucinated Citations

Input error signals:

- Did the model improperly weigh a user action or other signal? No - the LLM produced incorrect information because the RAG system retrieved inaccurate data from the input source.

Relevance error signals:

- Is the model lacking available data or requirements for prediction accuracy?
  Yes, there are two possible reasons this could occur:
  1. The model may not have enough data available to answer the question.
  2. The retrieval component of the RAG model may not be correctly fetching all the relevant document chunks.

Error 5: Vector Database Failure

System hierarchy error:

- Is your user connecting your product to another system?
  Yes, since our system relies on the FAISS vector database to store word embeddings, any failure in this dependency would also cause our system to fail.

Error 6: Ambiguous query interpretation

Input Error Signals

- Did the user anticipate the auto-correction of their input into an AI system?
  No clear expectation. User typed "Apple research" without specifying domain context. The user likely didn't realize the term has multiple meanings in different contexts (technology company vs. fruit).
  The system attempted to interpret intent without explicit user guidance, leading to mixed results.
- Did the model improperly weigh a user action or other signal?
  YES - This is a context error. The keyword extraction module treated "Apple" as equally likely to refer to the company or the fruit, without considering:

    - User's browsing history (if they typically search AI/ML topics)
    - Session context (previous queries in the conversation)
    - Domain-specific priors (in an AI research tool, "Apple" more likely refers to Apple Inc.)
    - The model failed to use available contextual signals to disambiguate.

Error 7: Slow response time

Relevance Error Signals

- Is the model receiving unstable or noisy data?
  Potentially contributing factor:
    - If external APIs (NewsAPI, arXiv) are slow or timing out, this cascades to user experience
    - Network latency to vector database or blob storage
    - Unstable infrastructure (database connection pools exhausted, CPU throttling)

System Hierarchy Error

- Is your user connecting your product to another system, and it isn't clear which system is in charge?
  YES - Multiple system dependencies create latency:
    - External APIs (NewsAPI, arXiv, Google Scholar) - each can add 1-3 seconds
    - Vector database (FAISS/Pinecone) queries - 500ms-2s depending on index size and load
    - LLM inference (OpenAI API or self-hosted) - 2-4 seconds for complex generations
    - Sequential dependencies: keyword extraction → retrieval → LLM generation
- The user experiences the sum of all system latencies, but doesn't know which component is slow.
- No transparency about what's happening during the wait.

Error Resolution Plans

Error Rationale	Solution Type	Error Resolution
Error 1: API Rate Limit Exceeded User expects immediate responses but external API limits prevent data retrieval	– User control - Display clear messaging about rate limits. – Feedback - Allow users to retry after cooldown	**User Path:** 1. User submits query 2. System detects rate limit

		3. System processes query when API available 4. The user receives a response with notification: "Thank you for waiting!" **Opportunity for Model Improvement:** ● Use multiple API keys ● Log rate limit patterns to predict and prevent future occurrences
Error 2: Irrelevant Research Results User receives results from wrong domain because keyword extraction lacks context disambiguation	– Feedback - Allow users to refine queries and provide relevance feedback – User control - Offer domain filters (AI/ML, computer vision, NLP, etc.)	**User Path:** 1. User submits ambiguous query 2. System detects potential ambiguity 3. System returns relevant filtered results 4. User can provide feedback and ask the correct query if results are not accurate. **Opportunity for Model Improvement:** ● Implement user history and preference learning ● Use conversation context from chat history ● Log ambiguous queries and user

		selections for training data
Error 3: Outdated Information User expects current information but data pipeline hasn't refreshed recently enough	Feedback ● Display data freshness timestamps ● Automated data pipeline monitoring and alerts	**User Path:** 1. User submits query about current topic 2. System retrieves best available data 3. User sees response with timestamp: "Based on sources up to [date]. Last updated: 2 days ago" 4. User queries for latest data 5. System triggers priority data refresh for that topic 6. User gets latest data **Opportunity for Model Improvement:** ● Implement continuous data ingestion (near real-time) ● Develop topic-based refresh prioritization (hot topics updated more frequently) ● Monitor data staleness by topic area ● Log user requests for fresh data to identify priority topics ● Set up automated alerts when knowledge base falls behind threshold (e.g.,

		>72 hours for AI news)
Error 4: Hallucinated Citations LLM generates convincing but false citations, undermining system credibility	• Other: RAG grounding enforcement, citation verification system	**User Path:** 1. User submits query 2. System retrieves relevant documents from vector DB 3. LLM generates response ONLY from retrieved documents 4. Every citation includes: source publication, date and other metadata 5. User can verify each citation **Opportunity for Model Improvement:** ● Enforce strict RAG-only responses (no knowledge outside retrieved docs) through prompt engineering ● Add citation validator that cross-references with original sources ● Log any detected hallucinations for model fine-tuning ● Regular audits
Error 5: Vector Database Retrieval Failure	• Other: Redundancy, monitoring, graceful degradation	**User Path:** 1. User submits query 2. System detects vector DB unavailability

Critical infrastructure component fails, making feature completely unusable		3. System sends alert to ops team 4. User receives degraded but functional response 5. Notification when full service restored **Opportunity for Model Improvement:** ● Implement vector DB replication and failover ● Set up health monitoring with automatic alerts (<99.5% uptime triggers page) ● Create backup retrieval systems (PostgreSQL full-text search, Elasticsearch) ● Implement connection pooling and retry logic with exponential backoff ● Log all failures for infrastructure improvement ● Regular load testing to identify capacity limits
Error 6: Ambiguous Query Interpretation User uses terminology with multiple meanings, system lacks context to choose correctly	Feedback: ● Interactive clarification	**User Path:** 1. User submits "Apple research" 2. System detects ambiguity (confidence: low)

		3. If the results are undesirable, user queries with clarification again 4. System remembers preference for session unless user states otherwise **Opportunity for Model Improvement:** ● Use conversation history for context ● Implement query expansion/refinement suggestions ● Track ambiguous term resolution patterns ● Fine-tune query understanding model on user selections
Error 7: Slow Response Time Users expect sub-5-second responses but system exceeds latency target	● Feedback - Progress indicators with estimated time ● Other: Performance optimization, caching	**Opportunity for Model Improvement:** ● Implement query result caching ● Optimize vector similarity search (approximate nearest neighbors) ● Parallelize API calls and retrieval operations ● Pre-compute embeddings during ingestion, not at query time ● Implement streaming responses (show results as they arrive)

		• Use faster embedding models for initial retrieval, detailed models for re-ranking • Log slow queries for performance analysis

2. Quality Assurance Plan

Goal: Ensure research sources are output with timestamps	**Review frequency**: Weekly - Manual spot checks of source dates
Method: - User feedback: "Was this information current?" - Comparison of system responses to latest publications	
Start date: Week 1 of deployment Review / End date: Ongoing	
Goal: <3 second average response time, 99.5% uptime	**Review frequency**: Daily - System performance metrics - Monthly - Deep dive user research interviews
Method: In-product metrics: response time, error rate, query success rate - In-product surveys: Post-interaction "How helpful was this?" - Quarterly diary studies with 10 power users - A/B testing of UX improvements	
Start date: Week 2 of deployment Review / End date: Ongoing	

Goal: 100% citation accuracy - every citation must be verifiable	**Review frequency**: Weekly - Manual verification of flagged citations
Method: User reporting: "Citation not found" - Weekly manual audit: randomly select 50 responses and verify every citation - Customer reports of citation issues	
Start date: Week 1 of deployment Review / End date: Ongoing	
Goal: Identify and fix edge cases before they impact >1% of users	**Review frequency:** Weekly - Review error logs and user reports Monthly - Dedicated edge case testing session
Method: Anomaly detection in query patterns - Error log analysis (group by error type, frequency, user impact) - User research interviews specifically about error experiences - Proactive testing of identified edge cases	
Start date: Week 3 of deployment Review / End date: Ongoing	

Error Reporting Sources (Comprehensive Monitoring)

We will monitor errors through:

1. **Customer Service Reports**
    - Escalation of critical issues
    - Weekly summary of top reported issues
2. **In-Product Metrics** (Automated Dashboards)
    - Error rate
    - Response time distribution
    - Query success/failure rates
    - Usage analytics
3. **In-Product Surveys**

    - Post-query satisfaction ("Was this helpful?")
    - Periodic NPS surveys
    - Feature-specific feedback forms
    - Exit surveys for churned users
4. **User Research**
    - Monthly user interviews (5-10 users)
    - Quarterly diary studies with power users
    - Usability testing of new features
    - Academic partner feedback sessions
5. **System Monitoring**
    - Infrastructure health (uptime, latency, errors)
    - Data pipeline status
    - API rate limit tracking
    - Vector DB performance metrics