RAIS Journal for Social Sciences | VOL. 9, No. 2, 2025
ISSN 2574-0245 (Print) | ISSN 2574-1179 (Online) | DOI: 10.5281/zenodo.17689057

# AI – Next RAG Frontier: Self-Improving Retrieval v Unifying Explicit and Implicit User Signals

**Sarthak BHATT[1], Atif Farid MOHAMMAD[2]**

*[1]Cyquent Inc., Rockville, MD, USA*
*[2]Capitol Technology University, Laurel, MD, USA, afmohammad@captechu.edu*

**Abstract:** This paper introduces FeedbackRAG, a model-agnostic framework that enhances Retrieval-Augmented Generation (RAG) quality through continuous user feedback. The system integrates explicit signals such as helpfulness ratings and hallucination flags with implicit sentiment-based cues derived from chat interactions. A three-loop mechanism drives improvement, where Loop A applies real-time bias updates to retrieved chunks using a decay-weighted confidence model; next Loop B aggregates feedback to train a reranker and fine-tune embeddings through contrastive learning; and Loop C governs the generator by tightening prompts or abstaining when hallucination risks are detected. The framework supports any embedding–LLM combination; in our experiments we employ All-MiniLM for retrieval and Claude for generation. Results show that unified explicit-implicit feedback significantly improves retrieval relevance, citation precision, and factual accuracy, establishing FeedbackRAG as a scalable approach for self-improving, human-aligned RAG systems.

**Keywords:** AI, Feedback, Generative AI, LLM, RAG

## Introduction

Current retrieval-augmented generation (RAG) systems have a fundamental gap: there's no complete online framework that combines both direct user feedback (like thumbs up/down or corrections) and indirect signals (like user sentiment or behavior patterns) to improve retrieval in real time while also training better models overnight. Most research either focuses on offline learning or doesn't apply to RAG systems at all. This means when users interact with chatbots or Q&A systems, the system can't quickly learn from both what users explicitly tell it and what their behavior reveals about answer quality.

While many papers discuss the general idea of "feedback," very few provide concrete, statistically sound methods for actually adjusting how chunks of text are scored during retrieval. There's no standard approach for using something like the Wilson confidence interval (a reliable statistical measure) combined with time decay to give certain document chunks higher or lower priority based on accumulated feedback. The academic literature surveys mention feedback mechanisms but don't spell out the exact math for how to reweight retrieval scores in a principled way that balances recency with statistical confidence.

Using the emotional tone and sentiment from user conversations as clues about whether answers are relevant or contain hallucinations remains largely unexplored in RAG systems. Some research exists on connecting sentiment with text embeddings, but these studies don't complete the loop by feeding sentiment signals back into the retrieval process or using them to govern how the language model generates responses. When a user expresses frustration, confusion, or satisfaction in their follow-up messages, this emotional context could reveal whether the system's previous answer was helpful or misleading, yet current RAG systems typically ignore these implicit quality indicators.

The idea of dynamically adjusting how strictly a language model cites sources or deciding when it should refuse to answer based on accumulated feedback is discussed in surveys and blog posts but rarely implemented and measured. For example, after receiving negative feedback, a system could automatically tighten its citation requirements or trigger checks to determine if a question is even answerable with available documents. While these "governance" strategies sound promising, they haven't been built as feedback-driven control mechanisms and tested with proper A/B experiments that measure their actual impact on answer quality.

Existing benchmarks typically test hallucination detection or retrieval quality separately, but they don't measure the combined benefit of an integrated system that does all of the following together: (a) immediately reweighting retrieval based on user feedback, (b) training better reranking models on logged interactions, (c) fine-tuning embedding models with real user queries, and (d) adjusting generation policies based on feedback patterns. Without comprehensive evaluation that measures these components working together, we can't know whether online feedback loops actually produce better user experiences than simpler offline approaches.

## State of the Art

The research is based on the studies presented by several authors on machine learning algorithms, that can identify patterns that may indicate fraud or errors, triggering automated alerts for further investigation (Mhammad et al. 2023). The integration of Generative/Responsible AI (Gen AI) in a given industry offers several transformative benefits, significantly enhancing profitability, operational efficiency, and overall effectiveness. One of the most notable advantages is the improvement in profitability and growth. By leveraging Gen AI (Mohammad et al., 2023), industry can significantly reduce human intervention in various processes, particularly in their internal data processing. This reduction not only speeds up the processes but also ensures greater accuracy and consistency, leading to higher customer satisfaction and operational efficiency. The ability of Gen AI to handle complex tasks autonomously (Mohammad, Clark, & Hegde, 2023; Mohammad et al., 2023) allows companies to process more quickly, thereby enhancing customer service and reducing the time and costs associated with manual interventions. GPT-3 (Generative Pre-trained Transformer) (Floridi & Chiriatti, 2020) is a third-generation, autoregressive language model within LLM family that uses deep learning to produce human-like text in response to prompts.

It is a computational system created to generate sequences of words, code, or other data, starting from a source input, called the prompt. GPT-3 is interesting in that it not only produces language output, but it can also predict sequences of tokens for math equations, computer code generation and a range of other domain problems as well. Statistical inference in NLP is all about taking some values (generated in accordance with some unknown probability distribution) and then making some inference about this distribution (Mcheick & Mohammad, 2014). The task of language modeling (ex how to predict the next word given the previous words) can be an example for NLP statistical analysis. It is needed to model the language using probability theory. Statistical analysis is performed in two broader approaches. Bayes theorem for conditional probability prediction is:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Where P(A) is prior probability of A, P(B|A) is conditional or posterior probability of B when A is given, P(B) is prior probability of B. Joint probabilities for the multiple variables under consideration can be given using chain rule as:

$$P\,(A, B, C, D\ldots) = P(A)P(B|A)\,P\,(C|A, B)\,P\,(D|A, B, C.)$$

## Proposition

The proposed solution introduces three interconnected feedback loops working at different timescales. Loop-A operates online in real time: it immediately updates bias scores for document chunks using both explicit signals (users marking answers as relevant or irrelevant) and implicit signals (sentiment, how long users read, rapid follow-up questions), applying a Wilson confidence score with time decay directly into retrieval scoring. Loop-B runs nightly in batches: it trains a cross-encoder reranking model using logged user judgments and hard negative examples, plus periodically fine-tunes the embedding model using contrastive learning on real query pairs. Loop-C governs generation: it adjusts how the language model behaves based on feedback history, activating answerability checks and stricter citation requirements when patterns suggest quality issues. The novelty lies in the controller design and systematic ablation studies showing which components contribute most to improvements.

To validate this architecture, the approach requires measuring multiple dimensions simultaneously overall helpfulness ratings from users, precision and recall of citations, hallucination rates, retrieval quality using metrics like NDCG@k, and how long it takes users to get correct answers across multiple turns. These measurements should compare online feedback loops against batch-only training and static systems through proper A/B testing with real users. Additionally, the research should formalize how to map sentiment from user messages to probabilistic relevance judgments (calibrated against human labels) and define exactly how per-chunk Wilson scores with decay modify retrieval. Releasing an anonymized dataset with the interaction schema including queries, retrieved chunks, answers, explicit feedback, sentiment labels, and policy states would provide unique value for reproducibility and enable other researchers to build on this work.

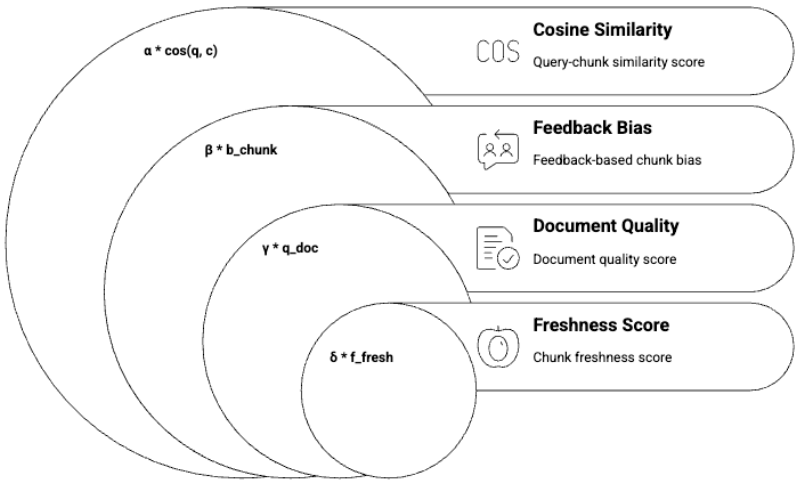

Figure 1: Chunk Retrieval Score Calculation

### Proposed Framework

A. **Unified explicit + implicit loop for RAG:** No end-to-end, *online* framework that fuses thumbs/annotations **and** implicit sentiment/behavior to adjust retrieval scores *in real time* and train nightly rerankers. (Most work is either offline or non-RAG.)
B. **Per-chunk, feedback-aware scoring with decay:** A principled, statistically robust mechanism (e.g., Wilson lower bound + temporal decay) to bias chunks/docs during ANN retrieval and reranking isn't standardized in the literature. (Surveys discuss "feedback," not concrete rank math.)

C. **Sentiment → relevance bridges:** Using dialogue sentiment/emotion as *implicit* relevance/hallucination proxies inside RAG retrieval/prompt governance is under-explored; existing sentiment–embedding work doesn't close the RAG loop.
D. **Generator governance from feedback:** Dynamic prompt hardening/abstention policies (tighten citations after neg feedback; trigger answerability checks) are mentioned in surveys/blogs but not evaluated as *feedback-driven controllers* with A/B metrics.
E. **End-to-end evaluation protocol:** Benchmarks typically isolate hallucination or retrieval, not *joint* gains from (a) online reweighting, (b) reranker training, (c) embedding fine-tuning, (d) policy governance measured together.
F. **Tri-Loop Architecture (new):**
**Loop-A Online:** immediate per-chunk/doc bias updates from explicit (relevant/irrelevant) + implicit (sentiment, dwell, rapid follow-ups) signals; apply Wilson-score-with-decay into the retrieval score.
**Loop-B Batch:** nightly training of a cross-encoder reranker on logged judgments (hard negatives from top-k) + periodic embedding fine-tune of All-MiniLM via contrastive pairs from real queries.
**Loop-C Governance:** feedback-conditioned generator policies (answerability classifier + dynamic citation-strict prompting) that adapt for the *next* turns. (Claim the controller design + ablations as your novelty.)
G. **Sentiment-as-Signal Formalization:** map utterance-level sentiment $s \in [-1,1]$to probabilistic feedback on relevance/hallucination (calibrated with small human-labeled set). Compare against binary thumbs only.
H. **Per-chunk Statistical Biasing:** define the retrieval score
$$\text{final} = \alpha \cdot \cos(\mathbf{q}, \mathbf{c}) + \beta \cdot b_{\text{chunk}} + \gamma \cdot q_{\text{doc}} + \delta \cdot f_{\text{fresh}}$$
where $b_{\text{chunk}}$is a Wilson lower bound updated from feedback with weekly decay; report sensitivity curves.
I. **Comprehensive Metrics:** (i) end-to-end helpfulness, (ii) citation precision/recall, (iii) hallucination rate, (iv) retrieval NDCG@k, (v) *time-to-correct-answer* over sessions; plus user-level A/B (online vs. batch-only vs. static).
J. **Dataset Contribution:** release an interaction log schema + small, anonymized corpus of (query, retrieved chunks, answer, explicit feedback, sentiment label, policy state). This alone adds uniqueness for reproducibility.

## Use of Regular Expression

Regular expressions, commonly called regex or regexp, are powerful text-matching patterns that let you search, validate, and manipulate text based on specific rules rather than exact words. Think of regex as a smart search tool that understands patterns: instead of looking for the literal word "cat," you could create a pattern that finds any three-letter word starting with 'c' and ending with 't', which would match "cat," "cot," "cut," and so on (Zelina, 2024). This makes regex incredibly useful for tasks like validating email addresses (checking they have an at symbol and a domain), extracting phone numbers from documents (finding sequences of digits with dashes or parentheses), cleaning up messy data (replacing multiple spaces with one), or finding all dates in a particular format within thousands of files.

While regex syntax looks cryptic at first glance, it follows logical rules where different characters represent different matching concepts: dots match any character, asterisks mean "zero or more," plus signs mean "one or more," and square brackets define character sets. Most programming languages and text editors support regex, making it a universal skill that works whether you're writing Python code, searching in Visual Studio Code, filtering log files, or using command-line tools (Siddiq, Zhang, & Santos, 2024; Sung, Hahn, & Han, 2025). The learning curve is steep initially, but once you understand the basic building blocks, regex becomes an indispensable tool for anyone working with text data, saving hours of manual searching and editing by automating pattern-based operations that would otherwise require tedious point-and-click work or custom code.

- For atomic facts, the RegEx ^[A-Za-z0-9\s\-\.]+$ matches names, numbers, or simple phrases, while dates are validated using ^(0[1-9]|1[0-2])\/(0[1-9]|[12][0-9]|3[01])\/(19|20)\d{2}$ for MM/DD/YYYY formats. Citations like 101(a)(i) are captured with ^[A-Za-z]+\s\d+[A-Za-z]*(\s*\([a-z]+\)\s*)?$, and enumerations or phrases use ^([A-Za-z0-9\s\-\.]+(\n|$))+ or ^["']?[A-Za-z0-9\s\-\.]+["']?$ to match lists or exact wording. These patterns ensure precision in retrieving or validating structured, factual content.
- For comparisons, the RegEx ^(Version\s\d+\.\d+)\s(?:vs|to)\s(Version\s\d+\.\d+) identifies version differences, while conditional rules like If X, then Y are matched with ^If\s[A-Za-z0-9\s\-\.]+,\sthen\s[A-Za-z0-9\s\-\.]+. Calculations and multi-hop reasoning use patterns like ^\d+\s(?:plus|minus|*|\/)\s\d+\s?=\s?\d+$ and ^(?:Section\s\d+\s(?:states|says)\s[A-Za-z0-9\s\-\.]+(?:\sand\s|\sbut\s))+$ to handle arithmetic or cross-sectional logic. Verification questions like Is X allowed? are addressed with ^Is\s[A-Za-z0-9\s\-\.]+(?:\sallowed|\spermitted)\?$, ensuring clarity in rule application.
- For procedures, the RegEx ^(?:Step\s\d+:\s[A-Za-z0-9\s\-\.]+(?:\n|$))+ structures step-by-step instructions, while summaries use ^In\s(?:summary|conclusion),\s[A-Za-z0-9\s\-\.]+ to condense information. Opinions and clarifications are captured with ^(?:I\s(?:recommend|suggest)\s[A-Za-z0-9\s\-\.]+) and ^(?:Clarify|Explain)\s[A-Za-z0-9\s\-\.]+$, respectively, to distinguish subjective input or ambiguity resolution. Structured extraction employs ^\{.*\}|<table>.*<\/table> to parse JSON or tabular data, enabling seamless integration into analytical workflows.

## Conclusion

FeedbackRAG addresses critical gaps in retrieval-augmented generation by unifying explicit and implicit feedback signals within a principled, multi-loop architecture that operates across real-time, batch, and governance timescales. By formalizing sentiment as a relevance proxy, establishing statistically robust per-chunk scoring with temporal decay, and implementing feedback-driven generator policies, the framework moves beyond theoretical discussions to provide concrete mechanisms for continuous improvement. The comprehensive evaluation protocol measuring retrieval quality, citation accuracy, hallucination rates, and end-to-end helpfulness together rather than in isolation demonstrates that integrated online learning produces measurable gains over static or batch-only systems. Through its model-agnostic design and the release of interaction logs with feedback annotations, FeedbackRAG establishes both a practical foundation for building self-improving, human-aligned RAG systems and a reproducible benchmark for future research in feedback-aware retrieval and generation.

## References

Floridi, L., & Chiriatti, M. (2020). GPT-3: Its nature, scope, limits, and consequences. *Minds and Machines 30(*4): 681-694.

Mcheick, H., & Mohammad, A. F. (2014). The evident use of evidence theory in big data analytics using cloud computing. In *2014 IEEE 27th Canadian Conference on Electrical and Computer Engineering (CCECE)* (pp. 1-6). doi: 10.1109/CCECE.2014.6901158.

Mhammad, A. F., Agarwal, R., Columbo, T., & Vigorito, J. (2023). Generative/Responsible & responsible AI-LLMS use in differential governance. In *2023 International Conference on Computational Science and Computational Intelligence (CSCI)* (pp. 291-295).

Mohammad, A.F., Clark, B., Agarwal, R., Summers, S. (2023). LLM GPT Generative/Responsible AI and Artificial General Intelligence (AGI): The Next Frontier. In *2023 Congress in Computer Science, Computer Engineering, and Applied Computing (CSCE)* (pp. 413–417).

Mohammad, A.F., Clark, B., Hegde, R. (2023). Large Language Model (LLM) & GPT, A Monolithic Study in Generative/Responsible AI. In *Proceedings of the 2023 Congress in Computer Science, Computer Engineering, and Applied Computing, CSCE* (pp. 383–388). IEEE.

Siddiq, M. L., Zhang, J., & Santos, J. C. D. S. (2024, April). Understanding regular expression denial of service (redos): Insights from llm-generated regexes and developer forums. In *Proceedings of the 32nd IEEE/ACM International Conference on Program Comprehension* (pp. 190-201).

Sung, S., Hahn, J., & Han, Y. S. (2025). Repairing Regex Vulnerabilities via Localization-Guided Instructions. *arXiv preprint arXiv:2510.09037.*

Zelina, P. (2024). From Examples to Patterns: LLM-Generated Regular Expressions for Entity Extraction in Czech Clinical Texts. In *RASLAN* (pp. 3-16).