RESEARCH ARTICLE OPEN ACCESS

Feedback Refine: User-Guided Refinement of LLM-Generated Professional Documents

C. Sreerag*, Prof. Dr. Shivani Budhkar**
*(Department of Master of Computer Applications,
Progressive Education Society's Modern College of Engineering, Pune, India
Email: c.sreerag17@gmail.com)
**(Department of Master of Computer Applications,
Progressive Education Society's Modern College of Engineering, Pune, India
Email: shivanibudhkar@gmail.com)

Abstract—

Most AI writing tools generate documents in a single pass, forcing users to spend hours manually editing the output. While newer systems attempt self-improvement through internal feedback, they ignore what users actually want to change. This paper proposes FeedbackRefine, a conceptual framework that lets users guide document improvement through iterative feedback cycles. The key idea is simple: generate an initial draft, ask the user for feedback, use that feedback to improve the document, and repeat until the quality is acceptable. FeedbackRefine operates in five stages: (1) initial draft generation, (2) quality evaluation against a multi-dimensional rubric, (3) user feedback collection, (4) feedback transformation into refinement instructions, and (5) targeted document refinement. This framework addresses a clear gap in current research: existing systems either produce one-shot documents or rely on internal AI feedback, but rarely incorporate actual user input to guide refinement. We propose evaluation methods to test whether user-guided iteration reduces editing time and improves document quality compared to standard approaches. This work contributes a conceptual foundation for building better human-AI document creation systems and identifies key research questions for future empirical validation.

Keywords — Large Language Models, Human-in-the-Loop AI, Iterative Refinement, Document Generation, User Feedback, Professional Writing, AI-Assisted Content Creation

I. Introduction

Large Language Models (LLMs) such as GPT-4 and LLaMA have demonstrated remarkable capabilities in generating coherent, structured text across a wide range of tasks [4], [8]. In professional settings, these models are increasingly being explored for generating documents such as policy briefs, grant proposals, project plans, and technical reports [2].

Despite this promise, a critical gap remains. Most current systems operate in a *one-shot* manner: the user provides a prompt, the LLM generates a single draft, and the user manually revises the output. This is time-consuming and cognitively demanding. More recent work has explored *self-refinement*, where the LLM critiques and revises its own output iteratively [1]. While useful for short tasks, self-refinement cannot incorporate domain expertise, personal preferences, or specific user requirements.

This paper proposes FeedbackRefine, a conceptual framework designed to bridge this gap by integrating real user feedback into an iterative document refinement pipeline. This shifts document generation from a one-shot automated task to a *collaborative, human-guided process*.

Key Contributions:

- Identification of a research gap in user-feedback integration for iterative LLM document generation.
- A conceptual five-stage FeedbackRefine framework with defined feedback types and transformation mechanisms.
- A multi-dimensional quality rubric tailored to professional documents (policy briefs, proposals).
- A system architecture design with two supporting diagrams.
- A proposed experimental protocol with testable hypotheses for future empirical validation.

Scope: This is a conceptual/propositional paper. No empirical results are reported. We propose an architecture and evaluation plan for future implementation.

II. Related Work

A. Self-Refinement and Iterative Generation

Self-Refine [1] by Madaan et al. uses a single LLM as generator, critic, and refiner. While effective on short outputs, it cannot incorporate external user preferences.

LLMRefine [10] extends this with span-level feedback, identifying specific error regions for targeted refinement. This approach remains fully automated without human input.

A²R [3] by Lee et al. uses evaluation metrics to auto-generate refinement feedback. Its structured assessment informs our quality rubric design, though it does not incorporate real user input.

DeCRIM [9] decomposes instructions into constraints, critiques compliance, and refines violations. This structured critique approach directly informs our feedback transformation mechanism.

B. Document Generation

Sci2Pol [6] introduces the first dataset for science-to-policy brief generation. However, it focuses on single-pass generation without a refinement loop. **IMPROVE [11]** applies iterative component-level refinement for ML pipeline design, inspiring our section-level strategy.

C. Research Gap

Table I summarizes how FeedbackRefine compares to prior approaches. The key gap is that no existing system supports systematic, multi-modal *user feedback integration* for long-form professional document refinement with convergence control.

TABLE I
COMPARISON OF ITERATIVE REFINEMENT APPROACHES

System	User Feedback	Iterative	Long Docs	Convergence Criteria
Self-Refine [1]	✗	✓	✗	✗
LLMRefine [10]	✗	✓	✗	✗
A²R [3]	✗	✓	✗	✗
DeCRIM [9]	✗	✓	✗	✗
Sci2Pol [6]	✗	✗	✓	✗
IMPROVE [11]	✗	✓	✓	✗
FeedbackRefine (Ours)	✓	✓	✓	✓

III. SYSTEM ARCHITECTURE AND METHODOLOGY

A. High-Level Architecture

FeedbackRefine is conceptualized as a four-layer architecture (Fig. 1):

- **User Layer:** The professional user provides document requirements and raw content, and supplies iterative feedback throughout the process.
- **Feedback Processing Layer:** Receives multi-modal user feedback (natural language, scores, annotations) and transforms it into structured prompt instructions.
- **LLM Inference Layer:** Interfaces with the language model for both initial draft generation and targeted section refinement.
- **Quality Evaluation Layer:** Scores each draft against the multi-dimensional rubric and determines whether convergence has been reached.

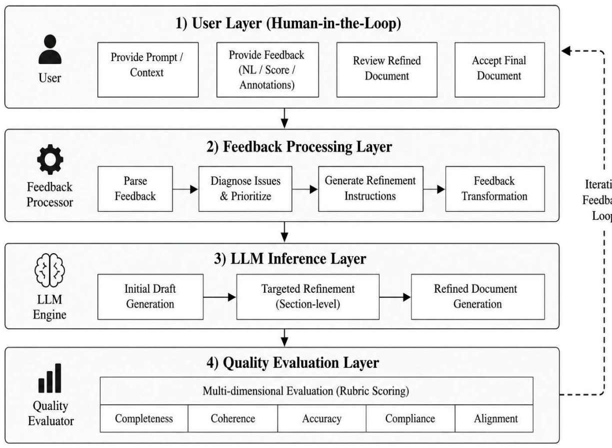

Fig. 1. FeedbackRefine: Four-Layer System Architecture.

B. Five-Stage Iterative Pipeline

Fig. 2 presents the detailed iterative pipeline that executes within the architecture above.

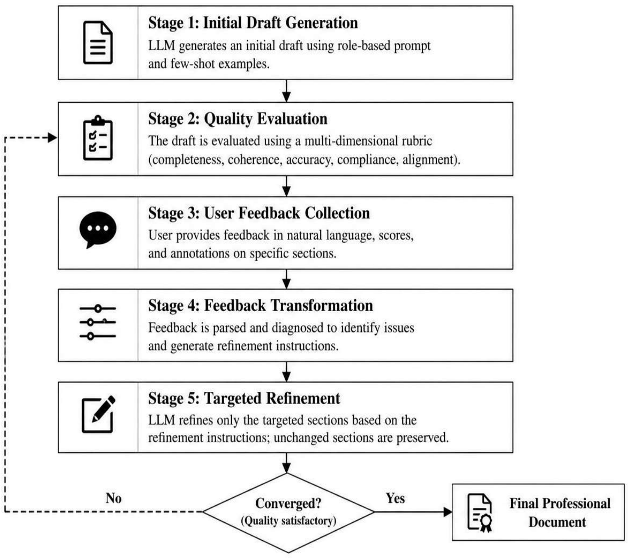

Fig. 2. FeedbackRefine: Five-Stage Iterative Refinement Pipeline.

C. Feedback Types and Quality Rubric

FeedbackRefine accepts three feedback modalities: **(A)** Natural language directives (e.g., “Expand rural healthcare statistics”), **(B)** Scalar quality scores triggering targeted improvement, and **(C)** Inline annotations on specific document passages.

Table II defines the five-dimensional quality rubric used for evaluation at each iteration.

TABLE II
MULTI-DIMENSIONAL QUALITY RUBRIC (SCALE 1–5)

Dimension	Description
Completeness	All required sections present and adequately detailed
Coherence	Logical flow and smooth transitions between sections
Factual Consistency	Claims grounded in provided data; no hallucinations

Domain Compliance	Follows professional conventions and appropriate tone
Stakeholder Alignment	Key stakeholder perspectives fairly represented
Overall Score	**Mean of above five dimensions (threshold: 4.0/5.0)**

D. Convergence Detection

Refinement stops when any of the following are met: (1) overall rubric score ≥ 4.0, (2) quality improvement < 0.1 for two consecutive iterations (plateau), (3) user explicitly terminates, or (4) a preset maximum iteration limit is reached (default: 10).

IV. Implementation Challenges and Solutions

Realizing FeedbackRefine as a working system introduces several non-trivial challenges. We outline anticipated problems and proposed mitigation strategies below.

Hallucination Accumulation: Multiple LLM calls risk compounding factual errors across iterations. Each refinement prompt should include the original user-provided facts as a grounding reference, and a post-hoc fact-checking step should flag unsupported claims before presenting any draft to the user [7].

Feedback Ambiguity: User feedback can be vague (e.g., "make it better"). The Feedback Processing Layer must include an intent-classification step to detect ambiguity and, when detected, prompt the user with clarifying questions rather than attempting a blind refinement pass.

Context Window Limitations: Long professional documents (5,000+ words) may exceed LLM context limits. Section-level refinement (operating on individual document sections rather than the full text) mitigates this by keeping each prompt well within token limits.

User Cognitive Load: Requiring structured feedback at every iteration may burden users. The system should offer a simplified feedback interface with pre-defined options (e.g., "Expand this section", "Fix accuracy") alongside free-text input.

External API Dependency: Relying on a third-party LLM API introduces availability and pricing risks. Future iterations should evaluate locally hosted open-source models (e.g., LLaMA) as alternative inference backends to improve reliability and strengthen data privacy [2].

V. Future Evaluation Methodology

As FeedbackRefine represents a conceptual architecture, empirical validation is deferred to a future implementation phase. We propose the following evaluation plan.

A. Research Hypotheses

- H1: User-guided iterative refinement —

H1: *may* produce higher rubric scores than one-shot generation.

H2: FeedbackRefine *may* reduce the number of manual edits required post-generation.

H3: User-provided feedback *may* yield greater improvement than automated self-refinement.

H4: Documents *may* converge to acceptable quality within 3–5 iterations.

B. Proposed Metrics and Baselines

Table III lists proposed evaluation metrics. Baselines include: **B1** one-shot LLM generation, **B2** Self-Refine [1] (automated self-critique), and **B3** expert manual authoring (quality gold standard).

TABLE III
PROPOSED EVALUATION METRICS

Category	Metric	How Measured
Content Quality	Rubric Score (1–5)	Per-dimension assessment
	Factual Accuracy	vs. source data
Efficiency	Manual Edit Count	Edits post-generation
	Iterations to Converge	Cycles to threshold
User Satisfaction	Expert Rating (1–5)	Domain expert panel
	Willingness-to-Use	Yes/No adoption survey

C. Evaluation Protocol

1) Generate policy document tasks across 3–5 domains (healthcare, education, climate, welfare).
2) Apply FeedbackRefine and all baselines to identical inputs.
3) Recruit 10–15 domain experts (policymakers, grant writers, project managers) for evaluation.
4) Collect rubric scores, edit counts, and satisfaction ratings per method.
5) Conduct semi-structured interviews to capture qualitative insights on usability and workflow fit.

VI. Limitations and Ethical Considerations

A. Limitations

Conceptual Scope: This paper proposes a framework without empirical validation. Hypotheses H1–H4 remain unconfirmed until a prototype is implemented and tested.

Domain Specificity: The quality rubric is designed for policy and governance documents. Adaptation to legal, medical, or technical domains requires domain-specific rubric extensions and expert input.

Feedback Quality Dependency: Refinement effectiveness is bounded by the quality of user feedback. Poorly specified or contradictory feedback may degrade document quality across iterations.

Generalization: It is unclear whether convergence behaviour and optimal iteration counts observed in one domain will transfer to others.

B. Ethical Considerations

Hallucination and Factual Accuracy: LLMs retain a non-trivial probability of generating plausible but factually inaccurate content [7]. In policy or governance contexts, fabricated facts could have serious real-world consequences. All AI-generated content must be presented as a draft requiring mandatory expert review, never as a final deliverable.

Bias in Generated Content: LLMs trained on large web corpora may encode social, political, or cultural biases that manifest in generated documents. Systematic bias auditing is essential during any future evaluation phase, with findings informing both prompt design and post-generation filtering strategies.

Data Privacy: Professional documents often contain sensitive information (organizational data, beneficiary details, budget figures). Any implementation must apply data minimization principles, encrypt sensitive fields, and evaluate the feasibility of locally hosted inference to eliminate third-party data exposure [4].

Over-Reliance Risk: Users may over-trust AI-generated content, reducing critical scrutiny. System design should make it structurally clear that FeedbackRefine is a writing assistant, not an authoritative author.

VII. Conclusion

This paper proposes **FeedbackRefine**, a conceptual framework for user-guided iterative refinement of LLM-generated professional documents. The framework addresses a clear gap: while self-refinement systems have shown promise, none systematically integrate real user feedback to guide iterative improvement of long-form, domain-specific documents.

FeedbackRefine contributes: (i) a four-layer system architecture, (ii) a five-stage iterative pipeline, (iii) a multi-dimensional quality rubric, (iv) a feedback transformation mechanism, (v) a principled convergence strategy, and (vi) a proposed evaluation protocol grounded in testable hypotheses.

Future work should implement a working prototype, conduct controlled experiments validating H1–H4, extend the framework to additional domains, and integrate retrieval-augmented generation (RAG) for improved factual grounding.

Acknowledgment

The authors express their sincere gratitude to the faculty members and staff of the Department of Master of Computer Applications, Progressive Education Society's Modern College of Engineering, Pune, for their assistance, encouragement, and academic support throughout this study.

Special thanks are extended to our friends and peers for their constructive discussions, motivation, and support during the research process.

Finally, the first author sincerely thanks his parents and family members for their unwavering encouragement, understanding, and constant support, which made this work possible.

References

[1] A. Madaan et al., "Self-Refine: Iterative refinement with self-feedback," in Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP), 2023. [Online]. Available: https://arxiv.org/abs/2303.17651

[2] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," OpenAI, 2019. [Online]. Available: https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf

[3] D. Lee, S. Park, E. Lee, and H. Lim, "Ask, assess, and refine: Rectifying factual consistency and hallucination in LLMs with metric-guided feedback learning," in Proc. 18th Conf. European Chapter Assoc. Computational Linguistics (EACL), 2024. [Online]. Available: https://aclanthology.org/2024.eacl-main.84.pdf

[4] L. Ouyang et al., "Training language models to follow instructions with human feedback," in Advances in Neural Information Processing Systems, vol. 35, 2022. [Online]. Available: https://arxiv.org/abs/2203.02155

[5] L. Pan, M. Saxon, W. Xu, D. Nathani, X. Wang, and W. Y. Wang, "Automatically correcting large language models: Surveying the landscape of diverse automated correction strategies," Transactions of the Association for Computational Linguistics, vol. 12, pp. 454–473, 2024. [Online]. Available: https://aclanthology.org/2024.tacl-1.27.pdf

[6] M. Furnas, J. Zhang, S. Freund, and N. Bhat, "Sci2Pol: Evaluating and fine-tuning LLMs on scientific-to-policy brief generation," arXiv preprint arXiv:2509.21493, 2025. [Online]. Available: https://arxiv.org/pdf/2509.21493.pdf

[7] R. Bommasani, D. A. Hudson, E. Adeli, et al., "On the opportunities and risks of foundation models," arXiv preprint arXiv:2108.07258, 2021. [Online]. Available: https://arxiv.org/pdf/2108.07258.pdf

[8] T. Brown et al., "Language models are few-shot learners," in Advances in Neural Information Processing Systems, vol. 33, 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[9] T. P. Ferraz et al., "LLM self-correction with DeCRIM: Decompose, critique, and refine for enhanced following of instructions," in Findings of EMNLP, 2024. [Online]. Available: https://aclanthology.org/2024.findings-emnlp.223.pdf

[10] W. Xu, D. Z. Wang, H. Geng, Z. Wu, X. Chen, M. Huang, and C. Shi, "LLMRefine: Pinpointing and refining large language models via fine-grained actionable feedback," in Proc. Conf. Empirical Methods Nat. Lang. Process. (EMNLP), 2023. [Online]. Available: https://arxiv.org/abs/2311.09336

[11] X. Xue, Z. Huang, Y. Ji, and H. Wang, "IMPROVE: Iterative model pipeline refinement and optimization leveraging LLM agents," arXiv preprint arXiv:2502.18530, 2025. [Online]. Available: https://arxiv.org/pdf/2502.18530.pdf