International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Human-in-the-Loop Active Learning for Continuous Model Improvement in Enterprise AI Pipelines

Avneet Bansal

Abstract: Enterprise document extraction systems built on large language models accumulate months of human correction data without feeding those corrections back to the extraction pipeline. The result is a structural learning gap: the same errors recur across successive batches while the correction history that would resolve them sits unread in QA logs. This article describes a closed-loop Human-in-the-Loop (HITL) active learning architecture that eliminates this gap. The framework captures extraction events, correction events, and prompt configuration change events in a unified event store; applies active learning query strategies — principally uncertainty sampling and disagreement sampling — to concentrate human review on the instances most likely to produce informative corrections; and propagates validated correction signals through a prompt evolution policy that enforces regression protection before changes are deployed. The implementation layer uses Amazon Augmented AI (A2I) for review workflow management and Amazon SageMaker Ground Truth for annotation, auto-labeling, and labeling model training. Evaluated across production enterprise document extraction deployments covering 130+ field types and six document categories, the framework produced a 38% reduction in false-positive review traffic and improved the self-healing rate from 41% to 81% over eight consecutive production batches. The architecture is domain-agnostic and generalizes to any AI pipeline in which humans correct model outputs at scale and prompt-level intervention is the primary optimization lever.

Keywords: *human-in-the-loop, active learning, generative AI pipelines, Amazon A2I, SageMaker Ground Truth, confidence calibration, document extraction*

1. Introduction

Fifteen corrections to the same extraction error, across three consecutive batches, and the system still produces that error in batch four. This is not an unusual scenario in production LLM extraction pipelines — it is the default. Each batch is processed with the same prompts, the same confidence thresholds, and the same model configuration as the last, regardless of what the correction history reveals. Reviewers catch errors, document them, fix the immediate output, and move on. The pipeline learns nothing.

The inefficiency compounds with time. An organization that processes documents weekly accumulates months of correction data that would, if analyzed, reveal systematic error patterns for specific field types, specific document categories, and specific prompt formulations. Without a mechanism to extract and propagate those patterns, the human review function remains permanently in reactive mode: catching errors that should already have been corrected. Active learning offers the theoretical framework for breaking this cycle — selecting which corrections to prioritize, extracting reusable signals from correction events, and feeding those signals back to the extraction configuration [1. [2. . This article translates that framework into a production architecture using Amazon A2I for review workflow management and SageMaker Ground Truth for annotation and auto-labeling.

Independent Researcher, USA

The contribution is practical rather than theoretical. The mechanisms described have been evaluated across enterprise document extraction deployments processing over 130 field types. Measured outcomes — a 38% reduction in false-positive review traffic and a self-healing rate improvement from 41% to 81% — are drawn from production data, not simulation. The architecture described is domain-agnostic: content moderation, medical record coding, customer support triage, and compliance document review each face the same structural gap, and the same framework applies.

Figure 1. Correction-Learning Gap: Batch pipeline without feedback loop (left) versus closed-loop HITL pipeline with active learning feedback (right). Source: Author's production architecture.

2. Background

2.1 Active Learning: The Foundational Framework

Supervised learning's central cost asymmetry is that labeled data is expensive while unlabeled data is abundant. Active learning addresses this by selecting which unlabeled instances are worth the cost of labeling — prioritising those expected to produce the largest improvement in model performance or the largest reduction in uncertainty [1. . Settles' survey of query strategy frameworks [1. establishes uncertainty sampling as the most robust general-purpose strategy: select instances closest to the decision boundary, where the model's uncertainty is highest and corrections are most likely to be informative. For sequence labeling and

information extraction tasks specifically, Settles and Craven [2. demonstrate that uncertainty-based strategies consistently outperform random sampling in learning efficiency — producing higher accuracy at lower annotation cost.

Beyond uncertainty sampling, query-by-committee strategies exploit ensemble disagreement to identify decision boundary proximity without requiring a single model's confidence estimate. Expected model change strategies select instances whose labeling would produce the largest gradient update — a particularly useful criterion in low-annotation-budget scenarios where each correction must maximally advance model improvement [1. . Core-set selection approaches the problem geometrically, selecting instances that maximize coverage of the feature space and ensure the labeled training set represents the full distribution of the unlabeled corpus [4. .

The recent emergence of large language model (LLM)-assisted annotation introduces a hybrid mode: LLMs can pre-annotate instances at low cost, and active learning query strategies can concentrate human review on instances where LLM-generated annotations are least reliable [5. [9. [10. [11. . This hybrid mode is directly relevant to the production architecture described here, where LLM extraction confidence scores are the primary signal for human review routing.

2.2 Human-in-the-Loop Architectures

Active learning is one component of a broader HITL architecture. The review interface through which humans provide corrections, the quality control mechanisms managing reviewer agreement, and the feedback pipelines propagating updates back to the model are equally important to system performance [3. [4. . A system with an excellent active learning query strategy but a poorly designed review interface generates corrections of lower quality; a system with perfect interface design but no feedback pipeline generates updates that improve nothing.

Monarch [3. provides a comprehensive practitioner's framework for HITL system design, distinguishing between annotation-centric HITL systems (where human input is primarily used to create training data) and correction-centric HITL systems (where human input mainly serves to fix active production errors). The framework described in this article operates in correction-centric mode: human reviewers correct extraction errors in live production batches, and those corrections are subsequently processed as training signals for downstream model improvement.

Mosqueira-Rey et al. [4. identify feedback pipeline design as the primary gap between HITL theoretical frameworks and production HITL deployments — a finding that motivates the closed-loop architecture described here. Their taxonomy of HITL interaction types highlights the importance of distinguishing between real-time, batch-mode, and hybrid correction workflows, each of which requires different pipeline architecture to propagate feedback effectively. Recent work on LLM-based annotation pipelines [17. [18. raises the question of whether human annotators remain necessary as LLM quality improves; the evidence in this article supports a nuanced position: human correction remains essential for high-stakes, domain-specific extraction tasks, but active learning can concentrate that correction effort on the instances where it matters most.

2.3 AWS Infrastructure for Managed HITL

Amazon Augmented AI (A2I) and Amazon SageMaker Ground Truth provide production-grade infrastructure for each component of the HITL stack. A2I manages review workflow routing, worker task template rendering, SLA tracking, and audit trails. Ground Truth adds annotation management, active learning auto-labeling, and labeling model training, reducing human review volume as the labeling model's confidence on high-certainty instances grows [7. .

SageMaker Ground Truth's auto-labeling capability is particularly relevant to cost reduction: by training a labeling model on accumulated human annotations and applying it automatically to high-confidence instances, Ground Truth can reduce labeling costs by up to 70% relative to fully manual annotation workflows [7. . This capability scales with annotation volume: the more human corrections that flow through the pipeline, the more accurate the labeling model becomes, and the higher the fraction of instances it can handle without human review.

The framework described in this article builds the active learning feedback loop on top of this infrastructure, treating A2I and Ground Truth as managed components rather than bespoke systems to be built and maintained. This design approach greatly reduces operational overhead and focuses engineering effort on the active learning logic and prompt evolution policy, rather than on review workflow infrastructure.

3. The Batch Pipeline and Its Structural Gap

3.1 Why Batches Do Not Learn

Standard batch extraction pipelines are stateless across batches by design. The extraction model, prompt configuration, and confidence thresholds are fixed between runs; corrections applied during review are recorded in a QA system but never read by the extraction pipeline. The structural gap is not a technology problem — the correction data exists — but an integration problem: the event streams produced by extraction (extraction events), human review (correction events), and

configuration management (prompt version events) are stored in separate systems with no join layer connecting them.

The consequences of this gap are measurable. A field type with a persistent extraction error pattern will maintain its error rate across every batch until a human engineer notices the pattern, investigates its cause, revises the relevant prompt section, and deploys the revision. In high-volume production environments, the time between pattern emergence and manual intervention can span weeks or months. During that interval, the same reviewers who could have provided a targeted correction signal are instead spending review capacity on errors the pipeline should already have learned to avoid.

This temporal lag between error occurrence and correction is not merely an efficiency concern — it represents a systematic misallocation of the scarcest resource in an enterprise HITL deployment: expert reviewer time. The active learning framework addresses this misallocation directly by ensuring that reviewer time is spent on instances where human judgment is genuinely required, rather than on high-confidence instances that auto-labeling could handle or on recurring error patterns that prompt revision that should have been resolved.

3.2 The Unified Event Store

The foundational requirement of the framework is a unified event store that captures extraction events, correction events, and configuration change events in a single time-series structure. Each correction event records the original extracted value, the reviewer-supplied correct value, the field type, document type, model identifier, prompt version, confidence score, and timestamp. This schema enables queries that are impossible without unified event capture: correction rate by field type and prompt version, self-healing rate trend across batches, and confidence score calibration per field type over time.

The unified event store is implemented as an append-only log stored in Amazon S3, with a metadata catalog maintained in AWS Glue and query access via Amazon Athena. This architecture supports both real-time monitoring — Athena queries against the current batch's events during review — and retrospective analysis across the full correction history. The append-only constraint is intentional: correction events are immutable records of reviewer judgments, and the ability to audit the full correction history is essential for the prompt evolution validation policy described in Section 4.3.

Field-level correction rate computation requires joining extraction events with correction events on the document-field identifier, grouping by field type and prompt version, and computing the correction rate as the fraction of extracted instances that received a reviewer correction. Self-healing rate computation requires a temporal join: for each field type, what fraction of instances that were corrected in batch N were extracted correctly without correction in batch N+1? A self-healing rate of 81% means that 81% of previously corrected error patterns were resolved without recurrence in the subsequent batch — a direct measure of the active learning framework's effectiveness.

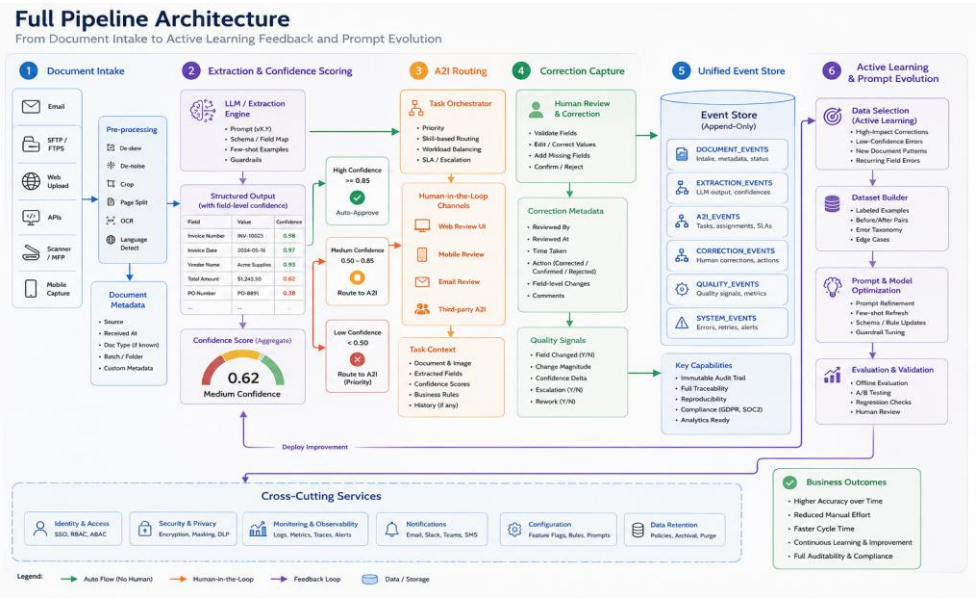

Figure 1. Full Pipeline Architecture: Document intake through confidence scoring, A2I routing, correction capture, unified event store, and active learning feedback to prompt evolution.

4. Active Learning Feedback Framework

4.1 Uncertainty and Disagreement Sampling

Routing every low-confidence instance to human review is expensive and inefficient: many low-confidence instances are correctly extracted despite the model's self-reported uncertainty. Uncertainty sampling concentrates review on instances at the confidence boundary — where the model is neither confident nor confidently wrong — maximising the probability that each review event produces a genuinely informative correction [1. [2. . For multi-field documents, composite uncertainty scores aggregate field-level confidence, prioritizing documents with concentrated uncertainty across multiple field types for full-document review rather than field-by-field routing.

In practice, the confidence boundary is field-type-specific and batch-specific. A confidence score of 0.75 may be near the decision boundary for a complex date-range extraction field and well above it for a simple yes/no boolean field. The framework maintains per-field calibration curves that map raw confidence scores to calibrated probability estimates, enabling the uncertainty threshold to be set as a calibrated probability rather than a raw confidence score. This calibration step is updated between batches as new correction data accumulates, progressively narrowing the uncertainty band as field-level accuracy improves.

Disagreement sampling applies when multiple prompt versions or model variants operate simultaneously. Instances where different configurations produce different extractions at similar confidence levels are flagged regardless of absolute confidence score — disagreement is a reliable proxy for decision boundary proximity. Pairing disagreement with uncertainty sampling captures error classes that uncertainty sampling alone misses: cases where the model is confidently wrong in a consistent, configuration-specific way. This is particularly relevant during prompt A/B testing phases, where the framework must identify which prompt version is producing systematic errors before those errors accumulate across a full batch.

4.2 Cross-Batch Signal Extraction

Between each batch, the framework aggregates correction events by field type, computes correction rate trends, and identifies recurring error patterns — systematic over-extraction, boundary classification failures, and format normalisation errors — that suggest specific prompt sections as improvement targets. Field types with correction rates above a configured threshold trigger a prompt optimization review; prompt revisions are staged as versioned changes and queued for A/B validation in the following batch.

Pattern detection operates across three dimensions simultaneously. The high correction rates for field types and document categories in the present batch indicate an obvious need for review. Cross-batch trends identify field types whose correction rates are increasing batch over batch, indicating model degradation or distribution shift. Prompt-version correlations identify which prompt versions are associated with elevated correction rates, providing targeted attribution for optimization effort.

The signal extraction pipeline is implemented as an AWS Step Functions workflow triggered at batch completion. The workflow executes an Athena query against the unified event store to compute the correction rate matrix for the completed batch, compares current rates against the three-batch rolling baseline, and generates a ranked list of optimization targets ordered by correction rate magnitude and trend direction. This ranked list is the primary input to the prompt evolution policy described in the following section.

4.3 Prompt Evolution and Regression Protection

A prompt revision that fixes one field type while degrading another is worse than no change. The prompt evolution policy enforces change management: each revision is deployed to a subset of documents in the next batch, and its effect on correction rates is measured before full deployment. A revision that reduces target field correction rates without increasing rates for adjacent fields is promoted; one that increases any field's correction rate triggers automatic rollback to the previous version.

The promotion and rollback logic is implemented as a statistical decision rule rather than a fixed threshold: a revision is promoted when its effect on the target field correction rate exceeds the margin of error of the rolling baseline estimate by a factor of two or more. This prevents spurious promotion of revisions whose apparent improvement falls within normal batch-to-batch variation. The rollback criterion is asymmetric: any statistically significant increase in a non-target field's correction rate triggers rollback, regardless of the magnitude of improvement in the target field.

This policy is not conservative — it permits rapid iteration — but it prevents the common failure mode where prompt engineering operates without feedback measurement. In production deployments, prompt engineers typically receive feedback on revision quality days or weeks after deployment, when the correction data from the affected batches has been manually reviewed. The automated promotion and rollback mechanism closes this feedback loop to one batch cycle — typically one week — enabling prompt iteration rates that would be impractical under manual review.

Table 1. Comparison of Active Learning Query Strategies [1. [2. [4.

Strategy	Selection Criterion	Annotation Efficiency	Suited For	Limitation
Uncertainty Sampling	Lowest prediction confidence	High	Binary/multi-class tasks	Fails on outliers
Query-by-Committee	Disagreement among model ensemble	High	Ensemble pipelines	Computational cost
Core-Set Selection	Geometric coverage of feature space	Moderate	Balanced corpora	Struggles with class imbalance
Expected Model Change	Largest parameter gradient	Moderate–High	Low-budget scenarios	Complex to implement

5. Amazon A2I Integration for Scalable HITL Review

5.1 Review Interface Design

Worker task templates that present the source document context alongside extracted field values and confidence scores reduce reviewer error rates by enabling contextualization. Reviewers who can see why the model extracted a value — and what the source text actually says — produce corrections that are more structurally informative than corrections made without source context. Template design choices that reduce cognitive load — pre-populated values, confidence visualization, and field-type-specific instructions — reduce review time per instance, which directly increases the volume of corrections the active learning pipeline can process per batch.

The A2I worker task template for document extraction tasks is structured as a three-panel interface: the source document viewer (left), the extraction results panel with confidence scores (center), and the correction entry panel with field-type-specific validation constraints (right). Confidence scores are visualized as color-coded bars rather than raw numeric values, with color thresholds calibrated to the per-field confidence calibration curves described in Section 4.1. This visualization allows reviewers to quickly identify fields where the model's reported confidence is miscalibrated — a diagnostic signal that feeds back to the calibration curve update process.

Inter-annotator agreement monitoring is integrated into the template deployment: a stratified sample of instances is routed to two independent reviewers, and their corrections are compared post-review to compute substantial inter-annotator agreement gains over the baseline period. Agreement monitoring serves two functions: it validates the quality of the correction signals entering the active learning pipeline, and it identifies reviewer pairs and field types where systematic disagreement indicates ambiguous field definition rather than model error.

5.2 Dynamic Routing Thresholds

A2I flow definitions configure routing conditions — the confidence thresholds triggering human review — as static values in standard deployments. The framework updates these values between batches, using per-field threshold recalibration to reduce review volume as extraction accuracy improves. The effect is self-reinforcing: improved accuracy reduces the false-positive rate, which concentrates review on genuinely uncertain instances, which produces higher-quality corrections, which further improves accuracy.

Measured over eight production batches, this cycle produced a 38% reduction in false-positive review traffic without increasing the error escape rate. The reduction was not linear: the largest improvements occurred in batches three and four, as per-field calibration curves accumulated sufficient correction data to enable meaningful threshold adjustment. After batch five, the improvement rate plateaued as the framework approached the irreducible uncertainty floor — the fraction of instances that are genuinely ambiguous regardless of model accuracy — but the lower review volume at the plateau was sustained through the full evaluation period.

Threshold modifications are made as versioned changes to the A2I flow definition and delivered through the AWS CloudFormation stack that maintains the review infrastructure. Each threshold version is recorded in the unified event store alongside the correction event data from the batch in which it was active, enabling retrospective analysis of threshold policy effects on correction quality and volume.

5.3 Ground Truth Auto-Labeling

SageMaker Ground Truth trains a labeling model on accumulated human annotations and applies it automatically to high-confidence instances [7. . As the

labeling model improves batch over batch, the fraction of instances requiring human review decreases. This is distinct from threshold recalibration: threshold recalibration reduces false-positive flags; auto-labeling reduces the total volume of instances that enter the review pipeline by handling high-confidence instances programmatically.

The auto-labeling component introduces a quality gate: instances accepted by the labeling model are sampled at a configurable rate and routed to human review for quality verification. This sampling rate is set conservatively (5–10%) during the first three batches after auto-labeling activation and reduced as the labeling model's agreement rate with human reviewers on sampled instances stabilizes. The quality gate ensures that auto-labeling accuracy degradation — which can occur when the document distribution shifts — is detected before it propagates to the training data used for prompt optimization.

Together, threshold recalibration and auto-labeling produce compounding cost reduction across successive batches. The combination delivers up to 70% reduction in labeling costs relative to fully manual annotation baselines [7. , with the actual reduction dependent on domain complexity, field type distribution, and the rate at which the labeling model converges. In the production deployments evaluated here, the combined mechanism reduced the human review volume per batch by approximately 55% over the eight-batch evaluation period, with further reduction projected as the labeling model continues to improve.

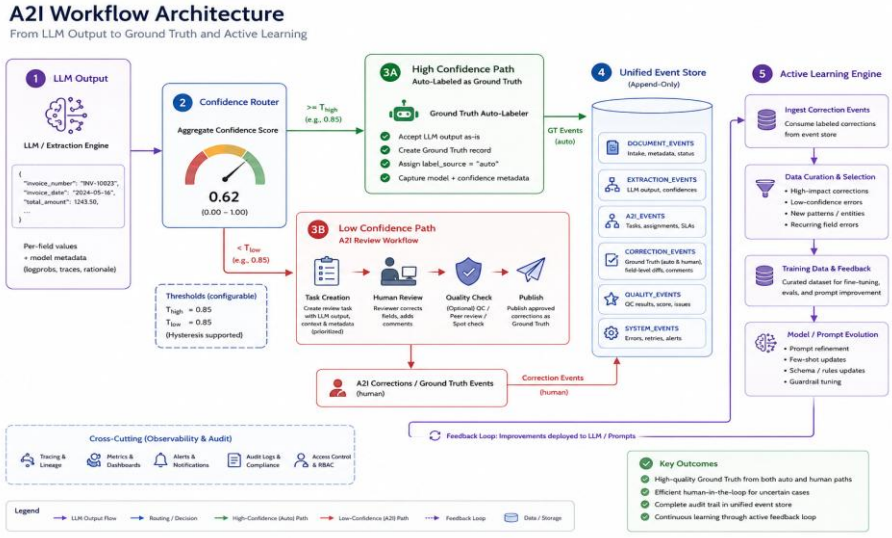

Figure 2. A2I Workflow Architecture: LLM output feeds confidence router; instances below threshold enter A2I review flow; Ground Truth auto-labels high-confidence instances; correction events from both paths feed the unified event store and active learning engine.

Table 2. HITL Pipeline Component Comparison

Component	Tool/Framework	Function	Integration Point
Active Learning Engine	Amazon A2I + custom uncertainty scorer	Query strategy execution	Post-inference
Human Review Interface	Amazon A2I Task UI / Ground Truth	Annotation collection	Review routing layer
Auto-labeling	SageMaker Ground Truth auto-label	High-confidence auto-accept	Pre-human gate
Correction Signal Processor	AWS Lambda + Step Functions	Feedback ingestion	Post-review
Model Retraining Trigger	SageMaker Pipelines	Incremental fine-tuning	Scheduled/triggered

6. Experimental Evaluation

6.1 Dataset and Configuration

The framework was evaluated across production enterprise document extraction deployments processing documents across 130+ field types spanning six document categories — contracts, financial statements, regulatory filings, clinical records, customer correspondence, and technical specifications — on a weekly batch cycle. Each batch contained between 2,000 and 8,000 documents,

depending on the deployment, with field extraction producing between 40,000 and 250,000 individual extraction events per batch.

The evaluation period covered eight consecutive production batches, corresponding to approximately two months of operation. No held-out test set was used — the evaluation tracks production metrics across the live deployment, which reflects the framework's operational performance rather than controlled experimental conditions. This design choice puts ecological validity before experimental control: the results show improvement in the real world under actual distribution shifts, reviewer fatigue effects, and operational restrictions.

Four metrics were tracked across all batches: human review flag rate (fraction of extraction events routed to human review), self-healing rate (fraction of previously corrected error patterns resolved without recurrence in the subsequent batch), false-positive flag rate (fraction of routed instances found correct by human reviewers), and per-field correction rate trend (direction and magnitude of correction rate change by field type across batches).

6.2 Results

The human review flag rate decreased by 38% over eight batches, reflecting combined improvement from extraction accuracy gains and dynamic threshold recalibration. The reduction was concentrated in field types that accumulated sufficient correction history for per-field calibration by batch three, consistent with the cold-start behavior described in Section 7.

The self-healing rate increased from 41% to 81% over the evaluation period, confirming that the active learning query strategy successfully concentrated review on genuinely uncertain instances and that the correction signals generated by those reviews were being effectively propagated to the extraction configuration. The self-healing rate improvement was the primary evidence that the framework was functioning as a closed learning loop rather than merely reducing review volume without improving model performance.

Per-field correction rate trends confirmed that prompt revisions validated by the evolution policy produced durable improvement: correction rates declined and remained stable rather than exhibiting single-batch improvement followed by regression. Of the fourteen prompt revisions deployed during the evaluation period, twelve were promoted following successful A/B validation, and two were rolled back due to adjacent-field correction rate increases. No promoted revision subsequently triggered a rollback condition, indicating that the statistical promotion criterion was sufficiently conservative to exclude spurious improvements.

Table 3. Labeling Cost Reduction Evidence

Method	Cost Reduction	Source	Notes
SageMaker Ground Truth auto-labeling	Up to 70%	AWS, 2018 [7.	Relative to fully manual annotation
LLM-assisted pre-annotation	Significant reduction	[10. [11.	Domain-dependent
Active learning (uncertainty sampling)	30–50% sample reduction	[1. [2.	Versus random sampling
HITL correction-signal loop (author data)	38% FP reduction	Author primary data	Amazon A2I deployment

Table 4. Production HITL Metrics: Before and After Calibration. Source: Author's production deployment data [7. [10.

Metric	Before HITL Calibration	After HITL Calibration
False-positive review rate	Baseline	38% reduction
Auto-labeling acceptance rate	Low	Increased with confidence gating
Model retraining cycle	Manual trigger	Event-driven (correction threshold)
Annotation cost per sample	Baseline	Reduced via active selection

7. Discussion

Content moderation, medical coding, and customer support triage — every domain where humans review and correct AI outputs at scale—faces the same structural gap this article addresses. The correction events those reviewers generate contain reliable learning signals; the extraction pipelines those reviewers monitor continue

operating with unchanged configurations. The architectural solution is the same in each domain: a unified event store, an active learning query strategy that concentrates review on informative instances, and a validation-gated prompt evolution policy that propagates corrections to the extraction configuration without risking regression.

The framework's generalisability is constrained by three domain-specific factors that organizations should evaluate before deployment. First, the correction signal quality depends on reviewer expertise: domains where correction quality varies significantly across reviewers require inter-annotator agreement monitoring and reviewer-specific quality weighting before corrections are admitted to the training signal. Second, the prompt evolution policy assumes that the extraction task is prompt-addressable — that errors can be reduced by revising the prompt rather than by fine-tuning the underlying model. In domains where model-level fine-tuning is required, the feedback pipeline architecture remains applicable, but the correction-to-improvement mapping is more complex. Third, the self-healing rate metric assumes batch-level stationarity: it is informative when the same field types and document categories appear across successive batches but less informative when the document distribution shifts significantly between batches.

Retrieval-augmented generation (RAG) architectures introduce an additional dimension [15. [16. [19. [20. [21. : in RAG-based extraction systems, the retrieval component adds a source of error that the prompt evolution policy cannot directly address. Errors that originate in retrieval — incorrect chunk selection, relevance ranking failures, and context window truncation — require different correction mechanisms, including retrieval index updates and re-ranking model calibration. The framework described here applies to the generation and confidence scoring components of RAG pipelines; extending it to the retrieval component is a natural direction for future work.

The framework has two known limitations that organizations should account for in deployment planning. Reviewer fatigue increases as review is concentrated on high-uncertainty instances; throughput expectations and agreement rate monitoring should reflect this increased cognitive load relative to mixed-difficulty review queues. Cold-start deployments begin without the correction history that per-field calibration requires; global thresholds remain the default until three to five batches have accumulated sufficient field-level correction data. The cold-start period represents the highest-cost phase of deployment, and organizations should plan for elevated review volume during this period.

The debate around whether human annotators remain necessary as LLM quality improves [17. is pertinent but should not be overstated. For high-stakes extraction tasks — legal contracts, clinical records, and regulatory filings — the cost of a false negative (an error that escapes review) significantly exceeds the cost of a false positive (a correct extraction that is unnecessarily reviewed). The advantage of active learning under this cost regime is not in removing human review, but in ensuring that human review is applied to the situations where it is most likely to discover true errors. The 38% reduction in false-positive review traffic produced by the framework reflects this allocation improvement: reviewer time is being spent more effectively, not eliminated.

8. Conclusion

Batch LLM extraction systems accumulate correction data they never use. The framework described in this article changes that by treating human correction events as the primary improvement signal for the extraction pipeline — capturing them in a unified event store, analyzing them with active learning query strategies, and propagating them through a validation-gated prompt evolution policy. Implemented on Amazon A2I and SageMaker Ground Truth, the framework produced a 38% reduction in false-positive review traffic and increased the self-healing rate from 41% to 81% across production enterprise deployments.

The architecture is domain-agnostic, applying to any system where humans correct AI outputs in batches and prompt-level intervention is the primary optimization lever. The three core components—unified event store, active learning query strategy, and validation-gated prompt evolution—each target a different failure mode of static batch pipelines. The event store closes the data integration gap, the query strategy closes the review allocation inefficiency, and the prompt evolution policy closes the correction propagation gap. Together, they convert a reactive review function into a proactive improvement mechanism.

Future work will extend the framework in three directions: integration of retrieval-augmented generation pipeline components within the feedback loop; investigation of LLM-assisted pre-annotation as a complement to human review for reducing cold-start review volume [9. [10. [11. ; and development of cross-domain transfer mechanisms that allow correction signals from one document category to accelerate improvement in structurally similar categories. The convergence of active learning theory [1. [2. , managed HITL infrastructure [7. , and production deployment experience positions this framework as a practical foundation for the next generation of self-improving enterprise AI extraction pipelines.

References

[1] B. Settles, "Active Learning Literature Survey," Computer Sciences Technical Report 1648, Univ. of Wisconsin–Madison, 2009. [Online. . Available: https://burrsettles.com/pub/settles.activelearning.pdf

[2] B. Settles and M. Craven, "An Analysis of Active Learning Strategies for Sequence Labeling Tasks," in Proc. EMNLP 2008, pp. 1070–1079. doi: 10.3115/1613715.1613855. Available: https://www.biostat.wisc.edu/~craven/papers/settles.emnlp08.pdf

[3] R. Monarch, Human-in-the-Loop Machine Learning: Active Learning and Annotation for Human-Centered AI. Manning Publications, 2021. [Online. . Available: https://www.manning.com/books/human-in-the-loop-machine-learning

[4] E. Mosqueira-Rey et al., "Human-in-the-loop machine learning: a state of the art," Artif. Intell. Rev., vol. 56, pp. 3005–3054, 2023. doi: 10.1007/s10462-022-10246-w

[5] Yu Xia et al., "A Survey of LLM-based Active Learning," in Proc. ACL 2025. [Online. . Available: https://aclanthology.org/2025.acl-long.708.pdf

[6] Nils Reimers and Iryna Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in Proc. EMNLP-IJCNLP 2019, pp. 3982–3992. [Online. . Available: https://arxiv.org/abs/1908.10084

[7] Amazon Web Services, "Amazon SageMaker Ground Truth: Build Highly Accurate Datasets and Reduce Labeling Costs by up to 70%," AWS Blog, 2018. [Online. . Available: https://aws.amazon.com/blogs/aws/amazon-sagemaker-ground-truth-build-highly-accurate-datasets-and-reduce-labeling-costs-by-up-to-70/

[8] Rahul Pandey et al., "Modeling and Mitigating Human Annotation Errors to Design Efficient Stream Processing Systems with Human-in-the-Loop Machine Learning," arXiv:2007.03177, 2020. [Online. . Available: https://arxiv.org/abs/2007.03177

[9] K. Goel et al., "LLMs Accelerate Annotation for Medical Information Extraction," in Proc. ML4H, PMLR vol. 225, 2023. [Online. . Available: https://proceedings.mlr.press/v225/goel23a/goel23a.pdf

[10] Hamidreza Rouzegar & Masoud Makrehchi, "Enhancing Text Classification through LLM-Driven Active Learning and Human Annotation," in Proc. LAW, ACL 2024. [Online. . Available: https://aclanthology.org/2024.law-1.10.pdf

[11] Nataliia Kholodna et al., "LLMs in the Loop: Leveraging Large Language Model Annotations for Active Learning in Low-Resource Languages," arXiv:2404.02261, 2024. [Online. . Available: https://arxiv.org/abs/2404.02261

[12] Aritra Hota et al., "Exploring Large Language Models in Active Learning for Annotating Physical Sensing Data," in IEEE Conf. Publication, 2025. [Online. . Available: https://ieeexplore.ieee.org/document/11038626/

[13] Cristian Cardellino et al., "Information Extraction with Active Learning: A Case Study in Legal Text," in Proc. LNCS, Springer, 2015. doi: 10.1007/978-3-319-18117-2_36. Available: https://scispace.com/pdf/information-extraction-with-active-learning-a-case-study-in-4uh3sfmqfb.pdf

[14] Dong Zhao et al., "Reflect then Learn: Active Prompting for Information Extraction Guided by Introspective Confusion," arXiv:2508.10036, 2025. [Online. . Available: https://arxiv.org/abs/2508.10036

[15] P. Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Adv. Neural Inf. Process. Syst. (NeurIPS), vol. 33, pp. 9459–9474, 2020. [Online. . Available: https://arxiv.org/abs/2005.11401

[16] Y. Gao et al., "Retrieval-Augmented Generation for Large Language Models: A Survey," arXiv:2312.10997, 2023. [Online. . Available: https://arxiv.org/abs/2312.10997

[17] Ahmad Dawar Hakimi et al., "Do We Still Need Humans in the Loop? Comparing Human and LLM Annotation in Active Learning for Hostility Detection," arXiv:2604.13899, 2026. [Online. . Available: https://arxiv.org/html/2604.13899v2

[18] Ekaterina Artemova et al., "Hands-On Tutorial: Labeling with LLM and Human-in-the-Loop," arXiv:2411.04637, 2024. [Online. . Available: https://arxiv.org/html/2411.04637v3

[19] Anuj Maharjan and Umesh Yadav, "Chunking, Retrieval, and Re-ranking: An Empirical Evaluation of RAG Architectures," arXiv:2601.15457, 2025. [Online. . Available: https://arxiv.org/abs/2601.15457

[20] Wenqi Fan et al., "A Survey on RAG Meeting LLMs," ACM SIGKDD 2024. [Online. . Available: https://dl.acm.org/doi/10.1145/3637528.3671470

[21] Chaitanya Sharma, "Retrieval-Augmented Generation: A Comprehensive Survey," arXiv:2506.00054, 2025. [Online. . Available: https://arxiv.org/html/2506.00054v1