An Agentic LLM Pipeline for Transforming Unstructured Clinical AI Literature into Structured Data

Bayarbaatar Bold[1] **Suyoung Yoo**[2] **Meong Hi Son**[3]

Abstract

Actionable clinical AI guidance is dispersed across heterogeneous, unstructured PDF documents, including reporting guidelines, implementation science frameworks, and digital health evaluation tools. We present a six-agent LLM pipeline that transforms this unstructured clinical AI literature into structured, machine-readable data through metadata extraction (1), count scanning (2), schema-constrained extraction (3), validation (4), reconciliation (5), and lifecycle-stage classification (6). Applied to 55 clinical AI framework documents, the pipeline extracted 2,386 classified leaf items mapped to eight predefined Clinical AI Lifecycle stages, enabling aggregate analysis of coverage patterns across the development and deployment lifecycle. A key methodological feature is the use of document-level checklist or item counts as weak supervision to detect extraction mismatches and trigger a self-correcting reconciliation loop. We demonstrate downstream utility through HOps, a hospital-facing digital product development framework informed by observed coverage gaps, while treating HOps as a case study rather than the primary technical contribution. Our work demonstrates that agentic LLM pipelines can convert unstructured clinical AI literature into auditable, machine-readable structured data suitable for aggregate coverage analysis and downstream framework development.

1. Introduction

A large share of healthcare knowledge remains stored in unstructured formats, including clinical notes, reports, guidelines, protocols, and scientific literature. Structured data enables comparison, computation, quality measurement, and reproducible analysis, yet many healthcare recommendations remain embedded in free text, tables, appendices, and PDF documents.

This challenge is especially visible in clinical AI. Guidance for developing and deploying clinical AI systems is dispersed across reporting guidelines, implementation science frameworks, digital health evaluation tools, and hospital governance documents. These sources contain actionable recommendations, but their heterogeneous PDF formats make them difficult to compare, query, or reuse in downstream governance workflows.

We frame this as an unstructured-to-structured data transformation problem. We present a six-agent sequential LLM pipeline that converts clinical AI guidance documents into hierarchical JSON and tabular data through (1) metadata extraction, (2) count scanning, (3) schema-constrained structure extraction, (4) validation, (5) reconciliation, and (6) lifecycle-stage classification. When source documents report checklist or item counts, the pipeline uses these counts as weak supervision to detect extraction mismatches and trigger reconciliation.

Applied to 55 clinical AI framework documents, the pipeline extracted 2,386 classified leaf items mapped to eight Clinical AI Lifecycle stages. The aggregate structured output revealed coverage gaps across the development and deployment lifecycle, particularly around user needs and design/prototyping. These findings informed HOps, a 20-question guiding framework for hospital digital product development, which we present as a downstream application supported by preliminary expert review.

2. Related Work

Structured information extraction from medical text. LLMs have demonstrated strong performance on clinical information extraction tasks, including span identification, sequence classification, relation extraction, and structured output generation from clinical text (Agrawal et al., 2022). Related work has also explored LLMs for medical evidence summarization and biomedical text processing (Guo et al., 2023). Our work extends this direction from clinical notes and biomedical articles to clinical AI guidance documents, where recommendations are distributed across heterogeneous PDFs and require hierarchical item extraction.

Agentic pipelines for complex document tasks. Sequential multi-agent LLM systems have been proposed for tasks

[1]Healthcare Digital Transformation Research Institute, Samsung Advanced Institute for Health Sciences & Technology, Seoul, Republic of Korea. [2]Research Institute for Future Medicine, Samsung Medical Center, Seoul, Republic of Korea. [3]Department of Emergency Medicine, Samsung Medical Center, Seoul, Republic of Korea. Correspondence to: Bayarbaatar Bold <bayarbaatar99@gmail.com>. Proceedings of the Workshop on Structured Data for Health at the 43rd International Conference on Machine Learning, Seoul, South Korea. Copyright 2026 by the author(s).

requiring decomposition, tool use, validation, and iterative improvement (Wu et al., 2023). Reflexion-style approaches further show how feedback from an intermediate attempt can improve subsequent outputs (Shinn et al., 2023). Our pipeline applies these ideas to unstructured-to-structured transformation by separating metadata extraction, count scanning, structure extraction, validation, reconciliation, and lifecycle-stage classification into distinct agents.

Clinical AI framework analysis. Clinical AI reporting and evaluation frameworks such as DECIDE-AI provide structured guidance for specific stages of AI development and evaluation (Vasey et al., 2022). However, guidance remains dispersed across reporting guidelines, implementation frameworks, governance tools, and digital health evaluation documents. Manual synthesis is difficult to scale and does not readily produce machine-readable item-level representations for lifecycle-stage comparison. Our approach complements prior framework work by transforming clinical AI guidance documents into structured data for aggregate coverage analysis and downstream framework development.

3. Method

3.1. Six-Agent Unstructured-to-Structured Pipeline

Figure 1 illustrates the full pipeline. The input is a set of clinical AI guidance PDFs, and the output is a structured representation of each document as hierarchical JSON and a flat tabular dataset. All agents use deterministic decoding where possible. Long documents are processed with a token-aware chunking strategy that splits text at paragraph boundaries and merges chunk-level outputs by category name.

3.2. Agent Descriptions

(1) Metadata Agent. The Metadata Agent extracts document-level provenance, including framework name, publication year, DOI when available, and primary focus area. These fields are attached to all downstream extracted items.

(2) Count Scanner Agent. The Count Scanner Agent searches the document for self-reported checklist or item totals, such as "17 items and 28 sub-items" or "42 reporting items across 6 domains." It returns the expected leaf-item count, supporting source sentence, and calculation logic. When no explicit count is found, the document is flagged for blind extraction.

(3) Structure Extraction Agent. The Structure Extraction Agent converts the document into a hierarchical schema of categories, items, and sub-items. Each extracted element includes an item label, item text, item type, and source text. Parent headers are distinguished from actionable leaf items and excluded from leaf-item counts. For narrative documents without explicit checklist formatting, the agent extracts inferred actionable recommendations while preserving source context for review.

(4) Validation Agent. The Validation Agent computes the extracted leaf-item count by summing standalone items and sub-items while excluding headers. It compares this count with the Count Scanner output and returns PASS, MISMATCH, or NO EXPECTED COUNT. This step is used as structural quality control: count agreement can detect omissions or over-extraction, but it does not by itself guarantee semantic correctness.

(5) Reconciliation Agent. When the Validation Agent returns MISMATCH, the Reconciliation Agent reinjects the expected count and supporting source sentence into the extraction prompt and retries extraction. Reconciliation is attempted up to two times. If no candidate exactly matches the expected count, the extraction with the smallest absolute count difference is retained and the mismatch is recorded for audit.

(6) Classification Agent. The Classification Agent assigns each leaf item to one of eight predefined Clinical AI Lifecycle stages (Table 2): Problem Framing, User Needs, Design & Prototyping, Development & Integration, Validation & Testing, Implementation & Adoption, Scale-Up & Sustainability, and Post-Deployment Evaluation. Items are classified in batches per document to reduce API calls.

3.3. Output Schema

The pipeline produces two linked outputs. The first is a hierarchical JSON document that preserves categories, items, sub-items, and provenance. The second is a flat table in which each row represents one leaf item, with fields for framework ID, framework name, year, category or phase, item label, item text, item level, item type, source text, DOI, CAL stage code, and CAL stage name.

4. Corpus and Structured Dataset

4.1. Corpus

We applied the pipeline to 55 clinical AI framework documents. The corpus was assembled to cover three types of guidance relevant to clinical AI development and deployment: AI/ML reporting guidelines, implementation science frameworks, and digital health or product development frameworks. These documents varied in structure, ranging from explicit checklists with numbered items to narrative guidance documents with recommendations embedded in prose, tables, and appendices.

For analysis, documents were grouped into three categories: G1, AI/ML reporting guidelines (n=15); G2, implementation science frameworks (n=10); and G3, product development, digital health, and evaluation frameworks (n=30). This grouping allowed us to examine whether different types of guidance emphasize different parts of the clinical AI lifecycle.

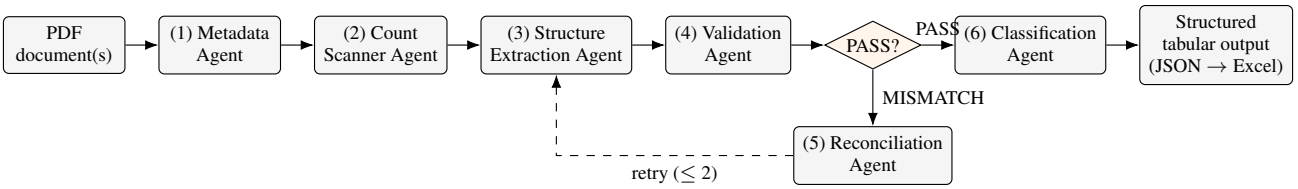

Figure 1. Six-agent pipeline for transforming unstructured clinical AI literature into structured data. The dashed arrow indicates the self-correcting reconciliation loop (up to two retries).

Table 1. Extracted items by corpus group and Clinical AI Lifecycle (CAL) stage.

Stage	G1	G2	G3	Total
PF	68	32	238	338
UN	27	24	151	202
DP	38	13	127	178
DI	120	2	138	260
VI	151	28	187	366
IA	30	191	265	486
SS	33	76	197	306
PE	42	61	147	250
Total	509	427	1450	2386

Table 2. Eight Clinical AI Lifecycle (CAL) stage codes used for classification.

Code	Description
PF	Problem framing, clinical rationale, target population
UN	User and patient needs, workflow context
DP	Model architecture, algorithm selection, output design
DI	Data, annotation, training, reproducibility
VI	External validation, bias testing, calibration
IA	Deployment, user training, rollout governance
SS	Regulatory status, funding, long-term planning
PE	Post-deployment surveillance, outcome tracking

4.2. Structured Dataset

Across the 55 documents, the pipeline extracted 2,386 leaf guidance items. Each item was represented as a row in a structured table and assigned to one of eight Clinical AI Lifecycle stages. Parent headers were retained in the hierarchical JSON output but excluded from the leaf-item count.

The structured output enables aggregate comparison across documents and lifecycle stages. Table 1 summarizes the distribution of extracted items by corpus group and lifecycle stage. Implementation & Adoption and Validation & Testing were the most represented stages, while User Needs and Design & Prototyping were less represented. This suggests that existing clinical AI guidance more often emphasizes evaluation and deployment than early-stage user-centered design and prototyping.

The row-level dataset is treated as a machine-assisted structured synthesis rather than a fully validated gold standard. Count matching and reconciliation provide structural quality control, but source-level semantic fidelity still requires expert review. In this paper, we therefore report aggregate coverage patterns and use the structured output to inform downstream framework development.

5. Case Study: Preliminary Development of HOps

We used the structured coverage analysis to inform HOps, a 20-question guiding framework for hospital digital product development. The goal of HOps is to translate gaps identified across the clinical AI lifecycle into practical questions for hospital teams developing or evaluating digital and AI-enabled products.

Table 3. HOps expert content validity results (n=5 experts).

Measure	n	Agreement
Question-level (≥4/5)	67/100 ratings	67%
Question range (min–max)	–	20–100%
Framework endorsement	13/15 statements	87%
Endorsement breakdown		
Addresses essential domains	5/5 experts	100%
Logically organised	4/5 experts	80%
Would strengthen practice	4/5 experts	80%

The extracted item distribution showed weaker coverage of User Needs and Design & Prototyping compared with Validation & Testing and Implementation & Adoption. These gaps motivated dedicated HOps questions on user needs elicitation, workflow fit, and the transition from clinical requirements to technical specifications. The "Mandatory Handshake" design gate was added to make explicit the connection between clinical problem definition, user needs, and product or model design decisions.

We conducted a preliminary content review with five domain experts in clinical informatics and digital health. Experts assessed the 20 HOps questions for clarity, completeness, and applicability using a 5-point scale, with agreement defined as

a rating of at least 4 out of 5. HOps achieved 87% framework-level endorsement and 67% question-level agreement (Table 3). Lower-agreement questions clustered around governance and accountability, and were revised to clarify ownership, decision authority, and accountability. Because the expert panel was small, we treat these results as preliminary and plan broader validation in future work.

6. Discussion and Conclusion

This work frames clinical AI guidance synthesis as an unstructured-to-structured data transformation problem. The six-agent pipeline shows that heterogeneous PDF guidance documents can be converted into hierarchical JSON and tabular representations using schema-constrained extraction, count-guided validation, reconciliation, and lifecycle-stage classification. The count-guided reconciliation loop is useful because many checklist-style documents report expected item totals, which can serve as weak supervision for detecting omissions or over-extraction.

The aggregate results suggest that existing clinical AI guidance is unevenly distributed across the development lifecycle. Validation & Testing and Implementation & Adoption are relatively well represented, while User Needs and Design & Prototyping receive less attention. This imbalance is important because early-stage design decisions influence whether clinical AI tools address real workflow needs before technical development and deployment. HOps demonstrates one downstream use of the structured synthesis by translating these gaps into practical questions for hospital digital product development.

Several limitations remain. First, count matching is a structural quality-control signal, not proof of semantic correctness. Extracted items may still contain segmentation errors, missing context, or incorrect lifecycle labels. Second, the row-level dataset should be treated as a machine-assisted synthesis requiring expert review rather than a definitive gold standard. Third, the corpus is English-only and focused on clinical AI framework documents, which may limit generalizability to other healthcare domains or multilingual guidance. Fourth, the HOps expert review involved only five experts, so the results should be interpreted as preliminary rather than definitive validation.

Future work will address these limitations through a human gold-standard audit of a document subset, comparison against simpler baselines such as single-pass extraction, a larger Delphi-style expert panel, and additional contextual metadata fields such as care setting, target population, and risk level. In conclusion, we present an agentic LLM pipeline for transforming unstructured clinical AI literature into structured data. Applied to 55 framework documents, the pipeline extracted 2,386 classified leaf items across eight lifecycle stages and enabled aggregate coverage analysis. These results support the use of agentic structured extraction as an auditable method for synthesizing clinical AI guidance and informing downstream governance and product development frameworks.

Impact Statement

This paper presents work aimed at advancing structured data methods in clinical AI governance. The structured dataset and HOps framework are intended to support safe, evidence-based deployment of AI systems in hospital settings. Potential societal benefits include improved governance of clinical AI, more systematic gap identification in development guidance, and a reproducible infrastructure for meta-research on clinical AI frameworks. We are not aware of specific harms arising from this work, though we note that automated extraction outputs require human expert review before use in safety-critical governance decisions.

References

Agrawal, M., Hegselmann, S., Lang, H., Kim, Y., and Sontag, D. Large language models are few-shot clinical information extractors. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1998–2022, 2022.

Guo, L., Hu, M., Zhao, Y., et al. Evaluating large language models on medical evidence summarization. *npj Digital Medicine*, 6(1):158, 2023.

Shinn, N., Cassano, F., Labash, B., et al. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2023.

Vasey, B., Nagendran, M., Campbell, B., et al. Reporting guideline for the early-stage clinical evaluation of decision support systems driven by artificial intelligence: DECIDE-AI. *Nature Medicine*, 28(5):924–933, 2022.

Wu, Q., Bansal, G., Zhang, J., et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*, 2023.