KnowMat: An Agentic Approach to Transforming Unstructured Material Science Literature into Structured Data

Hasan M. Sayeed[1], Casey Clark[1], Trupti Mohanty[1], and Taylor D. Sparks [*1]

[1]Department of Materials Science & Engineering, University of Utah, Salt Lake City, UT 84112, USA

Abstract

The materials-science literature is the richest reservoir of domain knowledge, yet converting its unstructured text—especially narrative passages and complex tables—into machine-readable data for analysis and ML model training remains challenging. To address this, we present KnowMat, an agentic, multi-stage pipeline that transforms full-text articles into schema-aligned, machine-readable JSON. KnowMat parses PDFs (text and tables), and performs iterative extraction with evaluation-driven re-runs to enhance coverage while curbing hallucinations. A two-stage manager then aggregates, validates, and corrects results, while properties are encoded with a fidelity-preserving dual representation (original textual form along with numeric surrogate with explicit value type); standardized labels are added without altering author-reported names to support database integration. Although demonstrated for materials literature, the workflow is schema-agnostic and readily adaptable to other scientific domains. Evaluation on real-world materials science papers demonstrates KnowMat's accuracy and efficiency, significantly reducing barriers to data-driven materials research.

1 Introduction

The field of machine learning (ML) for materials science faces a significant challenge due to the scarcity of high-quality data. Despite the rapid growth in research output, much of the experimental data remains locked within the confines of literature, which is often non-machine-readable and thus inaccessible for large-scale computational analysis [1, 2]. This limitation stifles the development of advanced ML models, as they rely heavily on rich, diverse datasets to achieve meaningful results.

Existing open-source databases in materials science predominantly focus on computationally derived properties [3–6]. While valuable, these datasets introduce potential systematic errors and are constrained to quantities that can be rapidly computed, leaving a gap in real-world, experimental data representation. Datasets extracted manually from literature do exist, but they are often proprietary, requiring significant financial and labor investments [7–9]. This exclusivity further hampers accessibility and widespread use.

Efforts to address these challenges have led to the development of various data extraction approaches. Rule-based methods like ChemDataExtractor, OSCAR, LeadMine, ChemicalTagger offer structured data extraction using predefined rules and ontologies [10–13], but they are limited in scalability and adaptability. Advancements in language models (LMs) have enabled data extraction tailored to field-specific tasks. Domain-specific pre-training and fine-tuning of LMs on manually annotated corpora have proven effective in domain-specific tasks [14–18]. With advances in large language models (LLMs), this has dramatically changed because LLMs can solve tasks for which they have not been explicitly trained. LLMs thus present a powerful and scalable alternative for structured data extraction. LLMs like GPT-4 and LLaMA-2, combined with prompt engineering techniques, have demonstrated remarkable success in extracting structured data directly from literature [19–28]. These methods not only identify relevant information but also ensure high precision and recall in the resulting datasets [24, 27].

For wider adoption, there is a growing need for general-purpose pipelines capable of transforming extracted data into structured formats suitable for database integration. Such databases can then serve as foundational resources for downstream ML models, fostering advances in materials discovery and property prediction. This growing interest in literature-to-data pipelines underscores a broader paradigm shift toward automating the capture and structuring of scientific knowledge.

Building upon this vision, KnowMat is developed as a schema-driven, multi-agent system designed to automate the transformation of unstructured materials science literature into machine-readable datasets. KnowMat employs an agentic architecture powered by GPT-based models orchestrated through the LangGraph framework. Specialized agents handle each stage of the workflow—from document parsing to data extraction, validation, and aggregation—ensuring modularity and robustness. By enforcing schema-constrained extraction using TrustCall-style tool invocation, KnowMat guarantees that outputs conform to predefined Pydantic models, triggering automatic corrections when struc-

*Corresponding author: sparks@eng.utah.edu

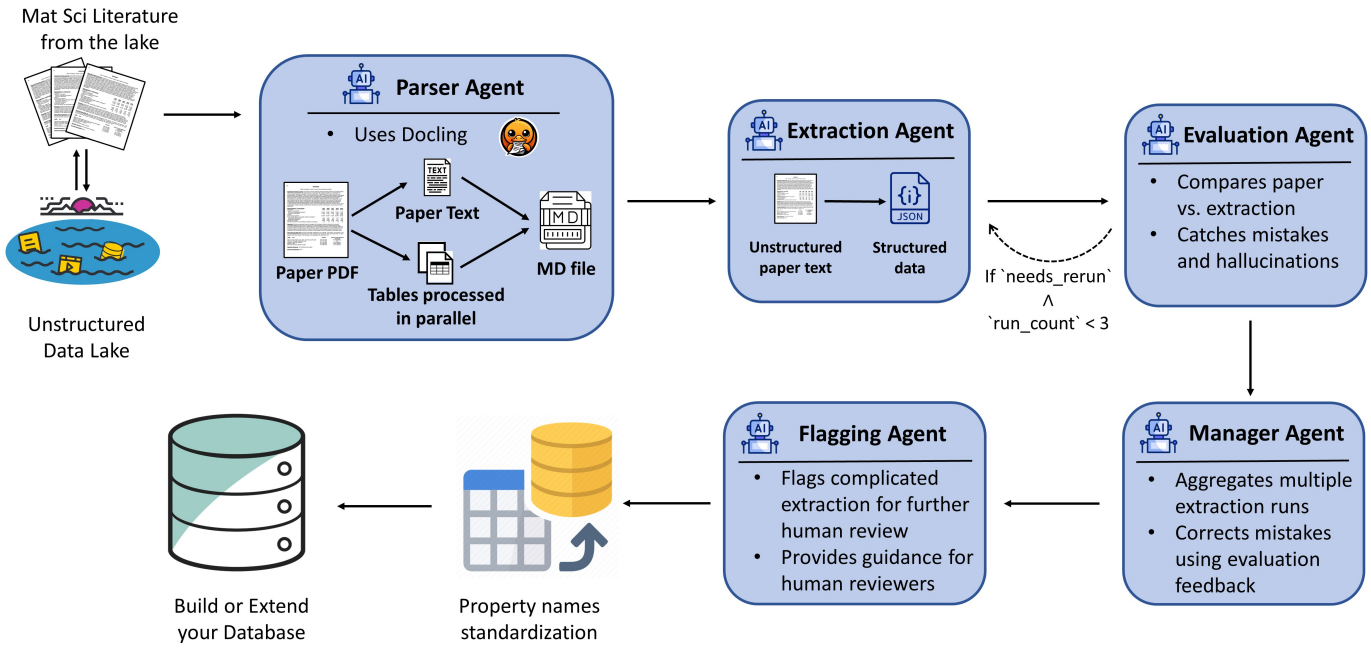

Figure 1: `KnowMat` agentic pipeline for turning materials-science PDFs into ML-ready JSON. The Parser Agent uses Docling to extract text and detect tables; when available, table images are converted in parallel to HTML and stitched back into the markdown. The Extraction Agent fills a schema via TrustCall-style, Pydantic-validated tool-calling. The Evaluation Agent compares the JSON to the source text, enumerates missing/hallucinated fields, and—if needed—automatically retriggers extraction (dashed loop; up to three cycles). A two-stage Manager first performs rule-based aggregation across runs, then an LLM Validation step corrects/filters fields and emits a human-review guide. Property-name standardization adds canonical labels without changing the author-reported names. Finally, the Flagging Agent assigns a confidence score and review recommendation, and the system writes the final JSON, analysis report, and run metadata for database/ML use.

tural mismatches are detected.

To ensure both fidelity and usability, `KnowMat` encodes extracted properties in an ML-ready form that preserves contextual fidelity (e.g., inequalities such as "<50 MPa") while also providing numerical representations required for downstream modeling. A property-name standardization module assigns canonical labels to extracted quantities while retaining the original phrasing to preserve interpretability and traceability. The system's modular and schema-agnostic design allows users to extend or replace its ontology, adapting it to different materials domains or even other scientific fields. By integrating advanced LLM reasoning with strict schema validation, `KnowMat` bridges the gap between unstructured scientific knowledge and structured, machine-readable data—laying the groundwork for scalable, reliable data curation in materials science.

2 Methodology

`KnowMat` is a LangGraph-orchestrated, multi-agent system that converts materials-science PDFs into machine-learning-ready (ML-ready) JSON (Figure 1). The graph executes: (i) PDF parsing, (ii) schema-constrained extraction, (iii) evaluation with optional re-runs (≤3), (iv) a two-stage manager (rule-based aggregation then LLM validation/correction), (v) property-name standardization, and (vi) final confidence scoring with human-review guidance. Model names and run limits are configurable.

Each component is described in detail below. Figure 1 provides a schematic representation of the workflow, highlighting the sequential interaction among these components.

2.1 Typed tool-calling (TrustCall)

All LLM interactions that return structured data are mediated through typed tool-calling (TrustCall-style). Each Agent is bound to a Pydantic model and the LLM must emit a payload that validates against the model. This gives three benefits:

- **Deterministic structure.** The extractor cannot "freestyle" keys: field names and types are canonical by construction.
- **Early failure on malformed outputs.** If an LLM response cannot be parsed/validated, it is not propagated; the Agent yields no structured update. In practice, that drives the downstream logic (evaluation and

manager fallbacks) rather than letting corrupt structure flow through.

- **Safer iteration.** Because the schema is explicit, prompt updates from the evaluation feedback can target specific fields (e.g., "preserve inequalities in `value`; set `value_type` accordingly"), leading to incremental improvements across re-runs.

Materials papers mix prose, tables, inequalities and ranges; typed tool-calling gives us a rigid envelope so we can reason about correctness and enforce ML-ready encoding.

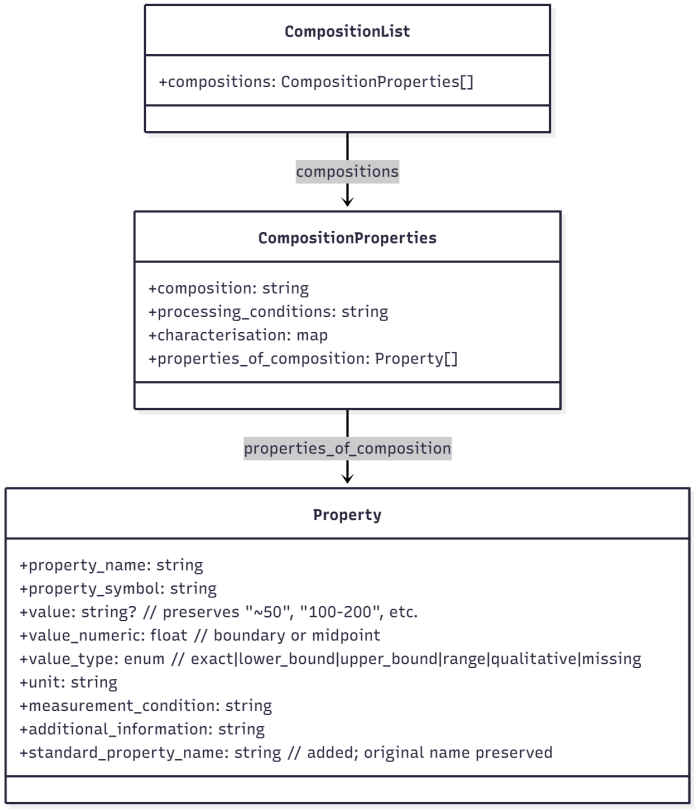

Figure 2: Schema used by `KnowMat` for structured materials-science data extraction. A `CompositionList` contains multiple `CompositionProperties` entries. Each `CompositionProperties` record stores the `composition`, optional free-text `processing_conditions`, a `characterisation` map (technique → finding), and a list of `Property` objects under `properties_of_composition`. Each `Property` includes the ML-ready triple—`value` (string that preserves author notation, including inequalities or ranges), `value_numeric` (boundary or midpoint surrogate), and `value_type` (exact, lower_bound, upper_bound, range, qualitative, or missing)—plus `unit`, optional `measurement_condition` and `additional_information`. A `standard_property_name` is added to support database joins and cross-study comparison, while the original `property_name` is preserved unchanged.

2.2 PDF parsing and table reconstruction

The Parser Agent uses Docling[29] to extract continuous text and detect tables. Each table is saved as a high-resolution PNG. Table PNGs are converted in parallel to HTML by a vision model and stitched back into a markdown. The output is a clean markdown document that preserves table structure (multi-row/column headers, merged cells, footnotes) alongside the article text.

2.3 Schema and ML-ready encoding (hierarchical, canonical, minimal)

Extraction targets a compact, hierarchical schema Figure 2:

- **Top level.**
  `"compositions": [CompositionProperties, …]`

- **CompositionProperties.**
  `composition` (string);
  `processing_conditions` (string);
  `characterisation` (map of technique → finding);
  `properties_of_composition` (list of `Property`).

- **Property.**
  `property_name`;
  `property_symbol` (optional);
  ML-ready triple:
  `value` (string; preserves original form including inequalities/ranges),
  `value_numeric` (float or null; boundary or midpoint),
  `value_type` ∈{exact, lower_bound, upper_bound, range, qualitative, missing};
  `unit` (string);
  `measurement_condition`;
  `additional_information` (optional).

This design is intentionally fidelity-first and ML-ready: we preserve the author's exact representation for auditability while simultaneously deriving a numeric surrogate that supports modelling.

2.4 Handling Fidelity and ML readiness

- **Inequalities.**
  A value like "`>50 mm`" is stored as `value=">50"`, `unit="mm"`, `value_type="lower_bound"`, and `value_numeric=50.0` (boundary).

- **Ranges.**
  "`100 - 200 m`" is stored as `value="100-200"`, `value_type="range"`, `value_numeric=150.0` (midpoint).

- **Missing.**
  "-" in a table is stored as `value=null`, `value_numeric=null`, `value_type="missing"`.

This dual representation (canonical string + numeric surrogate) allows downstream tasks to choose strict or permissive interpretations without re-parsing raw text.

2.5 Iterative extraction and automated reruns

The Extraction Agent (OpenAI GPT model; configurable) fills the schema above. Each run is immediately assessed by an Evaluation Agent that:

1. compares the JSON to the source (paper pdf) text;
2. emits missing and hallucinated fields in a strict actionable format:
   `FIELD_PATH | HALLUCINATED: … | PAPER_SAYS: … | FIX: …;`.
3. verifies the ML-ready triple and units;
4. proposes prompt updates.

If the Evaluator Agent thinks a rerun is necessary and rerun count is below a threshold, the graph automatically re-triggers the Extraction Agent with the updated prompt. This incremental loop is designed to increase recall (add missing entries) while reducing hallucinations across cycles. A safety check prevents accepting clearly invalid evaluation responses (e.g., zero confidence with empty rationale), forcing another pass.

2.6 Two-stage manager: aggregation then validation

To consolidate the multiple extraction/evaluation cycles and reduce single-pass error, `KnowMat2` uses a two-stage manager:

1. **Stage 1 — Aggregation (rule-based).**
   Choose the highest-confidence run as the base; merge in compositions from other runs; when a composition appears in multiple runs, prefer values from higher-confidence runs, preserve all characterisation entries, and avoid duplicate properties. No LLM calls here—this stage is deterministic and inexpensive.
2. **Stage 2 — Validation and correction (LLM).**
   Validate the aggregated result using the evaluator's evidence. Where the evaluation specifies a fix (e.g., "Converted >50 mm to 50.0 mm"), the validator restores the correct canonical representation (inequality as string) and enforces the ML-ready triple. Unfixable hallucinations are excluded. The validator outputs: (a) corrected final data, (b) an aggregation/validation rationale, and (c) a human-review guide with targeted checks. If a placeholder/lazy response is detected, the Agent retries with stronger instructions; on repeated failure it falls back to the best single run.

2.7 Property-name standardization

Before final scoring, a post-processing step maps property names to canonical labels using a curated property catalog with a lightweight GPT call for disambiguation. The original extracted property name is preserved; a new `standard_property_name` (see Figure 2) field is added (alongside optional `domain`/`category`). Standardization matters for:

- **Database builds and joins.**
  Harmonizes synonyms (e.g., "ultimate tensile strength", "UTS", "σmax") so entries aggregate correctly across papers.
- **Cross-study comparison.**
  Ensures that like-for-like properties are comparable without ad-hoc string matching.
- **Downstream analytics.**
  Stabilizes feature names in ML pipelines and facilitates ontology alignment when connecting to external resources.

2.8 Confidence scoring and human-in-the-loop guidance

A Flagging Agent assigns a final confidence score (0 - 1) and determines whether human review is required. This LLM-as-a-judge approach evaluates the aggregated results through a weighted assessment of four criteria: (i) manager correction quality (40%)—how successfully identified errors were corrected and documented; (ii) final data completeness (30%)—whether the extraction is comprehensive without excessive conservatism; (iii) original run consistency (20%)—cross-run agreement and variance; and (iv) residual uncertainties (10%)—complexity of remaining verification tasks in the review guidance.

Extractions with confidence $\geq$0.75 typically pass without mandatory review, while confidence $<$0.65 always triggers human verification. Importantly, successful corrections increase confidence rather than penalize the result—the system rewards effective error remediation. All extractions include comprehensive review guidance regardless of flag status, enabling optional expert verification in flexible deployment scenarios.

2.9 Modularity and schema extensibility

Although this work uses the schema above, the pipeline is schema-agnostic at the tool boundary:

- Swap-in alternative Pydantic models (e.g., domain-specific properties for polymers, catalysts, batteries, or even non-materials domains).
- Update the standardization map and the Evaluation/Validation prompts to reflect the new fields.

This allows teams to keep the graph logic (parsing, iterative evaluation, aggregation, validation, flagging) while adopting a different canonical schema.

2.10 Implementation and outputs

`KnowMat` is implemented in Python with modular Agents. For each paper the system writes: (i) the final JSON adhering to the schema; (ii) an analysis report with per-run rationales, missing/hallucinated lists, and aggregation/validation rationale; and (iii) run metadata for reproducibility. Configuration (model names, max number of reruns, thresholds, output directory) is via environment variables or CLI flags.

3 Results

3.1 Evaluation Dataset and Methodology

To evaluate `KnowMat`'s extraction performance, we curated a dataset of 20 scientific papers from materials science literature, specifically focusing on mechanical properties and thermal/thermodynamic properties sub-fields. Although `KnowMat` extracts comprehensive information including processing conditions, characterization methods, and material properties, this evaluation focuses on property extraction—the most critical component in materials informatics applications such as machine learning-based property prediction and inverse design tasks. Given that `KnowMat` is designed to process text and tabular data, we deliberately selected papers where properties are predominantly reported in these formats, ensuring alignment with the tool's current capabilities.

Ground truth annotations were created manually for each paper, with careful attention to four essential fields: composition, property name, numeric value, and unit. These fields form the foundation of quantitative materials data and are crucial for downstream computational applications.

3.2 Evaluation Metrics

We adapted the evaluation framework proposed by Schilling-Wilhelmi et al. [22] for materials property extraction. Our metrics are defined based on record-level matching as illustrated in Table 1. We evaluate extraction performance using standard information retrieval metrics: precision (the fraction of extracted properties that are correct), recall (the fraction of ground truth properties that are successfully extracted), and F1 score (the harmonic mean of precision and recall). A property is considered a True Positive (TP) only when all four fields—composition, property name, numeric value, and unit—match between the extracted and ground truth data.

To account for variations in terminology and formatting, we employ a hierarchical matching strategy:

- Composition matching: Exact string match (case-insensitive)
- Property name matching:
  - First, exact string match
  - If no match, fuzzy matching with a similarity threshold using sequence matching, including substring containment to handle prefix/suffix variations (e.g., "elastic stiffness" matching "elastic constant")
  - As a fallback, matching on standardized property name when property name differs
- Numeric value matching: Exact numerical equality
- Unit matching: After normalization to handle common variations

Unit normalization is critical for accurate evaluation, as the same physical unit can be represented in multiple mathematically equivalent forms across different papers. For example, thermal conductivity units may appear as "W/mK", "W/m · K", or "W(mK)$^{-1}$"—all representing the same quantity. Without normalization, these valid matches would be incorrectly classified as false positives or false negatives, significantly skewing evaluation metrics.

3.3 Overall Performance

Figure 3 presents the aggregated evaluation results across all 20 papers in our test dataset, comparing strict evaluation (property name matching only) with relaxed evaluation (allowing standardized property name fallback).

`KnowMat` achieves an F1 score of 86.2% in strict evaluation mode, demonstrating strong overall performance in extracting material properties from scientific literature. The precision of 90.4% indicates that when `KnowMat` extracts a property record, it is correct approximately 9 out of 10 times, suggesting a low rate of hallucination. The recall of 83.9% shows that the system successfully captures about 8 out of 10 properties present in the ground truth, indicating some properties are missed but the majority are extracted.

3.4 Impact of Standard Property Name Matching

When employing relaxed matching that considers standardized property names as a fallback when exact property names differ, we observe modest improvements: precision increases to 90.9% (+0.5 percentage points), recall to 84.3% (+0.4 percentage points), and F1 score to 86.6% (+0.4 percentage points). This improvement validates an important hypothesis: `KnowMat` captures the correct conceptual property even when the extracted terminology differs from ground truth. For instance, if the ground truth labels a property as "elastic modulus" but `KnowMat` extracts "Young's modulus", both may map to the same standardized

Table 1: Definition of outcome types for materials property extraction evaluation. A record is considered a True Positive (TP) only when all four fields (composition, property name, value, unit) match between extracted and ground truth data. False Positives (FP) occur when the system extracts data that either does not exist in the ground truth or contains incorrect values. False Negatives (FN) represent properties present in ground truth but missed by the extraction. True Negatives (TN) are not meaningful in this context as there are potentially infinite non-existing data points.

Outcome Type	Content Example	Extracted Data	Expected Data
True Positive (TP)	The Seebeck coefficient of $RuGa_3$ is -546 μV/K at 373 K.	Composition: $RuGa_3$ Property: Seebeck coefficient Value: -546 Unit: μV/K	Composition: $RuGa_3$ Property: Seebeck coefficient Value: -546 Unit: μV/K
False Positive (FP)	The Seebeck coefficient of $RuGa_3$ is -546 μV/K at 373 K.	Composition: $RuGa_3$ Property: Seebeck coefficient Value: -450 Unit: μV/K	Composition: $RuGa_3$ Property: Seebeck coefficient Value: -546 Unit: μV/K
False Negative (FN)	The Seebeck coefficient of $RuGa_3$ is -546 μV/K at 373 K.	Composition: Al_2O_3 Property: thermal conductivity Value: 25 Unit: W/m $\cdot$ K	Composition: $RuGa_3$ Property: Seebeck coefficient Value: -546 Unit: μV/K
False Negative (FN)	The Seebeck coefficient of $RuGa_3$ is -546 μV/K at 373 K.	None	Composition: $RuGa_3$ Property: Seebeck coefficient Value: -546 Unit: μV/K

property name, resulting in a semantically correct extraction despite the literal mismatch.

The relatively small magnitude of this improvement ($<$ 0.5%) is actually a positive indicator—it suggests that `KnowMat`'s fuzzy matching mechanism is already effectively handling most terminological variations. The standardized name matching primarily helps in edge cases where property names differ substantially but refer to the same underlying physical quantity. This demonstrates that the extraction system is robust and consistent in its terminology usage.

3.5 Performance Distribution and Outlier Analysis

The performance across individual papers shows that 7 out of 20 papers (35%) achieve perfect extraction with 100% precision and recall, demonstrating that for well-structured papers, `KnowMat` can perform flawlessly. A further 13 out of 20 papers (65%) achieve F1 scores above 70%, indicating generally strong performance across the dataset.

Notably, 10 out of 20 papers (50%) exhibit perfect precision (100%), with zero false positives. This indicates that `KnowMat` is highly conservative in its extractions and rarely hallucinates non-existent data—a critical characteristic for building reliable materials databases.

The lowest performing paper (paper_28) achieves an F1 score of 57.1% (precision: 100%, recall: 40.0%), demonstrating that even in challenging cases, the system maintains perfect precision while missing some properties. This pattern—high precision with incomplete coverage—reflects `KnowMat`'s conservative validation mechanisms that prioritize correctness over exhaustive extraction.

3.6 Precision and Recall Patterns

The high overall precision (90.4%) is especially valuable for building reliable materials databases, as it ensures that extracted records are trustworthy. In materials informatics applications such as machine learning model training or knowledge graph construction, false positives (hallucinated data) are more problematic than false negatives (missed data), as they introduce incorrect information that can propagate through downstream analyses.

Most false negatives occur in papers with complex reporting structures. For example, paper_09 shows perfect precision but only 69.7% recall, suggesting the system conservatively avoided extracting properties that were present but perhaps reported in unusual formats or embedded in complex narrative text. In contrast, paper_12 demonstrates interesting behavior with very high recall (98.8%) but lower precision (80.9%) due to 38 false positives—the system appears to have aggressively extracted from extensive tabular data, possibly including derived or calculated properties not annotated in ground truth.

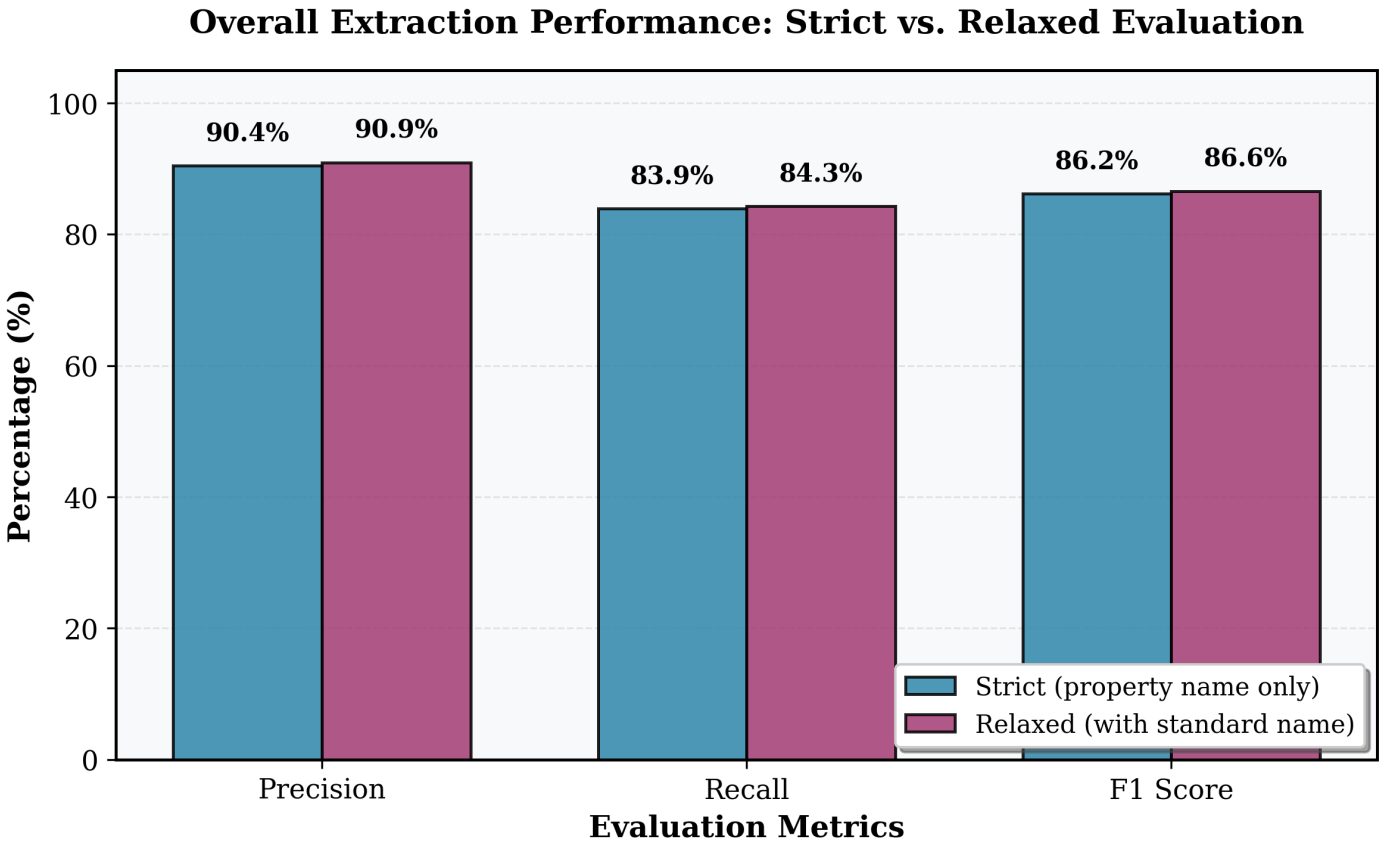

Figure 3: Comparison of average extraction performance between strict and relaxed evaluation modes across 20 materials science papers. Strict evaluation requires exact or fuzzy matching of property names, while relaxed evaluation additionally considers standardized property names as a fallback. The modest improvement in relaxed mode (precision: +0.5%, recall: +0.4%, F1: +0.4%) demonstrates that the system's fuzzy matching already captures most terminological variations effectively, with standardized names primarily assisting in edge cases where property names differ substantially but refer to the same physical quantity.

3.7 Human Review Guidance and Practical Deployment

An important feature of `KnowMat` is its built-in human review guidance system. Regardless of whether extraction results are flagged for review, the tool automatically generates detailed guidance indicating what information might be missing, where potential errors or ambiguities exist, and how to verify and correct the extracted data.

Importantly, the evaluation results presented here represent fully automated extraction without any human intervention or review. This ensures the metrics reflect real-world, out-of-the-box performance. However, the availability of structured review guidance means that in practical applications, users can systematically improve extraction quality by following the provided recommendations. This human-in-the-loop capability makes error correction efficient and reduces the cognitive burden on domain experts, as they receive specific, actionable feedback rather than needing to manually compare raw papers with extracted data.

3.8 Interpretation and Implications

The 86.2% F1 score achieved by `KnowMat` represents strong performance for automated materials property extraction from unstructured scientific literature. This level of accuracy is particularly notable given the challenges inherent in materials science papers: high diversity in property naming conventions, complex tabular structures with nested headers, multi-component compositions with varying notation systems, and mixed units and measurement conditions.

The moderate recall (83.9%) indicates room for improvement in capturing all available properties. Analysis suggests that missed properties often fall into categories such as: properties mentioned in narrative text without clear tabular structure, values embedded in figure captions or graphical representations, implicit properties (e.g., properties mentioned only by symbol without full name), and properties in complex multi-level tables with unclear composition-property associations. Future work incorporating enhanced image-based table extraction and deeper context understanding could address these gaps.

4 Discussion and Conclusion

KnowMat demonstrates strong performance in automated materials property extraction, achieving an F1 score of 86.2% with notably high precision (90.4%). This conservative extraction behavior—prioritizing correctness over exhaustive coverage—is particularly valuable for building reliable materials databases, as false positives (hallucinated data) are more problematic than false negatives in downstream applications like machine learning model training. The system's ability to achieve perfect extraction (100% precision and recall) on 35% of test papers indicates that for well-structured documents, the multi-agent architecture with iterative refinement effectively captures material properties.

However, current limitations must be acknowledged. KnowMat is designed to work exclusively with text and tabular data—it does not extract information from figures or graphical representations. This is a deliberate scope limitation rather than a performance issue. Pre-trained LLMs are not yet robust enough to reliably extract quantitative data from scientific plots, charts, and embedded graphs. While the system handles text-based tables well through Docling parsing and vision-model reconstruction (including complex multi-level tables with merged cells and nested headers), extracting precise numerical values from graph axes, trend lines, or scatter plots is outside its current scope.

This represents a notable gap in coverage, as many materials papers report critical property-composition relationships primarily through figures.

Future work should explore integrating specialized tools for figure-based extraction. An agentic architecture could be extended to detect figures and invoke external utilities like WebPlotDigitizer when graphical data is encountered. Fine-tuned vision models specifically trained on scientific figures might also improve capabilities in this area. Such extensions would significantly expand coverage while maintaining KnowMat's current strength in text and table extraction.

Despite this limitation, KnowMat presents an effective, open-source solution for transforming unstructured materials science literature into structured, ML-ready data. The fidelity-preserving dual representation of properties—retaining author notation while providing numeric surrogates—enables both auditability and downstream modeling. The schema-agnostic design and modular architecture ensure broad applicability; the same workflow can be adapted to other scientific domains by simply updating the Pydantic schema and property catalog. This flexibility positions KnowMat as a general framework for scientific knowledge extraction, establishing a foundation for scalable, reliable knowledge extraction that accelerates the path from published literature to actionable datasets for machine learning and data-driven discovery.

Data and Code Availability

The KnowMat tool is open-source and available at https://github.com/hasan-sayeed/KnowMat2

Acknowledgments

We acknowledge the assistance provided by ChatGPT, which was used for rephrasing and achieving coherence. However, it's important to note that all core ideas, text, tables, and figures were the original work of the authors.

H.M.S., C.C. and T.D.S. acknowledge support from the National Science Foundation, Division of Materials Research, under Award No. 2334411. T.M. and T.D.S. acknowledge support from the Army Research Office Materials Design, under Contract No. W911NF-23-1-0333.

Conflict of Interest

On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

(1) Olivetti, E. A.; Cole, J. M.; Kim, E.; Kononova, O.; Ceder, G.; Han, T. Y.-J.; Hiszpanski, A. M. Data-driven materials research enabled by natural language processing and information extraction. *Applied Physics Reviews* **2020**, *7*.

(2) Sayeed, H. M.; Smallwood, W.; Baird, S. G.; Sparks, T. D. NLP meets Materials Science: Quantifying the presentation of materials data in scientific literature.

(3) Curtarolo, S.; Setyawan, W.; Wang, S.; Xue, J.; Yang, K.; Taylor, R. H.; Nelson, L. J.; Hart, G. L.; Sanvito, S.; Buongiorno-Nardelli, M., et al. AFLOWLIB. ORG: A distributed materials properties repository from high-throughput ab initio calculations. *Computational Materials Science* **2012**, *58*, 227–235.

(4) Talirz, L.; Kumbhar, S.; Passaro, E.; Yakutovich, A. V.; Granata, V.; Gargiulo, F.; Borelli, M.; Uhrin, M.; Huber, S. P.; Zoupanos, S., et al. Materials Cloud, a platform for open computational science. *Scientific data* **2020**, *7*, 299.

(5) Jain, A.; Ong, S. P.; Hautier, G.; Chen, W.; Richards, W. D.; Dacek, S.; Cholia, S.; Gunter, D.; Skinner, D.; Ceder, G., et al. Commentary: The Materials Project: A materials genome approach to accelerating materials innovation. *APL materials* **2013**, *1*.

(6) Kirklin, S.; Saal, J. E.; Meredig, B.; Thompson, A.; Doak, J. W.; Aykol, M.; Rühl, S.; Wolverton, C. The Open Quantum Materials Database (OQMD): assessing the accuracy of DFT formation energies. *npj Computational Materials* **2015**, *1*, 1–15.

(7) Zagorac, D.; Müller, H.; Ruehl, S.; Zagorac, J.; Rehme, S. Recent developments in the Inorganic Crystal Structure Database: theoretical crystal structure data and related features. *Journal of applied crystallography* **2019**, *52*, 918–925.

(8) Groom, C. R.; Bruno, I. J.; Lightfoot, M. P.; Ward, S. C. The Cambridge structural database. *Acta Crystallographica Section B: Structural Science, Crystal Engineering and Materials* **2016**, *72*, 171–179.

(9) Blokhin, E.; Villars, P. The PAULING FILE project and materials platform for data science: From big data toward materials genome. *Handbook of Materials Modeling: Methods: Theory and Modeling* **2020**, 1837–1861.

(10) ChemDataExtractor 2.0: Autopopulated Ontologies for Materials Science | Journal of Chemical Information and Modeling.

(11) Jessop, D. M.; Adams, S. E.; Willighagen, E. L.; Hawizy, L.; Murray-Rust, P. OSCAR4: a flexible architecture for chemical text-mining. *Journal of Cheminformatics* **2011**, *3*, 41.

(12) Lowe, D. M.; Sayle, R. A. LeadMine: a grammar and dictionary driven approach to entity recognition. *Journal of Cheminformatics* **2015**, *7*, S5.

(13) Hawizy, L.; Jessop, D. M.; Adams, N.; Murray-Rust, P. ChemicalTagger: A tool for semantic text-mining in chemistry. *Journal of Cheminformatics* **2011**, *3*, 17.

(14) Shetty, P.; Rajan, A. C.; Kuenneth, C.; Gupta, S.; Panchumarti, L. P.; Holm, L.; Zhang, C.; Ramprasad, R. A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing. *npj Computational Materials* **2023**, *9*, Publisher: Nature Publishing Group, 1–12.

(15) Guo, J.; Ibanez-Lopez, A. S.; Gao, H.; Quach, V.; Coley, C. W.; Jensen, K. F.; Barzilay, R. Automated Chemical Reaction Extraction from Scientific Literature. *Journal of Chemical Information and Modeling* **2022**, *62*, Publisher: American Chemical Society, 2035–2045.

(16) Rocktäschel, T.; Weidlich, M.; Leser, U. ChemSpot: a hybrid system for chemical named entity recognition. *Bioinformatics* **2012**, *28*, 1633–1640.

(17) Kononova, O.; Huo, H.; He, T.; Rong, Z.; Botari, T.; Sun, W.; Tshitoyan, V.; Ceder, G. Author Correction: Text-mined dataset of inorganic materials synthesis recipes. *Scientific Data* **2019**, *6*, Publisher: Nature Publishing Group, 273.

(18) Huang, S.; Cole, J. M. BatteryDataExtractor: battery-aware text-mining software embedded with BERT models. *Chemical Science* **2022**, *13*, Publisher: The Royal Society of Chemistry, 11487–11495.

(19) Dunn, A.; Dagdelen, J.; Walker, N.; Lee, S.; Rosen, A. S.; Ceder, G.; Persson, K.; Jain, A. Structured information extraction from complex scientific text with fine-tuned large language models. *arXiv preprint arXiv:2212.05238* **2022**.

(20) Polak, M. P.; Morgan, D. Extracting accurate materials data from research papers with conversational language models and prompt engineering. *Nature Communications* **2024**, *15*, Publisher: Nature Publishing Group, 1569.

(21) Polak, M. P.; Modi, S.; Latosinska, A.; Zhang, J.; Wang, C.-W.; Wang, S.; Hazra, A. D.; Morgan, D. Flexible, Model-Agnostic Method for Materials Data Extraction from Text Using General Purpose Language Models. *Digital Discovery* **2024**, *3*, arXiv:2302.04914 [cond-mat], 1221–1235.

(22) Schilling-Wilhelmi, M.; Ríos-García, M.; Shabih, S.; Victoria Gil, M.; Miret, S.; T. Koch, C.; A. Márquez, J.; Maik Jablonka, K. From text to insight: large language models for chemical data extraction. *Chemical Society Reviews* **2025**, Publisher: Royal Society of Chemistry, DOI: 10.1039/D4CS00913D.

(23) Dagdelen, J.; Dunn, A.; Lee, S.; Walker, N.; Rosen, A. S.; Ceder, G.; Persson, K. A.; Jain, A. Structured information extraction from scientific text with large language models. *Nature Communications* **2024**, *15*, Publisher: Nature Publishing Group, 1418.

(24) Choi, J.; Lee, B. Accelerating materials language processing with large language models. *Communications Materials* **2024**, *5*, Publisher: Nature Publishing Group, 1–11.

(25) Lei, G.; Docherty, R.; Cooper, S. J. Materials science in the era of large language models: a perspective. *Digital Discovery* **2024**, *3*, Publisher: RSC, 1257–1272.

(26) Ye, Y.; Ren, J.; Wang, S.; Wan, Y.; Wang, H.; Razzak, I.; Hoex, B.; Xie, T.; Zhang, W. Construction and Application of Materials Knowledge Graph in Multidisciplinary Materials Science via Large Language Model, arXiv:2404.03080 [cs], 2024.

(27) Zheng, Z.; Zhang, O.; Borgs, C.; Chayes, J. T.; Yaghi, O. M. ChatGPT Chemistry Assistant for Text Mining and the Prediction of MOF Synthesis. *Journal of the American Chemical Society* **2023**, *145*, 18048–18062.

(28) Xie, T.; Wan, Y.; Huang, W.; Zhou, Y.; Liu, Y.; Linghu, Q.; Wang, S.; Kit, C.; Grazian, C.; Zhang, W.; Hoex, B. Large Language Models as Master Key: Unlocking the Secrets of Materials Science with GPT, arXiv:2304.02213 [cs], 2023.

(29) Livathinos, N. et al. Docling: An Efficient Open-Source Toolkit for AI-driven Document Conversion, arXiv:2501.17887 [cs], 2025.