Communications Materials

https://doi.org/10.1038/s43246-026-01267-x

Article in Press

An efficient AI-assisted approach for retrieving knowledge of polymer biodegradability from a corpus of publications

Received: 25 February 2026

Accepted: 29 June 2026

Published online: 15 July 2026

Cite this article as: Gupta, S., Mahmood, A., Xiong, W. *et al.* An efficient AI-assisted approach for retrieving knowledge of polymer biodegradability from a corpus of publications. *Commun Mater* (2026). https://doi.org/10.1038/s43246-026-01267-x

Sonakshi Gupta, Akhlak Mahmood, Wei Xiong & Rampi Ramprasad

An efficient AI-assisted approach for retrieving knowledge of polymer biodegradability from a corpus of publications

Sonakshi Gupta,[1] Akhlak Mahmood,[2] Wei Xiong,[3] and Rampi Ramprasad[3,*]

[1]*School of Computational Science and Engineering, Georgia Institute of Technology, 771 Ferst Drive NW, Atlanta 30332, GA, USA.*

[2]*Matmerize, Inc., Atlanta 30308, GA, USA.*

[3]*School of Materials Science and Engineering, Georgia Institute of Technology, 771 Ferst Drive NW, Atlanta 30332, GA, USA.*

Abstract

Polymer literature contains a large and growing body of experimental knowledge, yet much of it remains buried in unstructured text and inconsistent terminology, limiting systematic retrieval and reasoning. Existing tools typically extract narrow, study-specific facts in isolation, failing to preserve the cross-study context required to answer broader scientific questions. Retrieval-augmented generation (RAG) offers a promising approach by combining large language models (LLMs) with external retrieval, but its effectiveness depends strongly on how domain knowledge is represented. In this work, we develop two retrieval pipelines: a dense semantic vector-based approach (VectorRAG) and a graph-based approach (GraphRAG). Using over 1,000 polyhydroxyalkanoate (PHA) papers, we construct context-preserving paragraph embeddings and a canonicalized structured knowledge graph that supports entity disambiguation and multi-hop reasoning. We evaluate both pipelines using standard retrieval metrics, comparisons with state-of-the-art systems such as GPT and Gemini, and qualitative validation by a domain chemist. GraphRAG achieves higher precision and interpretability, while VectorRAG provides broader recall, highlighting complementary trade-offs. By grounding responses in explicit evidence, the framework enables trustworthy literature analysis at scale. More broadly, this work establishes a practical foundation for building materials science assistants using curated corpora and retrieval design, reducing reliance on proprietary models.

INTRODUCTION

Data has long been recognized as a cornerstone of materials informatics [1–3], and this has become increasingly evident as recent advances in the field continue to rely on data-driven techniques. From algorithm design to autonomous discovery workflows, recent breakthroughs have been enabled not only by improvements in machine learning, but also by the availability of large, high-quality materials datasets. For many polymer systems, however, such datasets remain limited or unavailable. Instead, relevant information is primarily reported through a vast, heterogeneous, and largely unstructured body of literature that continues to expand at an unprecedented pace, with decades of scientific knowledge dis-

* rampi.ramprasad@mse.gatech.edu

persed across text, figures, and tables [4]. As a result, much of this information is difficult to retrieve, integrate across studies, or use effectively for scientific reasoning.

Previous efforts have sought to tap into this reservoir [5–9], producing structured data that can support downstream tasks such as property prediction [10, 11], synthesis planning [12, 13], and materials design [14, 15]. While these databases provide valuable static snapshots of extracted knowledge, they do not fully address how polymer scientists interact with the literature in practice. Scientific inquiry often requires dynamically retrieving evidence across studies, reconciling inconsistent terminology, and generating results reported under varying experimental contexts. Consequently, even seemingly straightforward questions such as "What gas permeability values have been reported for polyhydroxyalkanoate films under different processing conditions?" or "What products form during PHB hydrolysis under acidic versus alkaline conditions?" cannot typically be answered by consulting a single paper or database record. Instead, addressing such queries requires identifying relevant experimental contexts and integrating fragmented findings reported across multiple sources, underscoring the need for systems that support literature-grounded reasoning rather than isolated fact extraction.

An effective polymer literature scholar must therefore go beyond extraction to (i) retrieve relevant evidence across papers, (ii) synthesize information while preserving experimental context, (iii) appropriately defer when the literature does not provide sufficient evidence, and (iv) explicitly ground conclusions in verifiable source material so that domain experts can assess reliability and trustworthiness.

Large language models (LLMs) offer powerful capabilities for synthesizing dispersed scientific information, owing to their ability to leverage knowledge learned from vast text corpora. However, a well-documented limitation of LLMs is hallucination, wherein they produce fluent and confident but factually incorrect statements [16, 17]. Recent work [18] shows that hallucinations arise because LLMs are optimized to complete text rather than to validate information, and standard training objectives reward confident answers even when the underlying evidence is weak. This behavior becomes particularly concerning in scientific settings, where incorrect statements can be difficult to detect without explicit grounding. To miti-

gate hallucinations and improve reliability, researchers have explored a range of strategies, including prompt engineering, refining the model's output during generation, self-feedback mechanisms, and retrieval-augmented generation (RAG) approaches [19–22]. In RAG systems, models retrieve relevant information from trusted external sources and condition their responses on this evidence, reducing unsupported claims while improving transparency and traceability; making RAG particularly well suited for scientific domains.

RAG methods have evolved rapidly in recent years, driven by advances that integrate retrieval more tightly with generation and enable models to incorporate external knowledge more effectively [23]. These developments have produced two prominent retrieval paradigms: vector-based dense retrieval and graph-based reasoning [24]. Vector-based RAG (VectorRAG) represents information as dense embeddings in a high-dimensional space and retrieves semantically similar passages through similarity search, with extensive work focused on optimizing chunking, embedding models, indexing, and scoring mechanisms [25]. Graph-based RAG (GraphRAG), in contrast, organizes knowledge as a network of interconnected entities and relations, enabling retrieval through graph traversal or subgraph extraction. This structured representation supports multi-hop reasoning, improves retrieval precision in domains with complex relational dependencies, and produces evidence trails that are more interpretable [26]. Together, these complementary retrieval paradigms provide the foundations needed to support reliable and grounded LLM reasoning in materials science, where both accurate evidence retrieval and interpretable relational reasoning are essential.

Adapting these retrieval strategies to materials science introduces unique challenges. Scientific text in the field is highly heterogeneous, marked by complex jargon, nested property–composition–processing relationships, and inconsistencies in nomenclature. For example, the same polymer may be described as "PLA," "poly(lactic acid)," or "polylactide," making it harder to bring together and compare results reported across different studies. These challenges are particularly pronounced for polyhydroxyalkanoates (PHAs), a diverse and rapidly expanding family of biodegradable polymers of significant interest for sustainable materials design [27]. As such, PHAs provide a compelling test case for evaluating complementary retrieval strategies such as VectorRAG and GraphRAG.

In this work, building on the framework illustrated in Fig. 1, we introduce the Polymer Literature Scholar, a RAG system designed to support literature-grounded reasoning for polymer science. The framework is designed around the specific challenges of polymer literature, where heterogeneous terminology, distributed experimental evidence, and inconsistent material naming make it difficult to connect scientific claims to their original sources. We develop and systematically benchmark VectorRAG and GraphRAG pipelines on a curated corpus of over 1,000 full-text PHA articles, tailoring each approach to the domain. VectorRAG preserves paragraph-level scientific context through section-aware chunking, while GraphRAG uses entity extraction, normalization, and canonicalization to connect semantically equivalent entity mentions into a structured graph representation.

To evaluate these retrieval pipelines in a domain where no PHA-specific benchmark for literature-grounded reasoning was available, we manually constructed a 113-question evaluation set from a subset of full-text PHA articles. This benchmark was used to assess retrieval quality and factual grounding, and was complemented by qualitative expert analysis spanning general scientific queries, paper-specific questions, and multi-paper reasoning tasks. The responses generated by five representative systems: GraphRAG with GPT-4o-mini, GraphRAG with Llama-3.1-70B, VectorRAG with GPT-4o-mini, OpenAI's built-in RAG with ChatGPT-5 (Web Search), and Google's built-in RAG with Gemini were evaluated based on content accuracy, citation completeness, and depth of scientific reasoning. The full set of expert-evaluation questions and representative system responses are provided in the Supporting Information.

To support practical use and expert assessment of the system, we also developed an interactive interface exposing both the VectorRAG and GraphRAG retrieval pipelines and their underlying evidence. This interface allows domain experts to query the system, inspect the retrieved paragraphs or knowledge graph tuples, verify evidence provenance, and assess the credibility and completeness of the generated responses.

Taken together, this work presents a transparent and reproducible framework for polymer literature reasoning that brings together curated corpus construction, domain-adapted vector and graph retrieval, benchmark-based evaluation in a single workflow. By grounding

responses in a focused PHA corpus and exposing supporting evidence at the level of retrieved paragraphs or knowledge graph tuples, Polymer Literature Scholar enables synthesis across studies while preserving provenance, citation awareness, and the ability to abstain when the available evidence is insufficient. This provides a practical and interpretable foundation for reasoning over complex and inconsistently reported polymer data, making literature-driven polymer research more transparent, verifiable, and credible.

RESULTS AND DISCUSSION

Overview of the Retrieval-Augmented Reasoning Pipelines

Fig. 1 provides an overview of the Polymer Literature Scholar, a retrieval-augmented framework and interactive application designed to support scientific reasoning over the experimental literature on PHAs. Rather than retrieving isolated facts, the system enables the synthesis of coherent, literature-grounded explanations that reflect how polymer properties, processing behavior, and structure–property relationships are discussed and interpreted across multiple studies.

As shown in Fig. 1a, the framework is built on a curated corpus of 1,028 full-text journal articles focused on PHA chemistry, processing, and properties. These articles are parsed into 44,609 paragraph-level text segments, which serve as the fundamental units of scientific evidence. This paragraph-level representation preserves experimental context that is central to how materials scientists reason about polymer behavior. To enable effective access to this literature, the Polymer Literature Scholar supports two complementary retrieval strategies: VectorRAG and GraphRAG.

In the VectorRAG pipeline, paragraph-level text is embedded into a continuous semantic space, allowing retrieval of experimentally relevant passages even when similar concepts are described using different terminology across papers. In contrast, the GraphRAG pipeline organizes extracted entities such as polymer systems, processing steps, and measured properties into a structured representation that explicitly captures relationships reported in the

literature. This graph-based view enables reasoning across interconnected statements, allowing evidence to be integrated from multiple studies that collectively describe a material behavior or trend.

During inference, a user interacts with the Polymer Literature Scholar through a developed application interface by submitting a query. Relevant evidence is retrieved using either of the two retrieval pathway, and the resulting paragraphs or subgraphs are provided as contextual input to a large language model. The model then synthesizes a response grounded in experimentally reported observations drawn directly from the PHA corpus, enabling users to obtain literature-supported explanations.

Fig. 1b illustrates representative example queries and responses generated through the interface. Rather than citing a single source, each response reflects aggregated evidence from multiple relevant studies, as indicated by the document symbol in the figure. This mode of aggregation mirrors how materials scientists typically engage with the literature, by integrating observations across papers to develop a consensus understanding of material behavior, while maintaining transparency about the evidentiary basis of each response. As a result, the framework allows scientists to query the literature in a way that aligns with experimental practice, enabling literature-grounded insights into polymer properties, processing considerations, and design choices that are difficult to obtain using conventional keyword search or database-driven tools.

VectorRAG: Dense Semantic Retrieval Pipeline

The VectorRAG framework represents the literature corpus in a high-dimensional semantic space capable of capturing contextual similarity across studies. As illustrated in Fig. 2, the workflow integrates three main components: knowledge base construction, semantic similarity–based retrieval, and grounded answer generation.

In the knowledge base construction stage (Fig. 2a), the 44,609 parsed paragraphs are grouped into context-preserving text chunks by concatenating adjacent paragraphs from the same section or subsection. This approach ensures that text segments describing related

concepts such as synthesis steps and resulting properties remain contiguous, and results in 10,665 context-preserving text chunks. These chunks are embedded using the *Qwen3-Embedding-4B* model to produce 3,584-dimensional dense vectors, which are then stored in a relational vector database for efficient similarity-based retrieval.

During query processing (Fig. 2b–c), a user query is embedded into the same vector space and compared with all corpus embeddings using cosine similarity. The number of retrieved segments (k) was varied between 2 and 8. The final value of $k = 8$ was selected based on the 21-DOI validation benchmark and full-corpus scaling analysis, where it provided the best practical trade-off between retrieval fidelity, downstream answer accuracy, inference latency, cost, and context-window usage (ref. Supporting Information). The retrieved chunks are concatenated with the user query and provided to an LLM, which generates a literature-grounded response supported by explicit citations (Fig. 2d).

This dense semantic retrieval approach effectively captures implicit conceptual relationships and broader narrative context, making it particularly effective for descriptive or exploratory questions such as "What factors influence the crystallinity of PHBV copolymers?" However, at the same time, the unstructured nature of retrieved text can introduce redundancy and higher token utilization during inference. VectorRAG therefore provides a strong semantic retrieval baseline that complements the structured, relational reasoning capabilities of GraphRAG.

GraphRAG: Knowledge Graph-Based Retrieval Pipeline

The GraphRAG framework restructures the parsed corpus into an explicit knowledge graph (KG), enabling interpretable, multi-hop reasoning across diverse relationships within scientific text. As illustrated in Fig. 3, the workflow comprises three main components: knowledge graph construction, query interpretation and retrieval, and grounded response generation.

In the backend construction phase (Fig. 3a), the 44,609 parsed paragraphs are processed through an entity extraction and normalization pipeline, as described in the Methods section.

Several LLMs were evaluated, including GPT-4o, GPT-4o-mini, GPT-4.1-mini, Llama-3.1-70B, and Llama-3.3-70B, to assess how model choice influences tuple quality and downstream retrieval. Among these, GPT-4o-mini and Llama-3.1-70B produced the most reliable tuples and were therefore selected for the final pipeline. The extraction prompts and all variations tested during this stage are provided in the Supporting Information.

Consequently, GPT-4o-mini extracted 390,864 relational tuples and Llama-3.1-70B extracted 311,475 tuples. These tuples capture scientific relationships such as `[PHB-has property-Tg]` or `[PHBV-synthesized with-hexanoate]`. Because polymers and related entites often appear under multiple textual variants, an entity normalization step was introduced to merge equivalent mentions and increase graph connectivity. This was applied to merge equivalent mentions and render the graph more interconnected. This step yielded 36,757 canonical entities for GPT-4o-mini and 37,289 for Llama-3.1-70B, improving consistency and retrieval quality across the corpus. All resulting tuples were stored in a relational database, creating a structured, queryable representation of scientific knowledge derived from the PHA literature.

During query processing (Fig. 3b), each user question is decomposed into its key entities and relations using an entity and keyword parser. GraphRAG retrieves relevant subgraphs by combining canonical entity matching with string and semantic similarity, as described in the Methods section. The retrieved tuples are then re-ranked through the path-reranking strategy and capped at a maximum subgraph size of 300 nodes. These subgraphs often contain multiple paths linking related entities, thereby enabling multi-hop reasoning across studies. For example, a query such as "How does 3HV content affect the melting temperature of PHBV copolymers?" triggers traversal from $PHBV \rightarrow 3HV \rightarrow Tm$, retrieving all relevant experimental contexts associated with this relationship.

As shown in Fig. 3c, the graph consolidates textual variations into canonical nodes, where contextually related entities (e.g., "3-armed PHB," "malleated PHB," and "PHB–Ag") are unified under a single canonical entity, PHB. The right-hand panel of Fig. 3c highlights the top canonical nodes ranked by cluster size, corresponding to the most frequently discussed entities, such as PHB, PHBV blends, crystallization temperature, and degradation behavior.

Finally, the retrieved subgraph context is concatenated with the user query and passed to the LLM for citation-grounded response generation (Fig. 3d). By representing evidence as explicit relational tuples, GraphRAG provides a transparent reasoning chain that enhances explainability and traceability. Compared with VectorRAG, this structured retrieval paradigm emphasizes relational precision and reasoning depth, making it particularly effective for mechanistic or comparative questions that require multi-hop inference across studies.

Baseline Human Effort

The main manual effort was concentrated in two stages. First, for corpus construction, we used keyword-based filtering on "web of science" to obtain an initial set of approximately 11,000 PHA-related candidate papers. These were then manually screened at the title level to remove papers that were not directly relevant to PHAs, resulting in the final corpus of 1,028 full-text papers. Second, for benchmark construction, 113 evaluation questions were manually curated after reading 21 representative PHA papers. These questions were designed to cover general, paper-specific, and multi-paper reasoning tasks. This benchmark preparation required approximately three weeks.

After the final corpus was defined, the remaining steps were automated. Full-text downloading and parsing were performed using our previously developed polymer literature parsing pipeline [5]. Paragraph extraction, section-aware chunking, embedding generation, vector database construction, knowledge-graph tuple extraction, entity normalization, retrieval, and answer generation were then carried out automatically by the VectorRAG and GraphRAG pipelines. Therefore, the main human input required to reproduce or extend this framework lies in defining a domain-relevant corpus and constructing a representative evaluation benchmark. Once these are established, the downstream processing and retrieval workflows are automated and can be reused for other materials domains.

Ingredients of a Well-Curated Corpus

By a "well-curated corpus" we do not simply mean a large collection of papers. Rather, we use this term to refer to a corpus that is (i) domain-relevant, (ii) context-preserving, and (iii) terminology-normalized. In the present work, domain relevance was ensured by keyword filtering the broader polymer literature corpus and then manually screening the candidate records to obtain the final set of 1,028 PHA-relevant full-text papers. Context preservation was addressed in VectorRAG by grouping adjacent paragraphs according to section and subsection structure, rather than splitting text arbitrarily. Terminology normalization was addressed in GraphRAG through entity canonicalization, which merges related or equivalent material mentions and improves graph connectivity.

We also clarify how this corpus quality was evaluated. The adequacy of the curated corpus was tested indirectly through retrieval-level and answer-level evaluation. In the controlled evaluation, both retrieval pipelines achieved high Recall and Accuracy. At the full corpus scale, performance remained high, especially for GraphRAG, indicating that the corpus representation remained useful even when the retrieval space became much larger and more heterogeneous. Thus, the corpus was tested not only by manual screening, but also by whether it supported successful retrieval and answer generation in both controlled and scaled evaluations.

To make the work more useful for readers who wish to build a similar scholar in another domain, we suggest practical corpus-curation guidelines. Researchers should: (1) define the scientific scope and search terms carefully; (2) filter the literature using domain-specific keywords; (3) screen papers for relevance and ensure that the selected papers can be parsed into clean text. Once such a corpus and benchmark are defined, the downstream VectorRAG and GraphRAG processing steps may be automated and can be reused for other materials domains.

Benchmarking and Evaluation

Controlled Evaluation: Benchmarking Retrieval Fidelity

As an initial test, a subset of 21 PHA-related DOIs was used to develop and optimize both retrieval pipelines. This controlled setup ensured that the RAG framework could effectively retrieve relevant information from a focused dataset where every paper was directly related to the query topics. A total of 113 questions were constructed manually after reviewing the full texts of these 21 DOIs to evaluate retrieval fidelity using the Recall and Accuracy metrics described in the Methods section.

The 21 full-text articles were parsed into 943 paragraphs, which were embedded and stored in the vector database for the VectorRAG pipeline. In parallel, entity extraction and normalization produced 10,814 raw tuples, consolidated into 5,904 canonical entity nodes using GPT-4o-mini, and 8,744 raw tuples consolidated into 4,973 canonical nodes using Llama-3.1-70B, for the GraphRAG pipeline. These datasets formed the benchmark for testing retrieval precision and grounding quality prior to scaling up to the complete corpus.

TABLE I. **Controlled Evaluation**: Reported metrics include recall, accuracy, and average per-query response time and cost for GraphRAG and VectorRAG on a subset of 21 PHA DOIs, using optimized context limits of 300 tuples and 8 paragraphs, respectively.

Models	**Context limit**	**Recall**	**Accuracy**	**Avg. response time (s)**	**Avg. total cost ($)**
GraphRAG					
GPT-4o-mini	300 tuples	0.982	0.991	7.18	0.00097
Llama-3.1-70B	300 tuples	0.991	0.991	10.23	0.00000
VectorRAG					
GPT-4o-mini	8 paragraphs	0.973	1.000	16.43	0.00186
Llama-3.1-70B	8 paragraphs	0.973	1.000	41.75	0.00000

As shown in Table I, both retrieval pipelines performed exceptionally well on the focused 21-DOI test set, achieving near-perfect recall across all models. The high recall values indicate that, when the corpus is narrowly defined and thematically consistent, both dense (VectorRAG) and structured (GraphRAG) retrieval frameworks can successfully retrieve all

relevant information. GraphRAG achieved faster retrieval and lower computational cost owing to its compact, relational representation, whereas VectorRAG exhibited higher latency primarily due to processing substantially longer text inputs, often approaching the 8,192-token context limit of the model. These results demonstrate that both pipelines can effectively ground model responses under controlled conditions and establish a reliable baseline for subsequent large-scale evaluations.

Scaled Evaluation: Retrieval Performance at Corpus Scale

To assess scalability and generalization beyond the controlled benchmark, both retrieval pipelines were evaluated on the complete PHA corpus comprising of 1,028 full-text journal articles. Performance was measured using recall and accuracy, complemented by average per-query response time and total inference cost to capture efficiency, as summarized in Table II.

When scaled to the full corpus, both pipelines exhibited a decline in recall, reflecting the increased difficulty of retrieving the precise context as the search space expanded. The reduction was more pronounced for VectorRAG, with a recall of 0.717, suggesting that dense semantic retrieval becomes less discriminative when the number of embeddings increases and contextual overlap grows. In contrast, GraphRAG maintained substantially higher recall, 0.938 with GPT-4o-mini, benefiting from its structured, entity-linked retrieval that confines searches to semantically consistent regions of the knowledge graph. Despite these differences, the overall answer accuracy remained consistently high for both frameworks, approximately 0.96-0.97, indicating that even when the expected paragraph was not retrieved, the pipeline often produced correct, literature-grounded responses. This observation highlights that retrieval precision alone does not fully dictate downstream answer quality. Instead, GraphRAG's relational representation provides greater robustness and evidence traceability at scale. Here, robustness refers to the retention of retrieval and answer quality when moving from the controlled benchmark to the full corpus. Under this definition, GraphRAG showed a smaller decline in recall than VectorRAG while maintaining comparable answer accuracy.

Evidence traceability refers to the ability to inspect the retrieved subject–relation–object tuples, canonical entity matches, and source-linked evidence used to generate each answer, making the graph-based pipeline more fine-grained and verifiable.

TABLE II. **Scaled Evaluation**: Reported metrics include recall, accuracy, and average per-query response time and cost for GraphRAG and VectorRAG on 1028 PHA DOIs, using optimized context limits of 300 tuples and 8 paragraphs, respectively.

Models	**Context limit**	**Recall**	**Accuracy**	**Avg. response time (s)**	**Avg. total cost ($)**
GraphRAG					
GPT-4o-mini	300 tuples	0.938	0.973	34.14	0.00070
Llama-3.1-70B	300 tuples	0.903	0.964	24.18	0.0
VectorRAG					
GPT-4o-mini	8 paragraphs	0.717	0.960	17.69	0.00184
Llama-3.1-70B	8 paragraphs	0.717	0.960	40.35	0.0

These results also point to different practical scaling considerations for the two retrieval strategies. VectorRAG is simpler to construct and preserves rich paragraph-level context, but retrieval becomes more challenging as the number of semantically similar chunks grows. GraphRAG requires additional preprocessing for entity extraction and normalization, but its canonical, relational representation enables more targeted and inspectable retrieval at corpus scale. Additional scalability considerations are provided in the Supporting Information.

Qualitative and Expert Evaluation

Detailed Analysis of Representative Queries

To qualitatively assess retrieval and reasoning behavior, a set of representative polymer science questions were analyzed, as shown in Table III. The table presents side-by-side responses from the GraphRAG and VectorRAG pipelines for questions spanning three distinct reasoning categories: mechanistic understanding, process–structure correlation, and property comparison.

In Query 1, both pipelines produced literature-consistent and chemically accurate answers. However, VectorRAG provided a more detailed mechanistic explanation by elaborating on reaction conditions specifically under acidic and alkaline environments demonstrating its advantage when relevant information is captured at the paragraph level. In contrast, GraphRAG generated a concise but less comprehensive response; while correct, it omitted additional contextual details that would otherwise have contributed to a richer set of graph tuples.

In Query 2, GraphRAG delivered a more structured and hierarchical explanation, identifying five major categories of challenges in producing low-density PLA foams and citing relevant literature. This highlights its strength in multi-hop reasoning and information aggregation, linking dispersed evidence across multiple sources into a coherent narrative. The VectorRAG response, though factually accurate, was comparatively brief and lacked descriptive depth.

In Query 3, VectorRAG outperformed GraphRAG by providing quantitative thermal and mechanical property data for P(3HB) and P(4HB). GraphRAG's response, while limited in scope due to knowledge sparsity within the retrieved subgraph, remained fully factual and free from hallucination, reflecting a key advantage of graph-grounded reasoning in maintaining reliability even when coverage is incomplete. This again emphasizes VectorRAG's ability to leverage paragraph-level semantic understanding to generate more detailed, context-rich answers.

Overall, the complementary performance of the two pipelines underscores their distinct retrieval paradigms: GraphRAG enhances interpretability through structured reasoning and contextual aggregation while maintaining factual precision, whereas VectorRAG excels in producing fluent, paragraph-level explanations when detailed textual information is present in the corpus.

TABLE III: Qualitative comparison of GraphRAG and VectorRAG responses to representative polymer science questions, illustrating differences in retrieval behavior, reasoning depth, and grounding style between the two pipelines.

Question	**GraphRAG Response**	**VectorRAG Response**
1. What are the main monomeric products of abiotic PHB hydrolysis and under what conditions are they typically formed?	The main monomeric products of abiotic PHB hydrolysis are (R)-3-HBA and CA [4]. This process typically occurs under conditions of 70–220 °C [5].	The main monomeric products of abiotic PHB hydrolysis are D-(-)-3-hydroxybutyric acid (3HBA) and crotonic acid (CA). These products are typically formed under acidic or alkaline conditions, where the ester linkages in the polymer chain undergo hydrolysis. Specifically, at neutral pH (7.4), 3HB is a major intermediate product of PHB degradation [1]. Under acidic conditions (0.1–4.0 M H), 3HBA and CA are formed, with CA being the dominant monomeric product in highly concentrated acid solutions (80–98 wt% H2SO4) [2]. Conversely, alkaline conditions (0.1–4.0 M OH-) result in higher amounts of both 3HB and CA.

Question	GraphRAG Response	VectorRAG Response
2. What challenges are associated with producing low-density PLA foams, and how can these be addressed?	Producing low-density PLA foams presents several challenges, primarily related to the material's intrinsic properties and processing conditions. 1. **Melt Strength and Elasticity:** PLA exhibits low melt elasticity and low melt strength, which complicates the production of low-density foams. These properties can hinder the ability to achieve the desired foam structure during processing [211][212]. 2. **Crystallinity and Foamability:** High crystallinity in PLA can limit the expansion of the produced foam, reducing its foamability [200]. Additionally, slow crystallization kinetics contribute to the challenges in achieving uniform cell structures and desirable densities [266]. 3. **Cell Structure Uniformity:** Achieving uniform cell structures in large-scale settings is particularly challenging for low-density PLA foams. This uniformity is critical for maintaining consistent mechanical properties throughout the foam [213]. 4. **Surface-to-Volume Ratios:** Producing foams with high surface-to-volume ratios is another challenge, as it impacts the foam's overall performance and applications [214]. 5. **Processing Variables:** The interaction between material properties and processing conditions is crucial in determining foam characteristics. Variations in pressure, temperature, and the type of foaming agents used can significantly affect the final foam structure [220][250]. To address these challenges, several strategies could be employed: – **Additives and Blending:** Incorporating additives or blending PLA with other polymers, such as PHBV, can enhance the foaming properties and mechanical strength of the resulting foams [18][130]. **Optimized Processing Techniques:** Utilizing advanced processing techniques, such as film/foam coextrusion, may enhance the production of low-density PLA foams by allowing better control over processing parameters [21][264].	Producing low-density PLA foams is challenged by factors such as PLA's poor melt strength and low melt elasticity, which lead to inadequate foaming behavior without modifications. Additionally, the crystallization behavior of PLA can hinder foaming, particularly at higher crystallization rates, which can prevent effective cell formation and growth. The CO_2 induced crystallization at high pressures can further complicate the foaming process by increasing crystalline regions that reduce foamability. These challenges can be addressed by blending PLA with polyhydroxybutyrate (PHBV), which improves the rheological properties and enhances foamability. PHBV acts as a nucleating agent for PLA, potentially leading to better cell structure and reduced density in the foams. Adjusting the composition of the blend and optimizing processing conditions, such as foaming temperature and pressure, can also help achieve the desired foam properties while minimizing the adverse effects of crystallization and melt strength limitations [1][2][4][8].

Question	GraphRAG Response	VectorRAG Response
3. How do the physical and thermal properties of P(3HB) and P(4HB) differ and what are the implications of these differences for their potential applications?	The knowledge graph does not provide specific details on the physical and thermal properties of P(3HB) and P(4HB) directly. However, it does mention the analysis of PHB, which is related to P(3HB) and has implications for thermal behavior and processing. 1. PHB (a homopolymer of 3HB) is analyzed by thermal analysis to understand its properties [4]. 2. The difference between melting and thermal degradation temperatures affects the processability of PHB production [20]. 3. Blending with other polymers can improve mechanical properties, flexibility/ductility, and processability of PHAs [14]. While these points provide some context about the thermal properties of PHB, they do not directly compare P(3HB) and P(4HB) or provide their specific physical properties and implications for applications. Therefore, there is insufficient information in the knowledge graph to answer the question regarding the differences between P(3HB) and P(4HB) and their implications comprehensively.	The physical and thermal properties of P(3HB) and P(4HB) differ significantly, which has implications for their potential applications. P(3HB) is a highly crystalline, water-insoluble polymer with a high Young's modulus (3.5 GPa) and low tensile strength (43 MPa). It is brittle, with an elongation at break of only 5%, making it less suitable for applications requiring flexibility [3]. Its melting temperature is around 180 °C, and it possesses a glass transition temperature of approximately 4 °C [5]. These characteristics limit its use primarily to applications where rigidity and stability are needed. In contrast, P(4HB) is a strong, malleable thermoplastic with a tensile strength comparable to polyethylene and an elongation at break of 100% [4]. This high elasticity makes P(4HB) suitable for applications requiring flexibility and resilience. Its thermal properties include a lower melting point of about 164 °C and a glass transition temperature of 3 °C [6], which allows for easier processing and adaptability in various applications. The differences in mechanical properties imply that while P(3HB) may be used in rigid applications, such as packaging or structural components, P(4HB) is better suited for applications requiring elasticity and flexibility, such as in biomedical devices or flexible films. Additionally, the ability to create copolymers, such as P(3HB-co-4HB), allows for tailored properties that can merge the strengths of both polymers for diverse applications [8].

Expert Validation

Beyond retrieval fidelity, the systems were evaluated in a realistic expert-use setting. A useful polymer literature assistant must not only retrieve relevant evidence, but also synthesize it into scientifically sound answers with clear and verifiable citations. To assess

this capability, a domain chemist independently evaluated the RAG pipelines using the interactive interface developed in this work. The evaluation focused on factual grounding, contextual coverage, and citation reliability across different question types.

The comparison included two GraphRAG variants, built with GPT-4o-mini and Llama-3.1-70B, along with VectorRAG, ChatGPT-5 with built-in Web Search, and Gemini with built-in Web Search. These systems represent two complementary comparison settings. VectorRAG and GraphRAG form the corpus-matched comparison, as both operate over the same curated PHA corpus, use the same question set, and generate answers from retrieved corpus evidence. ChatGPT-5 and Gemini, in contrast, were included as practical web-enabled comparators, reflecting tools that researchers may use for literature-oriented search, evidence gathering, and citations. These systems were not intended to serve as algorithmically equivalent baselines, since they retrieve from broader web sources through proprietary search mechanisms rather than from our curated corpus.

Each model was tested on a three-tier question set, General, Paper-specific, and Multi-paper, designed to reflect the increasing complexity of scientific reasoning, from conceptual clarification to interpretation of individual studies and finally to synthesis across multiple publications. Details of the question set are provided in the Supporting Information. For each answer, the expert assigned a score from one to ten, with five points allocated to content quality and factual correctness and five points allocated to citation quality and relevance.

As shown in Fig. 4, GraphRAG powered by GPT-4o-mini and ChatGPT-5 with Web Search achieved the highest overall scores, averaging between nine and ten across all question categories. Both demonstrated strong factual grounding, coherent reasoning, and reliable citation behavior. The GraphRAG variant built on Llama-3.1-70B scored slightly lower but maintained consistent factual precision, illustrating how language-model capacity influences the quality and completeness of the knowledge-graph tuples and subsequently the downstream answer quality. VectorRAG and Gemini with Web Search produced moderate yet stable performance across all categories as well. A detailed comparison of representative responses is provided in the Supporting Information.

Building on these results, the expert evaluation also revealed clear and consistent differ-

ences in how each system reasons over the literature. GraphRAG, particularly when paired with GPT-4o-mini, produced the most precise and well supported answers. Its responses were tightly anchored to the extracted relational evidence and showed very low rates of hallucination, a property that is essential for scientific use. ChatGPT-5, supported by web retrieval, generated more expansive narrative answers, reflecting its broader access to information, but its citations were sometimes less specific than those produced by the in house systems. VectorRAG performed well for questions that depend on paragraph level context, often supplying richer explanations when the needed information was concentrated in a few passages. Gemini offered wider coverage but less reliable referencing due to the variable quality of retrieved web sources. The GraphRAG with Llama-3.1-70B, although more literal, still produced accurate and well grounded answers, demonstrating that the framework remains effective even with smaller open weight models.

An additional and important observation was that the in-house pipelines routinely responded with "I do not know" when the extracted knowledge did not contain sufficient information to answer the question. This behavior stood in contrast to the commercial systems, which were more likely to provide speculative or incomplete answers. The tendency to abstain rather than hallucinate is particularly valuable for scientific applications, where incorrect claims can mislead downstream interpretation.

Taken together, these results show that retrieval pipelines built on a carefully curated knowledge base and domain aware processing can match, and in several respects surpass, the performance of web driven proprietary systems. At the same time, they offer greater degree of transparency, scientific rigor, and substantially lower computational cost. Unlike commercial interfaces that often return links to entire articles, leaving users to manually locate the supporting evidence, the pipelines developed here provide precise evidence trails at the level of individual paragraphs or knowledge graph tuples. This reduces the effort required for verification, strengthens trust in the generated responses, and offers researchers a more reliable foundation for interpreting and analyzing the literature across the polymer domain.

CONCLUSION

Recent advances in LLMs have shown remarkable promise for scientific knowledge discovery, yet their usefulness ultimately depends on the quality and structure of the information they draw from. This work demonstrates that a reliable literature scholar for polymer materials does not require large proprietary systems or opaque retrieval engines. Instead, carefully curated corpora combined with domain-aware retrieval pipelines can provide a foundation that is both trustworthy and tailored to the scientific questions materials researchers need to ask.

By focusing on PHAs as a representative case study, we show howretrieval methods can be adapted to reflect the structure and reporting practices of polymers literature. Dense semantic retrieval through VectorRAG and structured relational retrieval through GraphRAG both benefit from careful corpus construction, systematic transformation of text, and robust entity normalization, which together create the basis required for LLMs to reason reliably over heterogeneous scientific content. These design choices make the resulting answers traceable, interpretable, and scientifically credible.

The results also reveal a division of strengths across the two retrieval pipelines. VectorRAG delivers broad semantic recall and rich paragraph-level context, making it effective for descriptive or exploratory questions. GraphRAG excels at multi-hop, relational reasoning with compact, interpretable evidence trails. At corpus scale, GraphRAG maintains higher recall and lower latency while VectorRAG often provides more detailed narrative answers when the relevant information is localized in text. Together, they highlight the value of combining complementary retrieval paradigms in a single workflow.

Across controlled benchmarks and expert review, both pipelines achieve high accuracy, with GraphRAG in particular matching the performance of commercial web-connected RAG systems on domain questions while exhibiting more conservative, evidence-driven behavior and more consistent citations. In particular, the in-house pipelines more frequently defer with “I do not know” when the available literature does not support a definitive answer, reducing the risk of speculative or unsupported claims. At the same time, the study shows

that conventional retrieval metrics such as recall cannot fully capture the scientific usefulness of a RAG system. Lower recall does not necessarily indicate weak grounding or limited relevance. Human expertise remains essential for determining whether retrieved context is truly relevant, whether reasoning chains are scientifically sound, and whether citations are complete and trustworthy.

Overall, this work establishes a system-level, domain-adapted retrieval framework for building evidence-grounded AI scholars in materials science. By combining curated corpus construction, domain-aware VectorRAG and GraphRAG pipelines, benchmark-based evaluation, and inspectable evidence trails, Polymer Literature Scholar provides a reliable foundation for reasoning over complex and inconsistently reported polymer data. While demonstrated here for PHAs, the same design principles are broadly relevant to other materials domains where scientific insight depends on integrating fragmented literature evidence.

METHODS

Corpus. Our literature corpus comprises $\sim$ 3 million documents obtained from publishers including Elsevier, Wiley, Springer Nature, the American Chemical Society, and the Royal Society of Chemistry, covering articles published up to 2025. Details of the parsing pipeline are provided in [5]. To construct the PHA-specific corpus, we applied a keyword-based filter to the titles and abstracts of all articles, using PHA-relevant terms listed in the Supplementary Information. The filtered results were then manually verified, producing a final set of 1,028 PHA-relevant papers. The paragraphs were subsequently extracted from the full texts of these articles for downstream processing.

Evaluation Metrics. Retrieval performance was evaluated using Recall@K, Recall PID@K and Accuracy. Recall@K is a standard metric in information retrieval for measuring the accuracy and ranking quality of retrieved results. It quantifies whether the expected ground-truth context paragraph appears among the top-K retrieved results.

In scientific literature, relevant information is often distributed across multiple paragraphs

within the same paper. As a result, Recall@K may underestimate retrieval performance when the correct paper is retrieved but the exact ground-truth paragraph is not. To address this limitation, we additionally report Recall PID@K, which considers a retrieval correct if any of the top-K retrieved contexts originate from the ground-truth paper, irrespective of the specific paragraph matched.

Because both Recall@K and Recall PID@K focus solely on retrieval behavior, they may still underestimate the practical usefulness of a RAG system. We therefore also report Accuracy, defined as a binary (0/1) score assigned by a domain expert after assessing the correctness and completeness of the generated answer with respect to the query.

Recall@K. Recall@K measures whether the expected ground-truth paragraph P_{expected} is included within the top-K retrieved paragraphs $P_{\text{retrieved}}$:

$$\text{Recall@K} = \begin{cases} 1, & \text{if } P_{\text{expected}} \in P_{\text{retrieved}}[:K], \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

For N total queries, the mean Recall@K is:

$$\text{Mean Recall@K} = \frac{1}{N} \sum_{i=1}^{N} \text{Recall@K}_i. \tag{2}$$

Recall PID@K. Recall PID@K measures whether the paper containing the ground-truth paragraph, identified by its Paragraph ID ($\text{PID}_{\text{expected}}$), is included within the top-K retrieved papers ($\text{PID}_{\text{retrieved}}$):

$$\text{Recall PID@K} = \begin{cases} 1, & \text{if } \text{PID}_{\text{expected}} \in \text{PID}_{\text{retrieved}}[:K], \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The mean Recall PID@K is computed analogously to Equation (2).

Knowledge Graph Construction Phase

Entity Extraction. Knowledge graph tuples were extracted from the corpus using LLMs via in-context learning. Each extracted tuple contains five fields: *(subject, relation, object, reference_relation, reference_node)*. When available in the source text, explicit citations and figure/table references are also recorded. The optimized prompt used for tuple generation is provided in the Supplementary Information.

Entity Canonicalization. To unify semantically equivalent entities among the extracted tuples, 384-dimensional sentence embeddings were generated using the `all-MiniLM-L6-v2` model from the SentenceTransformers library. A two-stage hybrid clustering strategy was employed to balance scalability and precision. First, MiniBatchKMeans was used to perform coarse-grained grouping of entities into 2,000 clusters. Within each cluster, AgglomerativeClustering with average linkage was applied to perform fine-grained merging based on a cosine distance threshold of 0.5.

For each resulting cluster, the canonical entity was selected using centroid-based ranking, where the entity closest to the cluster centroid was chosen as the representative label. For clusters dominated by numerical entities (defined as clusters with over 60% numeric content), a synthetic canonical label of the form `numerical_value` was assigned. This canonicalization step reduces vocabulary fragmentation while preserving semantic relationships, thereby improving graph connectivity and retrieval robustness.

GraphRAG Retrieval Phase

Query Preprocessing. User queries undergo a preprocessing step to extract domain-relevant terms and normalize textual variations prior to retrieval. Queries are first lowercased and processed using tokenization and lemmatization implemented with the NLTK framework. A custom regular expression pattern is then applied to identify domain-specific entities, including polymer names, chemical formulas, composite material notations, and numerical values. Stop words are removed while preserving scientifically meaningful terms. This

preprocessing standardizes query representations and facilitates robust matching against entities in the knowledge graph.

String-Based Retrieval. String-based retrieval identifies candidate knowledge graph tuples through exact keyword matching. Regular expressions with word-boundary constraints are applied to the subject, relation, and object fields of each tuple, with matching performed in a case-insensitive manner. Tuples containing all query keywords are prioritized. If no such tuples are found, a relaxed matching strategy is applied, requiring at least two keyword matches. Retrieved tuples are ranked using a frequency based scoring scheme that prioritizes tuples matching a larger number of query terms.

Canonical-Based Retrieval. To account for lexical variation in the literature, semantic retrieval is performed using canonical entity embeddings produced as part of the entity normalization step. Query terms are embedded using the `all-MiniLM-L6-v2` model and compared against precomputed canonical entity embeddings using cosine similarity. Canonical entities exceeding a similarity threshold of 0.7 are retained, and all tuples containing these entities are retrieved. Exhaustive similarity comparison across the canonical entity set ensures coverage of semantically related tuples that may not be captured through string-based matching alone.

Hybrid Retrieval and Re-ranking. Hybrid retrieval integrates string-based and canonical-based matching to leverage both exact keyword overlap and semantic similarity. For each tuple, a hybrid score S_{hybrid} is computed as a weighted combination of the string-match score S_{string} and the canonical-match score $S_{\text{canonical}}$:

$$S_{\text{hybrid}} = \alpha \cdot S_{\text{canonical}} + (1 - \alpha) \cdot S_{\text{string}}, \tag{4}$$

where $\alpha = 0.7$, reflecting greater emphasis on semantic similarity. Hybrid scores are min–max normalized, and tuples with S_{hybrid} below a threshold of $\tau = 0.6$ are filtered out.

Path Re-ranking. The retained candidate tuples after hybrid matching are re-ranked using a cross-encoder model, `cross-encoder/ms-marco-MiniLM-L-6-v2`, which assigns a re-ranking score S_{rerank} by jointly encoding each query–tuple pair. Unlike entity-level matching, this path-level evaluation considers the complete relational path (subject, relation, object) in the context of the query, enabling the model to assess semantic coherence and relevance across the full tuple. This step improves retrieval precision by prioritizing tuples that more directly and collectively address the query.

The final ranking score is computed as:

$$S_{\text{final}} = \lambda \cdot S_{\text{rerank}} + (1 - \lambda) \cdot S_{\text{hybrid}}, \tag{5}$$

where $\lambda = 0.7$ biases the final ranking toward the context-aware re-ranking score.

The number of tuples returned is controlled by the `max_tuples` parameter. To avoid prematurely discarding potentially relevant candidates, the cross-encoder evaluates up to four times `max_tuples`, normalizes the resulting scores to the range $[0, 1]$, and retains the top `max_tuples` tuples. We evaluated `max_tuples` values between 300 and 500 and found that 300 provides the best balance between retrieval quality, computational cost, and inference latency.

VectorRAG

Embedding Model. VectorRAG employs the `Qwen/Qwen3-Embedding-4B` model from the SentenceTransformers library to generate 3,584-dimensional embeddings for both the corpus and the user query. Corpus embeddings are pre-computed and stored in a PostgreSQL database for efficient retrieval.

Text chunking strategy. Scientific articles are typically organized into major sections such as Introduction, Methods, Results and Discussion, and Conclusion, each of which may contain multiple hierarchical subsections. To preserve the scientific context during retrieval, paragraphs were combined based on this structural hierarchy. Specifically, all

paragraphs belonging to the same first-level subsection (i.e., the lowest common subheading under a major section) were merged into a single chunk. This approach maintained the logical flow and coherence of scientific arguments within a section while optimizing semantic understanding and retrieval performance.

AUTHOR CONTRIBUTIONS

Conceptualization: S.G., R.R., Data Curation: S.G., Methodology: S.G, A.M., Validation: S.G., W.X., Original draft: S.G., Review & editing: S.G., A.M., W.X., R.R.

DATA AVAILABILITY

Data sharing is not applicable to this article as no new data was created or analyzed in this study.

CODE AVAILABILITY

The code used in this work can be found at https://github.com/Ramprasad-Group/rag-polymer-lit-scholar

ACKNOWLEDGEMENT

This work was supported by the National Science Foundation (Grant number: 2515411) and by the Office of Naval Research (Grant number: N00014-19-1-2586).

COMPETING INTERESTS

The authors declare no competing interests.

SUPPORTING INFORMATION

The online version of this article contains Supplementary Information available at https://doi.org/

Includes Optimized prompts for VectorRAG and GraphRAG components, LLM Benchmarking, Domain Expert Evaluation Questions etc.

FIGURE CAPTIONS

Figure 1. (a) Overview of the retrieval-augmented generation (RAG) framework for literature-grounded question answering on PHAs. The curated PHA corpus, comprising 1,028 articles and 44,609 parsed paragraphs, serves as the shared knowledge base for both VectorRAG and GraphRAG retrieval pipelines. Retrieved contextual evidence from each pipeline is provided to a LLM to generate literature-grounded responses. (b) Example scientific queries and corresponding responses generated by the Polymer Literature Scholar. Responses are synthesized by aggregating evidence retrieved from multiple relevant studies within the PHA corpus, as indicated by the document symbol.

Figure 2. VectorRAG workflow for literature-grounded question answering on PHAs. (**a**) Backend processing of the corpus, where full-text articles are parsed, contextually grouped into condensed text chunks, and embedded into a dense semantic space for similarity-based retrieval. (**b**) Retrieval-augmented inference, in which user queries are encoded in the same latent space and matched to the most semantically aligned text segments from the corpus. (**c**) Conceptual representation of the embedding space illustrating how relevant paragraphs are identified through cosine similarity between the query vector and its nearest neighbors. (**d**) Example of a representative query showing retrieved literature passages and the corresponding grounded, citation-linked response generated by the language model.

Figure 3. GraphRAG workflow for knowledge graph–based question answer-

ing on PHAs. (**a**) Backend processing of the corpus, where entities and relationships are extracted from literature, normalized, and stored as relational tuples in a relational database. (**b**) Retrieval and reasoning stage showing how user queries are decomposed into entity–relation pairs, matched to canonical entities, and re-ranked through a path-based scoring strategy to identify the most relevant subgraphs for grounded response generation. (**c**) Example of entity canonicalization, where multiple related mentions (e.g., PHB–Ag, maleated PHB, 3-armed PHB) are merged into a unified canonical node representing PHB. The accompanying bar plot highlights the top canonical entities ranked by cluster size, demonstrating how normalization improves graph connectivity and recall. (**d**) Representative example showing a user query, retrieved knowledge graph tuples, and the corresponding grounded response synthesized by the language model with supporting citations.

Figure 4. Domain-expert evaluation of five RAG pipelines across General, Paper-specific, and Multi-paper questions. Scores reflect how well each system balanced factual grounding, contextual coverage, and citation reliability. GraphRAG (GPT-4o-mini) and ChatGPT-5 with Web Search achieved the highest overall performance, with other pipelines showing moderate but consistent results.

REFERENCES

[1] A. Agrawal and A. Choudhary, Perspective: Materials informatics and big data: Realization of the "fourth paradigm" of science in materials science, Apl Materials **4** (2016).

[2] R. Ramprasad, R. Batra, G. Pilania, A. Mannodi-Kanakkithodi, and C. Kim, Machine learning in materials informatics: recent applications and prospects, npj Computational Materials **3**, 54 (2017).

[3] L. Himanen, A. Geurts, A. S. Foster, and P. Rinke, Data-driven materials science: status, challenges, and perspectives, Advanced Science **6**, 1900808 (2019).

[4] T. B. Martin and D. J. Audus, Emerging trends in machine learning: a polymer perspective, ACS Polymers Au **3**, 239 (2023).

[5] S. Gupta, A. Mahmood, P. Shetty, A. Adeboye, and R. Ramprasad, Data extraction from polymer literature using large language models, Communications materials **5**, 269 (2024).

[6] P. Shetty, A. C. Rajan, C. Kuenneth, S. Gupta, L. P. Panchumarti, L. Holm, C. Zhang, and R. Ramprasad, A general-purpose material property data extraction pipeline from large polymer corpora using natural language processing, npj Comput Mater **9**, 1 (2023).

[7] J. Cheung, Y. Zhuang, Y. Li, P. Shetty, W. Zhao, S. Grampurohit, R. Ramprasad, and C. Zhang, Polyie: A dataset of information extraction from polymer material scientific literature, in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)* (2024) pp. 2370–2385.

[8] P. Kalhor, N. Jung, S. Bräse, C. Wöll, M. Tsotsalas, and P. Friederich, Functional material systems enabled by automated data extraction and machine learning, Advanced Functional Materials **34**, 2302630 (2024).

[9] F. Bai, J. Kang, G. Stanovsky, D. Freitag, M. Dredze, and A. Ritter, Schema-driven information extraction from heterogeneous tables, in *Findings of the Association for Computational Linguistics: EMNLP 2024* (2024) pp. 10252–10273.

[10] S. Gupta, A. Mahmood, S. Shukla, and R. Ramprasad, Benchmarking large language models for polymer property predictions, Macromolecular Rapid Communications , e00388 (2025).

[11] K. Stergiou, C. Ntakolia, P. Varytis, E. Koumoulos, P. Karlsson, and S. Moustakidis, Enhancing property prediction and process optimization in building materials through machine learning: A review, Computational Materials Science **220**, 112031 (2023).

[12] L. Chen, J. Kern, J. P. Lightstone, and R. Ramprasad, Data-assisted polymer retrosynthesis planning, Applied Physics Reviews **8**, 031405 (2021).

[13] E. Kim, K. Huang, A. Saunders, A. McCallum, G. Ceder, and E. Olivetti, Materials synthesis insights from scientific literature via text extraction and machine learning, Chemistry of Materials **29**, 9436 (2017).

[14] A. Savit, H. Sahu, S. Shukla, W. Xiong, and R. Ramprasad, polybart: A chemical linguist for polymer property prediction and generative design, arXiv preprint arXiv:2506.04233 (2025).

[15] H. Sahu, W. Xiong, A. Savit, S. S. Shukla, and R. Ramprasad, An encoder-decoder foundation chemical language model for generative polymer design, arXiv preprint arXiv:2510.18860 (2025).

[16] I. Augenstein, T. Baldwin, M. Cha, T. Chakraborty, G. L. Ciampaglia, D. Corney, R. DiResta, E. Ferrara, S. Hale, A. Halevy, *et al.*, Factuality challenges in the era of large language models and opportunities for fact-checking, Nature Machine Intelligence **6**, 852 (2024).

[17] A. Alansari and H. Luqman, Large language models hallucination: A comprehensive survey (2025), arXiv:2510.06265 [cs.CL].

[18] A. T. Kalai, O. Nachum, S. S. Vempala, and E. Zhang, Why language models hallucinate (2025), arXiv:2509.04664 [cs.CL].

[19] S. M. T. I. Tonmoy, S. M. M. Zaman, V. Jain, A. Rani, V. Rawte, A. Chadha, and A. Das, A comprehensive survey of hallucination mitigation techniques in large language models (2024), arXiv:2401.01313 [cs.CL].

[20] J.-W. Shim, Y.-J. Ju, J.-H. Park, and S.-W. Lee, Multi-stage prompt refinement for mitigating hallucinations in large language models (2025), arXiv:2510.12032 [cs.CL].

[21] A. Madaan, N. Tandon, P. Gupta, S. Hallinan, L. Gao, S. Wiegreffe, U. Alon, N. Dziri, S. Prabhumoye, Y. Yang, S. Gupta, B. P. Majumder, K. Hermann, S. Welleck, A. Yazdanbakhsh, and P. Clark, Self-refine: Iterative refinement with self-feedback (2023), arXiv:2303.17651 [cs.CL].

[22] O. Ayala and P. Bechard, Reducing hallucination in structured outputs via retrieval-augmented generation, in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)* (Association for Computational Linguistics, 2024) p. 228–238.

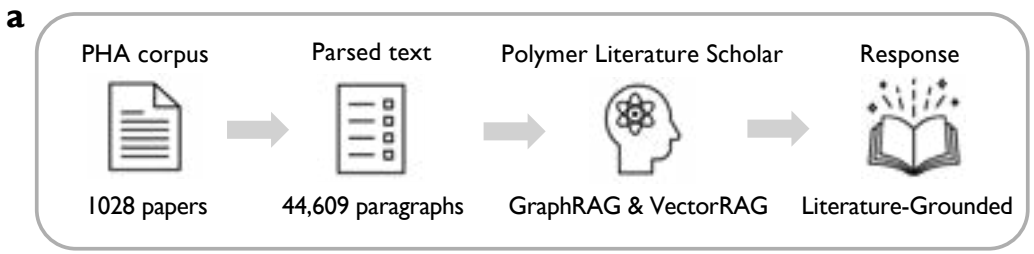
[23] W. Fan, Y. Ding, L. Ning, S. Wang, H. Li, D. Yin, T.-S. Chua, and Q. Li, A survey on rag meeting llms: Towards retrieval-augmented large language models (2024), arXiv:2405.06211 [cs.CL].

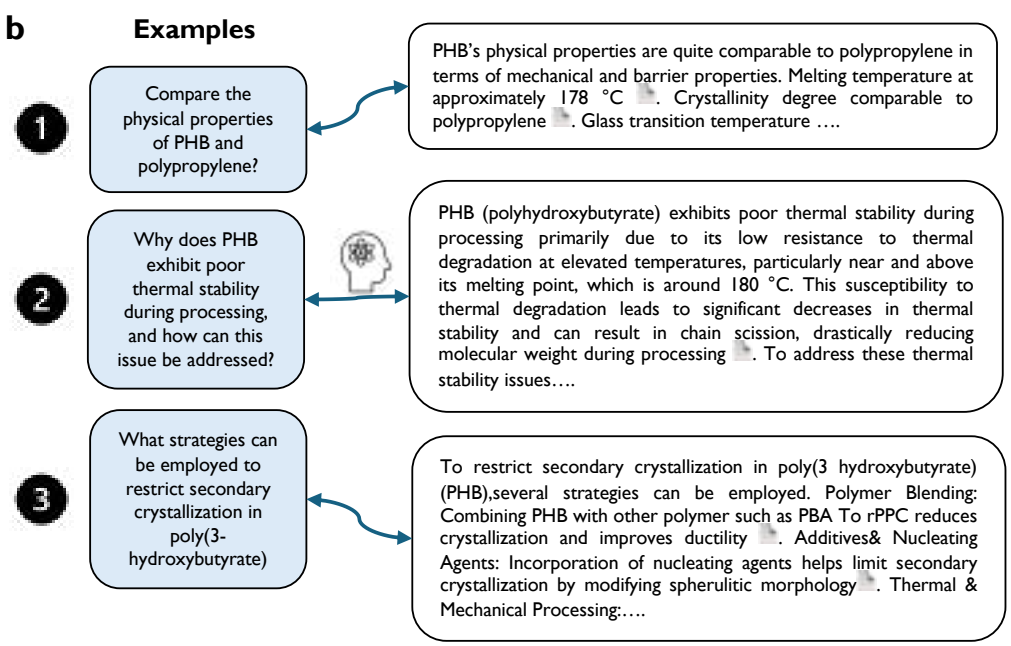
[24] S. Ahmad, Z. Nezami, M. Hafeez, and S. A. R. Zaidi, Benchmarking vector, graph and hybrid retrieval augmented generation (rag) pipelines for open radio access networks (oran), arXiv preprint arXiv:2507.03608 (2025).

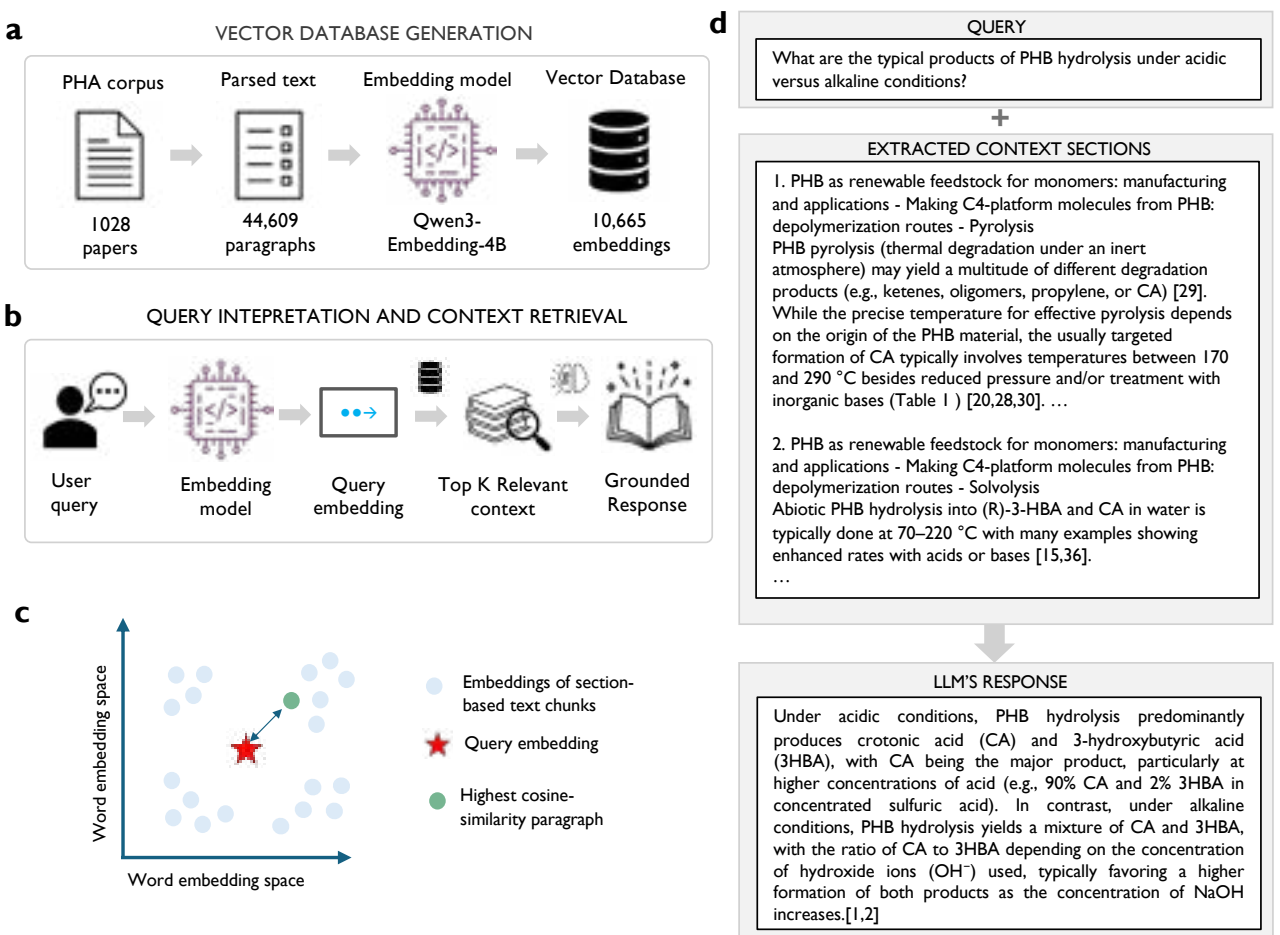
[25] S. Kukreja, T. Kumar, V. Bharate, A. Purohit, A. Dasgupta, and D. Guha, Performance evaluation of vector embeddings with retrieval-augmented generation, in *2024 9th International Conference on Computer and Communication Systems (ICCCS)* (2024) pp. 333–340.

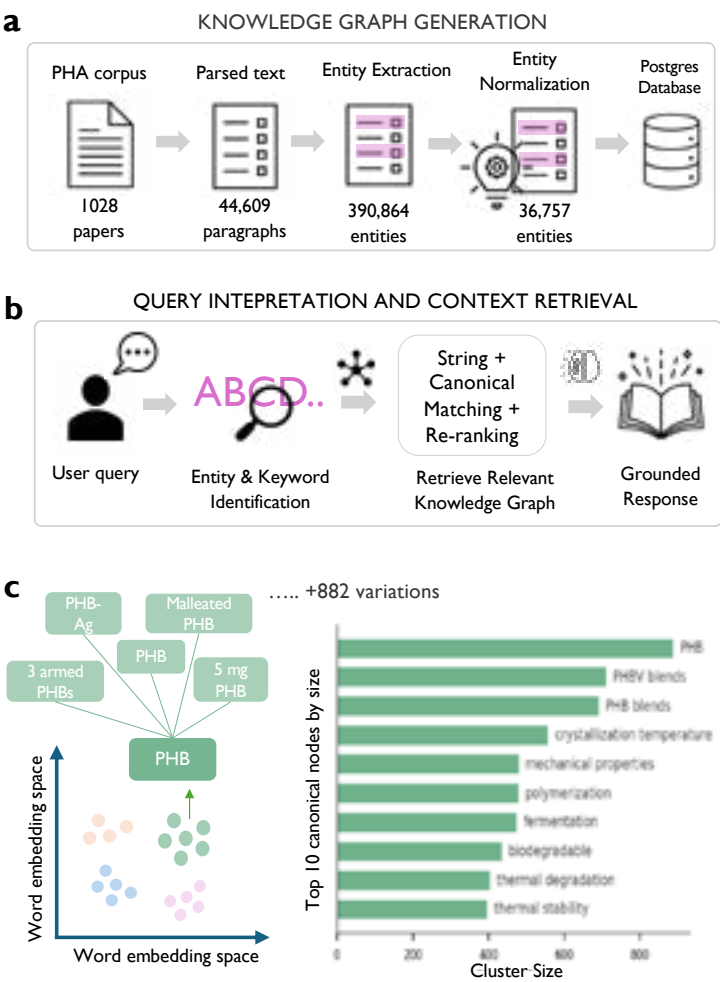
[26] B. Peng, Y. Zhu, Y. Liu, X. Bo, H. Shi, C. Hong, Y. Zhang, and S. Tang, Graph retrieval-augmented generation: A survey (2024), arXiv:2408.08921 [cs.AI].

[27] K. Olonisakin, A. K. Mohanty, M. Thimmanagari, and M. Misra, Recent advances in biodegradable polymer blends and their biocomposites: a comprehensive review, Green Chemistry **27**, 11656 (2025).


a
PHA corpus
Parsed text
Polymer Literature Scholar
Response
1028 papers
44,609 paragraphs
GraphRAG & VectorRAG
Literature-Grounded


b
Examples
1
Compare the physical properties of PHB and polypropylene?
PHB's physical properties are quite comparable to polypropylene in terms of mechanical and barrier properties. Melting temperature at approximately 178 °C. Crystallinity degree comparable to polypropylene. Glass transition temperature ….
2
Why does PHB exhibit poor thermal stability during processing, and how can this issue be addressed?
PHB (polyhydroxybutyrate) exhibits poor thermal stability during processing primarily due to its low resistance to thermal degradation at elevated temperatures, particularly near and above its melting point, which is around 180 °C. This susceptibility to thermal degradation leads to significant decreases in thermal stability and can result in chain scission, drastically reducing molecular weight during processing. To address these thermal stability issues….
3
What strategies can be employed to restrict secondary crystallization in poly(3-hydroxybutyrate)
To restrict secondary crystallization in poly(3 hydroxybutyrate) (PHB),several strategies can be employed. Polymer Blending: Combining PHB with other polymer such as PBA To rPPC reduces crystallization and improves ductility. Additives& Nucleating Agents: Incorporation of nucleating agents helps limit secondary crystallization by modifying spherulitic morphology. Thermal & Mechanical Processing:….


a
VECTOR DATABASE GENERATION
PHA corpus
Parsed text
Embedding model
Vector Database
1028 papers
44,609 paragraphs
Qwen3-Embedding-4B
10,665 embeddings
b
QUERY INTEPRETATION AND CONTEXT RETRIEVAL
User query
Embedding model
Query embedding
Top K Relevant context
Grounded Response
c
Word embedding space
Word embedding space
Embeddings of section-based text chunks
Query embedding
Highest cosine-similarity paragraph
d
QUERY
What are the typical products of PHB hydrolysis under acidic versus alkaline conditions?
+
EXTRACTED CONTEXT SECTIONS
1. PHB as renewable feedstock for monomers: manufacturing and applications - Making C4-platform molecules from PHB: depolymerization routes - Pyrolysis
PHB pyrolysis (thermal degradation under an inert atmosphere) may yield a multitude of different degradation products (e.g., ketenes, oligomers, propylene, or CA) [29]. While the precise temperature for effective pyrolysis depends on the origin of the PHB material, the usually targeted formation of CA typically involves temperatures between 170 and 290 °C besides reduced pressure and/or treatment with inorganic bases (Table 1) [20,28,30]. …
2. PHB as renewable feedstock for monomers: manufacturing and applications - Making C4-platform molecules from PHB: depolymerization routes - Solvolysis
Abiotic PHB hydrolysis into (R)-3-HBA and CA in water is typically done at 70–220 °C with many examples showing enhanced rates with acids or bases [15,36].
…
LLM'S RESPONSE
Under acidic conditions, PHB hydrolysis predominantly produces crotonic acid (CA) and 3-hydroxybutyric acid (3HBA), with CA being the major product, particularly at higher concentrations of acid (e.g., 90% CA and 2% 3HBA in concentrated sulfuric acid). In contrast, under alkaline conditions, PHB hydrolysis yields a mixture of CA and 3HBA, with the ratio of CA to 3HBA depending on the concentration of hydroxide ions (OH^-) used, typically favoring a higher formation of both products as the concentration of NaOH increases.[1,2]



d

QUERY

What are the typical products of PHB hydrolysis under acidic versus alkaline conditions?

+

EXTRACTED KNOWLEDGE GRAPH

Each line is a tuple in the format:\n<tuple_number>. <subject> <relation> <object>
1. abiotic PHB hydrolysis produces (R)-3-HBA and CA
2. abiotic PHB hydrolysis temperature_range 70C
3. acidic solutions enhance hydrolysis
.
25. alkaline hydrolysis yields mixture of CA (26%) and 3-HBA (48%)
.
33. strongly acidic and basic media promote hydrolysis

LLM'S RESPONSE

Under acidic conditions, the hydrolysis of PHB is enhanced [23], and generally, both acidic and basic media promote hydrolysis [33]. However, the specific products of hydrolysis under acidic conditions are not explicitly detailed in the provided triples.

Under alkaline conditions, PHB hydrolysis yields a mixture of crotonic acid (CA) and (R)-3-hydroxybutyric acid (3-HBA), with CA making up 26% and 3-HBA making up 48% of the products [25]. Additionally, abiotic PHB hydrolysis produces (R)-3-HBA and CA [21].

Thus, the typical products of PHB hydrolysis under alkaline conditions are CA and 3-HBA, but the specific products under acidic conditions are not detailed in the provided information.