

Eliot: Interactively Exploring Fast-Changing Scientific Literature Trends with Online Data and Learning

Bernardo A. Denkvitts, Nitin Gupta, and Biplav Srivastava

Department of Computer Science and Engineering

University of South Carolina

Columbia, SC, USA

{denkvitb, niting}@email.sc.edu, biplav.s@sc.edu

Abstract—The rapid growth of scientific publishing has made it increasingly difficult to track how fast-moving areas evolve. Search engines and LLM-based assistants retrieve or summarize papers, but often hide how the corpus was selected, organized, or connected to temporal patterns. We present *Eliot*, a publicly deployed interactive system for traceable exploration of evolving scientific literature. *Eliot* generalizes literature-evolution analysis beyond hand-built taxonomies and domain-specific scripts. Given explicit query terms and filters, it retrieves arXiv papers at query time, represents each paper by title and abstract, clusters the corpus into themes, assigns representative keywords, and visualizes each cluster’s publication-year distribution. We evaluate *Eliot* as both an applied system and an interactive research aid. An offline configuration study across eight arXiv domains compares document representations, dimensionality reduction methods, and clustering algorithms using intrinsic clustering and topic-coherence metrics; the results support MiniLM embeddings with 10-dimensional UMAP and Agglomerative Clustering as a practical default. A scenario-based survey and expert focus group assess interpretability and use contexts: participants rated cluster labels as meaningful in 85% of scenario responses, and feedback indicated that *Eliot* is most valuable for auditable overviews of rapidly changing technical areas. These results suggest that query-time clustering and temporal inspection can complement search and generation tools by helping researchers inspect and refine the evidence behind literature trends.

Index Terms—scientific literature analysis, information retrieval, document clustering, temporal analysis

I. INTRODUCTION

Scientific publishing is growing quickly, making it difficult to track fast-changing areas ([1], [2]). Researchers must find relevant papers, understand how they separate into themes, identify emerging directions, and verify which documents support an apparent temporal pattern ([3], [4]). Recent literature assistants make search more interactive ([5], [6]), but ranked lists and generated syntheses can hide retrieval, selection, and organization decisions. Literature evolution analysis therefore needs an inspectable workflow in which users can see the corpus, themes, labels, temporal patterns, and source papers.

We present *Eliot*, an interactive system for exploring

evolving scientific literature from online data.¹ Given keywords, category filters, date ranges, and sorting criteria, *Eliot* retrieves papers from a data source at query time, clusters them, labels clusters with representative keywords, and visualizes temporal distributions. The public Streamlit prototype² (Figure 1) demonstrates this pipeline and emphasizes traceability: users can revise the query, inspect cluster memberships, compare labels, and open source records rather than relying only on an opaque summary.

Eliot is motivated by two ICAPS studies on LLMs in Automated Planning and Scheduling (APS), or *LLM-planning*, which focused on literature evolution. [7] manually reviewed 126 papers into eight categories, including plan generation, language translation, model construction, tool integration, and interactive planning. [3] then added a semi-automated arXiv pipeline for fetching, deduplication, classification, and trend analysis. It reported category drift when comparing the original corpus to an updated one: six categories declined in relative share of papers, two increased, and two new categories emerged. These studies showed that category-level literature evolution is useful, but the workflow required a fixed taxonomy, domain-specific scripts, supervised labels, expert review, and post-hoc temporal analysis. *Eliot* generalizes this workflow from one expert-maintained domain to arbitrary topics. Rather than classifying papers into prebuilt categories, it performs *unsupervised, query-time grouping* over the retrieved corpus, *labels* the discovered clusters, and exposes the source evidence behind each trend.

Our evaluation is organized around three empirical questions: **(RQ1)** *Which automated configuration should Eliot use by default?* We ask which combination of document representation, dimensionality reduction, and clustering provides robust structure across diverse arXiv domains. **(RQ2)** *Do users find the generated clusters interpretable and useful?* Through survey-based and focus-group-based user studies,

¹Here, *online data* means that the corpus is retrieved at query time from an evolving data source rather than drawn from a fixed benchmark dataset, and *trends* refers to inspectable temporal patterns in that retrieved data rather than a fully automated trend discovery.

²<https://ai4society-paper-searcher.streamlit.app>

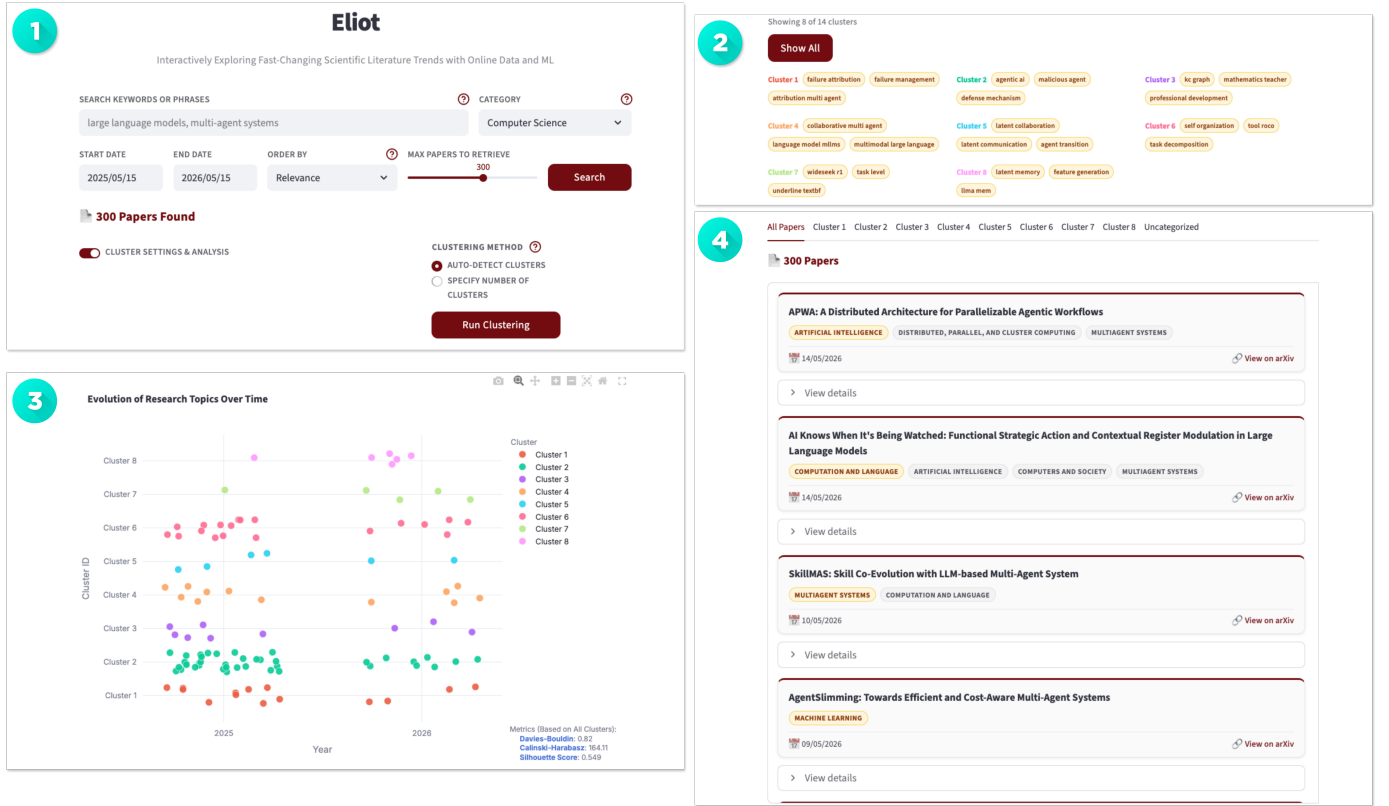


Fig. 1. The *Eliot* user interface, shown across four screenshots of a single scrollable application. (1) The search panel with default run configurations. (2) Auto-detected cluster overview, showing representative keywords for 8 of 14 discovered clusters (the full set is accessible via ‘Show All’). (3) Temporal interactive scatter plot showing the evolution of research topics over time across clusters. (4) Paginated paper list with per-cluster tabs. Additional papers are not shown due to space.

we ask whether users perceive the cluster labels, groupings, and interaction flow as meaningful for literature exploration. **(RQ3)** *When do users expect Eliot to be most useful, and what limitations matter?* We ask which research contexts benefit from traceable cluster-and-time views, and where users need additional guidance when interpreting changing corpora. Thus, our contributions are:

- A public interactive system for literature evolution exploration that combines query-time dataset retrieval, clustering, keyword labeling, temporal visualization, and document-level inspection.
- A traceable workflow that exposes query settings, retrieved papers, cluster assignments, and source links, supporting auditability in exploratory literature analysis.
- An automated offline configuration evaluation across eight arXiv domains, complemented by a scenario-based user survey and expert focus group.

II. RELATED WORK

This section situates *Eliot* within three lines of work: scientific text clustering, AI-assisted search and synthesis, and literature discovery systems.

A. Scientific Text Clustering

The rapid growth of scientific publications has motivated unsupervised methods for organizing academic texts. [1] compare TF, TF-IDF, and binary weighting with K-Means [8], finding that TF-IDF [9] performs best under their keyword-overlap connection coefficient and that introductions can yield stronger clusters than abstracts. *Eliot* uses titles and abstracts because they are concise and computationally lighter for interaction.

Prior work has also compared clustering algorithms and semantic representations for scientific documents. [10] evaluate K-Means, Hierarchical Agglomerative Clustering, and DBSCAN using intrinsic metrics including Silhouette Score, Calinski-Harabasz Index, and Davies-Bouldin Index. Transformer-based approaches similarly explore richer document representations: [11] report improved clustering with BERT-based representations over Word2Vec, while [12] combine GPT-derived embeddings with HDBSCAN for scientific abstracts.

These studies motivate the representation and clustering components in *Eliot*, but our setting adds two requirements. First, intrinsic clustering metrics alone do not indicate whether the resulting cluster labels are interpretable to users. Second, a public exploration system should use configurations that

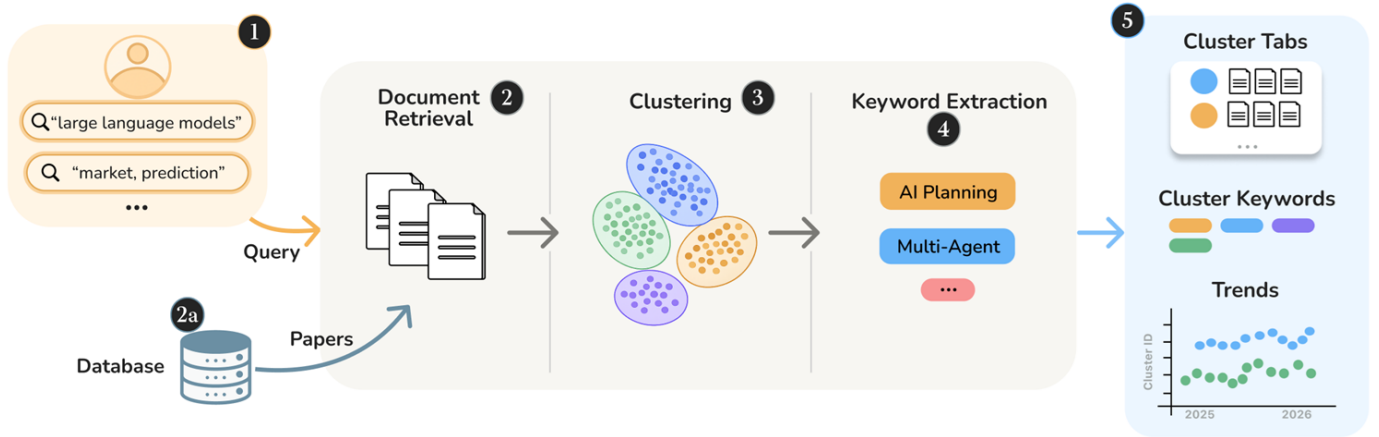


Fig. 2. System pipeline overview. (1) The tool supports multiple keywords, both single-word and multi-word expressions. (2, 2a) After receiving user keywords, the system retrieves papers from a database. (3) It performs clustering and (4) keyword extraction. (5) The UI displays cluster keywords, cluster tabs, and a temporal visualization to support structured exploration.

remain effective across heterogeneous scientific domains rather than a single fixed corpus. Therefore, we evaluate document representations, dimensionality-reduction methods, and clustering algorithms across eight scientific domains using both clustering and topic-coherence metrics, complemented by user evaluation.

B. AI-Assisted Literature Search and Synthesis

Recent AI researcher assistants combine scholarly retrieval with language-model based analysis. *Scholar Labs* decomposes research questions and provides per-paper explanations [13], while *Undermind* combines semantic search, citation trails, and adaptive exploration [14]. Other systems focus on downstream evidence workflows, such as *Elicit* supports systematic-review tasks [15], *Consensus* combines literature retrieval with AI-generated summaries [16], and *Scite* exposes citation context and citation intent [17].

Retrieval-augmented research systems extend this workflow toward answer generation. *PaperQA* and *PaperQA2* retrieve and evaluate full-text evidence to generate cited answers [18], [19], while *OpenScholar* retrieves passages from a large scientific corpus to produce citation-backed responses [20]. These systems are important comparators because they address the same high-level problem: researchers need help searching, reading, and synthesizing rapidly growing literature.

Our goal is narrower and more inspectable. *Eliot* does not attempt to produce a final evidence synthesis or determine which claims are correct. Instead, it supports an earlier exploratory stage by exposing the retrieved corpus as themes, representative labels, temporal patterns, and paper-level evidence. In this way, *Eliot* complements researchers and AI assistants focusing on query expansion, summarization, or claim verification.

C. Literature Discovery and Exploration

Literature discovery systems also organize scientific corpora to support exploratory search. [5] propose a conversational

search system built on an ACL Anthology knowledge graph with precomputed clusters and keyword labels. Because its analysis is computed offline over a fixed corpus, it cannot dynamically adapt to emerging topics. [21] present *Knowledge Navigator*, which organizes documents from broad scientific queries into a two-level hierarchy of named topics and subtopics. This is close to our emphasis on structured overviews, but *Eliot* instead provides a deployed, query-time workflow linking clusters to keywords, temporal distributions, and inspectable paper records.

[6] introduce *IntellectSeeker*, a personalized literature management platform that combines probabilistic recommendation with LLM-based semantic enhancement and dynamic web acquisition. However, beyond summaries and word clouds, it provides limited structure for analyzing retrieved documents. Citation-network tools such as *Litmaps* [22] and *ResearchRabbit* [23] help users expand from seed papers and visualize connections. *Eliot* instead starts from explicit query terms and arXiv filters, then constructs semantic clusters and temporal views over the retrieved set.

Prior work on research trend analysis often studies topic evolution and emerging directions using static datasets and offline analysis [3], [24], [25]. These systems demonstrate the value of temporal views, but they usually require precomputed corpora or expert-created taxonomies. *Eliot* complements this line of work combining dynamic retrieval, on-the-fly clustering, keyword labeling, temporal visualization, and deployment into a traceable workflow for researchers working with changing corpora.

III. SYSTEM OVERVIEW

Eliot is designed for exploratory analysis of scientific literature. It combines query-time document retrieval, machine-learning-based clustering, keyword labeling, temporal visualization, and document inspection so that users can move from a broad query to an evidence-backed view of themes in the retrieved corpus. An overview of the pipeline is shown

in Figure 2. This section outlines overall system design and implementation details for `Eliot` demo are provided in Section IV.

`Eliot` is guided by **four design goals**. First, it should support *query-time exploration* so users can study fast-changing topics without waiting for a curated corpus or maintaining a domain-specific extraction script. Second, it should provide *structured overviews* by grouping retrieved papers into themes rather than showing only a ranked list. Third, it should support *temporal inspection* by showing how clustered papers are distributed across publication years, while leaving trend interpretation auditable by the user. Fourth, it should preserve *traceability*: every theme and temporal pattern should remain connected to the papers, abstracts, metadata, and source links that produced it.

The **system workflow** begins by retrieving a set of relevant documents from a chosen paper database (for example, arXiv, bioRxiv, PubMed, etc.) given a user query composed of keywords or comma-separated phrases, along with user filters such as paper category, date range, and sorting criterion. The query is intentionally explicit: each term is searched in titles or abstracts, category filters are visible, and the date range defines the temporal window used in downstream analysis. Keeping retrieval terms explicit supports traceability and makes the system complementary to automatic query expansion methods.

The retrieved documents are represented using their titles and abstracts, then grouped into clusters of semantically related papers. To enhance interpretability, each cluster is associated with representative keywords that summarize what distinguishes it from the rest of the retrieved corpus. The keyword layer is important because users first encounter clusters through short labels, then decide which groups warrant deeper inspection.

Users can influence the clustering process through two modes: *automatic clustering*, where the system estimates clusters from the structure of the retrieved articles, and *user-controlled clustering*, where users specify the desired number of clusters; latter supporting sensitivity analysis when the automatic grouping is too coarse or too fragmented.

The interface presents the result count, cluster labels, temporal scatterplot, quality metrics, and cluster-specific tabs. Within each tab, users can inspect titles, metadata, abstracts, and links to the original database records. This design keeps the system grounded in documents rather than generated summaries, which is essential for trust in exploratory literature analysis.

IV. IMPLEMENTATION DETAILS

This section describes the deployed prototype and the offline evaluation datasets. The user interface is shown in Figure 1. As mentioned in Section III, a central implementation goal is *traceability*: the interface exposes query settings, retrieved papers, cluster assignments, representative keywords, metrics, and temporal visualizations rather than only returning a generated narrative. The machine-learning layer is unsupervised and is used to structure query results through text representa-

tion, dimensionality reduction, clustering, and cluster-keyword extraction.

For the implementation, ArXiv was chosen as the data source due to its open access nature, broad coverage of scientific topics, and up-to-date content. The system constructs queries, maps arXiv subject categories, and retrieves papers using Python packages [26], [27]. User keywords are interpreted as comma-separated terms; each term is searched in the title or abstract, and all terms are combined conjunctively. Multi-word expressions are preserved as phrases. Category and date filters are also applied. The API returns the latest version of each paper. The deployed search client further allows the option to choose the maximum number of results (20–500) from arXiv, while the offline evaluation datasets in Table II are capped at 300 papers per domain to keep configuration sweeps comparable and computationally manageable.

For text processing, only the title and abstract of each paper are used, as they provide a concise and readily accessible representation of the paper while remaining computationally efficient to process. Then, for text representation, Sentence-Transformers’ all-miniLM-L6-v2 [28] is applied to the pre-processed texts before clustering.

Clustering in the deployed prototype supports two interaction modes. In *automatic mode*, HDBSCAN is used to infer clusters from the document representation and to separate low-confidence points as an uncategorized group. In *user-controlled mode*, Agglomerative Clustering assigns papers to a specified number of clusters, allowing users to inspect alternative granularities. Agglomerative Clustering was part of the offline evaluation (Section V), where K-Means and Fuzzy C-Means were also evaluated to identify the ideal configuration for *user-controlled mode*. HDBSCAN was not included in this evaluation because *automatic mode* is intended to operate without requiring users to specify the number of clusters, and HDBSCAN has been used in previous work for theme discovery [12], [29].

Cluster keywords are extracted using Class-Based TF-IDF (c-TF-IDF), inspired by BERTopic [30]. For each cluster, c-TF-IDF treats documents within a cluster as a single concatenated document and ranks terms by their importance relative to the other clusters, each likewise also represented as a single document [30]. The experimental evaluation also considers c-TF-IDF cluster keyword extraction for topic coherence scoring.

The interface visualizes each clustered paper by year and cluster identifier, shows intrinsic clustering metrics when available, and provides cluster tabs that expose titles, authors, categories, abstracts, and arXiv links. The choice of text processing and clustering methods was guided by the need for a responsive user experience in the deployed prototype, while the offline evaluation explores a broader range of configurations to identify optimal settings for different scientific domains and query types, as detailed in Section V.

The **public interface and release** is implemented in Streamlit [31] which allows the retrieval and clustering pipeline to be deployed as an interactive Python web appli-

TABLE I
PIPELINE EXECUTION TIME (SECONDS) ACROSS DIFFERENT CORPUS SIZES

Keywords	100 Docs	300 Docs	500 Docs
Explainable AI	11.32	32.49	58.84
Large Language Models	10.26	32.95	54.14
Market, Prediction	11.30	33.05	61.78

cation. The released prototype supports keyword or phrase search, arXiv category selection, date filtering, sorting by relevance or submission date, automatic clustering, user-selected cluster counts, paper-level inspection, and source links. To support reproducibility and transparency, the source code for the deployed prototype and evaluation scripts has been made publicly available³. This repository includes the Streamlit application, data processing scripts, and configuration files for the offline evaluation. The code is organized to allow others to replicate the deployment and evaluation processes, ensuring that the results can be independently verified and extended in future work.

Runtime analysis. We additionally examined the runtime behavior of the deployed pipeline as the number of retrieved papers increased. We measured pipeline execution time for three queries spanning the Computer Science and Quantitative Finance categories, using corpus sizes of 100, 300, and 500 papers.

As shown in Table I, pipeline execution time increased consistently with corpus size, from approximately 10-11 seconds for 100 papers to 54-62 seconds for 500 papers. In our implementation, the 500-paper setting also requires arXiv pagination, which introduces an enforced delay of about 3 seconds. These measurements reflect the current deployment environment and are therefore affected by both computational resources and network latency; deployment on better hardware settings would reduce execution time.

V. AUTOMATED OFFLINE EXPERIMENT

This section addresses **RQ1** through an automated offline configuration evaluation. The goal is to identify a robust default for interactive literature exploration; it is distinct from the user-facing evaluation in Section VI, which studies perceived interpretability, usefulness, and use contexts.

A. Evaluation Setup

Because users may search across very different domains, we evaluate configurations on eight arXiv datasets spanning various fields (Table II). Each dataset is collected using a domain-specific query and arXiv category filter, ordered by relevance applied through the same retrieval layer used by the prototype. We cap each dataset at 300 papers so that enough results are available for insightful analysis.

TABLE II
SUMMARY OF ARXIV DATASETS USED FOR OFFLINE CONFIGURATION EVALUATION. ELECTRICAL ENG. IS SHORTHAND FOR ELECTRICAL ENGINEERING AND SYSTEMS SCIENCE, AND QUANT. IS SHORTHAND FOR QUANTITATIVE.

Dataset Area	Topic	#Docs
Computer Science	Trustworthy AI	300
Physics	Condensed Matter (Superconductivity)	253
Electrical Eng.	Medical Image Processing	300
Mathematics	Fluid Dynamics	300
Statistics	Statistical Machine Learning	300
Quant. Biology	Neural Dynamics	169
Quant. Finance	Portfolio Optimization	252
Economics	Econometric Modeling	194

The **configurations** for the experiment are organized across three different axes. We evaluate three document representation strategies: MiniLM (all-MiniLM-L6-v2) [28], SciBERT [32], and TF-IDF [9]. The TF-IDF vectorizer uses minimum and maximum document-frequency thresholds of 3 and 0.7, at most 3000 features, and 1-3 word n-grams. We combine these representations with two dimensionality reduction methods, UMAP [33] and SVD [34], and three clustering algorithms with controllable K : K-Means [8], Agglomerative Clustering [35], and Fuzzy C-Means [36]. For dimensionality reduction, we test $n_{\text{components}} \in \{5, 10, 15\}$. For clustering, we test $K \in \{3, 4, \dots, 12, 15\}$; the value 15 is included as a higher-granularity stress test without exhaustively searching every larger value.

Evaluation **metrics** are further divided into three intrinsic clustering metrics and two topic coherence metrics. Silhouette Score (SIL) [37] measures how close points are to their assigned cluster relative to other clusters, with higher values preferred. Calinski-Harabasz Index (CHI) [38] measures the ratio of between-cluster separation to within-cluster dispersion, with higher values preferred. Davies-Bouldin Index (DBI) [39] measures average cluster similarity based on dispersion and separation, with lower values preferred. Because these intrinsic metrics evaluate cluster geometry rather than label interpretability, we also compute two topic coherence metrics using OCTIS [40] over the c-TF-IDF keywords assigned to each cluster [41]. C_V estimates whether the top keywords in a cluster tend to co-occur in related contexts, while C_{NPMI} measures normalized pairwise co-occurrence among keywords; for both metrics, higher values indicate more coherent and potentially more interpretable cluster labels.

Given the large number of configurations, we use *rank aggregation* to select the best configuration for each dataset. For each metric, configurations are ranked independently, with DBI ranked in ascending order and all other metrics ranked in descending order. We sum the metric ranks to obtain an aggregate score, where a lower score indicates a stronger

³<https://github.com/ai4society/ELIOT>

configuration. We then collect the top configuration from each dataset and analyze the frequency of its components: representation, dimensionality reduction method, number of components, and clustering algorithm. We determine the final recommended configuration from the most frequent components, using average metric performance to resolve ties.

Note that this evaluation is designed to identify robust defaults, not to prove that one clustering method is universally optimal. The datasets are query-dependent, arXiv-centered, and limited to titles and abstracts. The coherence metrics evaluate keyword interpretability indirectly and should be complemented by user judgments. These limitations motivate the user study and focus group described in Section VI.

B. Evaluation Results

The experiment results answer **RQ1**. Table III summarizes the top-ranked configurations for each dataset, selected via rank aggregation over clustering and coherence metrics, explained in this section. The coherence scores should be interpreted comparatively across configurations rather than as absolute quality judgments, since they are computed from short title-abstract texts and automatically extracted cluster keywords.

Regarding **text representation**, MiniLM appears in the top-ranked configurations across all eight datasets, while neither SciBERT nor TF-IDF appears as the best-performing representation. This result is promising given the task. MiniLM is trained to produce sentence-level embeddings that capture semantic similarity, making it well-suited for clustering title-abstract representations. In contrast, SciBERT produces contextualized token-level representations and often benefits from task-specific pooling or fine-tuning for downstream use. TF-IDF is fast and transparent, but it relies on surface word overlap, which can miss semantically related papers [12].

About **dimensionality reduction and clustering**, UMAP is present in all top-ranked configurations, while SVD does not appear in any of them. This might suggest that nonlinear dimensionality reduction better preserves the structure useful for clustering in these datasets. Among UMAP configurations, $n_{\text{components}} = 10$ is the most frequent value. Agglomerative Clustering appears most frequently among the top configurations, followed closely by K-Means. The difference is modest, so the result should be interpreted as a default-selection signal rather than a decisive algorithmic result. For an applied interactive system however, this result still matters: a stable default reduces the need for users to understand clustering hyperparameters before beginning exploration.

Based on these findings, the default *Eliot* configuration uses MiniLM, 10-dimensional UMAP, and Agglomerative Clustering, since each component occurs most frequently among the top-ranked configurations across datasets; *automatic mode* uses the same setting, but with HDBSCAN as clustering algorithm. These findings answer **RQ1** by selecting a practical default from the tested configuration space, rather than claiming universal optimality.

The strongest applied takeaway is that *Eliot* is usable with its defaults selected from a benchmark-driven discovery process, even though it uses off-the-shelf methods. We do not assume that users will tune representation models, dimensionality reduction, and clustering algorithms. The system should therefore expose simple default interaction choices while internally relying on a configuration that generalizes across several domains. These automated results support the machine-learning component of *Eliot* as a practical structuring layer for retrieved literature, but they do not by themselves validate whether users find the clusters interpretable or useful; those questions are addressed through the user evaluation in the following section.

VI. USER EVALUATION

This section addresses **RQ2** and **RQ3** through two complementary methods: a scenario-based survey study and an expert focus group. Unlike the automated offline evaluation in Section V, this evaluation studies user-facing interpretability, perceived usefulness, and the contexts in which actual academic researchers expect *Eliot* to be most valuable.

We designed the **survey-based user study** to evaluate usefulness, interpretability, and interaction flow for **RQ2**. Participants were asked to evaluate *Eliot* using automatic clustering mode for two predefined scenarios. In **Scenario 1**, they reviewed a *Computer Science* search with the query *large language models* over the April 2024-2026 period. In **Scenario 2**, they reviewed a *Quantitative Finance* search with the query terms *market* and *prediction* over the same period. Each scenario presented retrieved papers, cluster keywords, and temporal views. Participants also used *Eliot* independently with their own queries in a self-directed task. Both the predefined scenarios and open-ended interaction were evaluated with 5-point Likert-scale questions and free-response prompts. The complete survey instrument, including the exact scenario configurations, representative cluster keywords, visualizations, survey questions, and free-response prompt, is available as supplementary material⁴.

A total of 10 respondents, consisting of graduate students and faculty members at a public university, participated in the survey. Across the two predefined scenarios, participants perceived the group labels as meaningful; 85% rated 4 or 5. A one-tailed Wilcoxon signed-rank test rejected the null hypothesis that users are neutral regarding label meaningfulness ($p < 0.001$). Since the respondents may not share their backgrounds with the example scenarios, the perceived usefulness of the clusters was moderate, with 50% of the responses rated 4 or 5. The difference from neutral was not statistically significant according to the same test ($p = 0.137$). To examine this pattern further, we convened a *focus group of experienced researchers* whose feedback we discuss next. In addition, participants showed a positive perception that *Eliot* could be useful for their future research ($p = 0.047$).

⁴<https://drive.google.com/file/d/1NA3mcGRJ4obaFIudOhanZGfbm1Y1vNnV>

TABLE III
TOP-RANKED CONFIGURATION OF EACH DATASET SELECTED VIA RANK AGGREGATION OVER CLUSTERING AND COHERENCE METRICS. EESS IS SHORTHAND FOR ELECTRICAL ENGINEERING AND SYSTEMS SCIENCE.

Dataset	Representation	Dim. Red.	n_{comp}	K	Algorithm	CHI	DBI	SIL	C_V	C_{NPMI}
Computer Science	MiniLM	UMAP	10	3	K-Means	310.29	1.02	0.58	0.47	-0.27
Physics	MiniLM	UMAP	15	10	K-Means	129.48	1.07	0.51	0.45	-0.26
Mathematics	MiniLM	UMAP	5	4	Fuzzy C-Means	406.23	0.91	0.57	0.73	-0.21
EESS	MiniLM	UMAP	15	9	Agglomerative	145.65	0.93	0.48	0.47	-0.21
Statistics	MiniLM	UMAP	10	6	Agglomerative	105.21	0.98	0.47	0.38	-0.28
Quantitative Biology	MiniLM	UMAP	10	10	Agglomerative	65.97	0.98	0.47	0.43	-0.21
Quantitative Finance	MiniLM	UMAP	10	4	Agglomerative	528.42	0.80	0.59	0.58	-0.18
Economics	MiniLM	UMAP	10	11	K-Means	123.94	0.86	0.59	0.38	-0.28

To study **RQ3**, we convened a **focus-group-based user evaluation** consisting of four experienced academic researchers at a public university. They all held PhDs, and together had research experience of over six decades in computer science and engineering (with two having experience of less than 10 years each and two more than 25 years each). In a recorded session, we reviewed their experience with the literature exploration problem in recent years, explained *Eliot*, demonstrated its capabilities, and asked for feedback.

Participants highlighted the system’s interactive cluster visualizations, cluster quality information and labels, and direct access to papers as useful features. They also suggested areas of improvement, including making the details and limitations of the data source explicit (e.g., arXiv’s API limits and category-filter recall), and providing information about query capabilities. In particular, one concern was that a query restricted to one arXiv category can miss relevant papers assigned to another category or distributed across multiple categories. This is a data-source constraint faced by arXiv-based tools generally, rather than a limitation specific to *Eliot*’s clustering pipeline, and thus, classified as a future improvement to the tool’s documentation and user interface rather than a change to the underlying system.

The two studies reveal important **findings**. With respect to **RQ2**, the survey results suggest that users find the generated labels meaningful and see promise for future research use, while perceived usefulness of predefined clusters is more dependent on the user’s background and the scenario topic. Regarding **RQ3**, the focus group suggests that *Eliot* is especially useful when researchers need a traceable overview of fast-changing technical areas. The focus-group findings are qualitative, and the survey evidence remains preliminary given the small sample size.

These findings connect back to the *LLM-planning* studies that motivated *Eliot* [3], [7]. Those studies required expert taxonomies and manual or semi-automated updates to make category shifts interpretable. User feedback suggests that *Eliot* addresses the same need: meaningful labels, cluster

inspection, quality information, and direct access to source papers help make evolving literature categories interpretable and auditable, even when users start from a query rather than a predefined taxonomy.

We also noticed that users were sometimes confused when the same query produced substantially different cluster granularities in *automatic mode* across different time windows, such as one year versus three years. This can occur because *Eliot* caps retrieved documents by default (300), and wider time windows with a fixed fetch limit may change the density structure in the embedding space used by HDBSCAN in *automatic mode*. To address this limitation, *Eliot* exposes two complementary controls. First, users can increase the maximum number of retrieved papers (in our observation, increasing this limit mitigated the issue). Second, users can switch to *user-controlled mode* and specify the desired number of clusters directly, with the system’s suggested value as an optional starting point.

VII. CONCLUSION

We presented *Eliot*, an interactive system for exploring evolving scientific literature through query-time retrieval, document clustering, keyword labeling, temporal visualization, and paper-level inspection. The system grew from a practical need identified in our prior LLM-planning literature studies: fast-moving research areas benefit from traceable views of changing corpora rather than workflows tied to fixed taxonomies or one-off analyses

Our evaluation separated automated configuration selection from user-facing assessment. For **RQ1**, MiniLM and UMAP appeared in every top-ranked configuration, and Agglomerative Clustering appeared most often, motivating MiniLM with 10-dimensional UMAP and Agglomerative Clustering as the default. For **RQ2**, the survey results indicate that users generally found the cluster labels meaningful and saw promise for future research use, while cluster usefulness depended more on scenario fit. For **RQ3**, the expert focus group suggested

that *Eliot* is most valuable as a traceable overview tool for fast-changing technical areas.

Several directions remain for future work. On the retrieval side, query expansion using lexical resources, domain ontologies, relevance feedback, or LLM-suggested terms could improve recall while preserving the explicit-query design that supports traceability. Expanding retrieval beyond arXiv to additional scholarly sources could also reduce dependence on arXiv-specific coverage and category boundaries. On the analysis side, the current evaluation is limited to titles and abstracts from eight arXiv domains. The deployed prototype is also limited by Streamlit's free hosting tier, including scale-to-zero after periods of inactivity. Custom hosting could provide more consistent availability and greater deployment flexibility. Finally, the survey and focus group reported here are preliminary, and a larger controlled user study would provide stronger evidence about how researchers use and interpret *Eliot* across different scientific domains.

GenAI Usage Disclosure. The authors used generative AI tools for limited language editing, organization, and phrasing assistance during manuscript preparation. Generative AI tools were also used to support prototype development and to generate the illustrative cluster image shown in step 3 of Figure 2. All AI-assisted text, image, and code were reviewed, edited, and verified by the authors. The authors take full responsibility for the paper's content, implementation, and conclusions. Generative AI was not used to collect data, run experiments, compute evaluation results, or draw scientific conclusions.

ACKNOWLEDGMENT

We thank Aarohi Goel and Thrinadh Sai for contributions to an earlier version of *Eliot*, and participants in our user studies.

REFERENCES

- [1] B. Probiez, J. Kozak, and A. Hrabia, "Clustering of scientific articles using natural language processing," *Procedia Computer Science*, vol. 207, pp. 3449–3458, 2022.
- [2] M. A. Hanson, P. G. Barreiro, P. Crosetto, and D. Brockington, "The strain on scientific publishing," *Quantitative Science Studies*, vol. 5, no. 4, pp. 823–843, 2024.
- [3] V. Pallagani, N. Gupta, B. C. Muppasani, and B. Srivastava, "Revisiting llms in planning from literature review: a semi-automated analysis approach and evolving categories representing shifting perspectives," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 35, no. 1, 2025, pp. 386–390.
- [4] C. Zhang, Z. Li, and J. Zhang, "A survey on visualization for scientific literature topics," *Journal of Visualization*, vol. 21, no. 2, pp. 321–335, April 2018.
- [5] P. Schneider and F. Matthes, "Conversational exploratory search of scholarly publications using knowledge graphs," *arXiv preprint arXiv:2410.00427*, 2024.
- [6] W. Bian, S. Liu, Y. Zhou, D. Chen, Y. Liao, Z. Fan, and A. Wang, "Intellectseeker: A personalized literature management system with the probabilistic model and large language model," in *International Conference on Knowledge Science, Engineering and Management*. Springer, 2024, pp. 270–282.
- [7] V. Pallagani, B. C. Muppasani, K. Roy, F. Fabiano, A. Loreggia, K. Murugesan, B. Srivastava, F. Rossi, L. Horesh, and A. Sheth, "On the prospects of incorporating large language models (llms) in automated planning and scheduling (aps)," in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 34, 2024, pp. 432–444.
- [8] J. B. McQueen, "Some methods of classification and analysis of multivariate observations," in *Proc. of 5th Berkeley Symposium on Math. Stat. and Prob.*, 1967, pp. 281–297.
- [9] S. Qaiser and R. Ali, "Text Mining: Use of TF-IDF to Examine the Relevance of Words to Documents," *International Journal of Computer Applications*, vol. 181, no. 1, pp. 25–29, July 2018.
- [10] M. Cherradi and A. El Haddadi, "Exploration of Scientific Documents through Unsupervised Learning-Based Segmentation Techniques," *Seminars in Medical Writing and Education*, vol. 3, p. 68, April 2024.
- [11] S. F. Taşkıran and E. Kaya, "Academic Text Clustering Using Natural Language Processing," *Konya Journal of Engineering Sciences*, vol. 10, pp. 41–51, Dec. 2022.
- [12] M.-H. Weng, S. Wu, and M. Dyer, "Identification and Visualization of Key Topics in Scientific Publications with Transformer-Based Language Models and Document Clustering Methods," *Applied Sciences*, vol. 12, no. 21, p. 11220, Nov. 2022.
- [13] Google Scholar. Scholar Labs: An AI powered scholar search. [Online]. Available: <https://scholar.googleblog.com/2025/11/scholar-labs-ai-powered-scholar-search.html>
- [14] T. Hartke and J. Ramette. (2024) Benchmarking the undermind search assistant. [Online]. Available: https://www.undermind.ai/static/Undermind_whitepaper.pdf
- [15] Elicit. (2026) Systematic literature reviews. [Online]. Available: <https://elicit.com/solutions/literature-review>
- [16] Consensus. (2026) How Consensus works. [Online]. Available: <https://help.consensus.app/en/articles/9922673-how-consensus-works>
- [17] J. M. Nicholson, M. Mordaunt, P. Lopez, A. Uppala, D. Rosati, N. P. Rodrigues, P. Grabitz, and S. C. Rife, "scite: A smart citation index that displays the context of citations and classifies their intent using deep learning," *Quantitative science studies*, vol. 2, no. 3, pp. 882–898, 2021.
- [18] J. Lála, O. O'Donoghue, A. Shtedritski, S. Cox, S. G. Rodrigues, and A. D. White, "Paperqa: Retrieval-augmented generative agent for scientific research," *arXiv preprint arXiv:2312.07559*, 2023.
- [19] M. D. Skarlinski, S. Cox, J. M. Laurent, J. D. Braza, M. Hinks, M. J. Hammerling, M. Ponnappati, S. G. Rodrigues, and A. D. White, "Language agents achieve superhuman synthesis of scientific knowledge," *arXiv preprint arXiv:2409.13740*, 2024.
- [20] A. Asai, J. He, R. Shao, W. Shi, A. Singh, J. C. Chang, K. Lo, L. Soldaini, S. Feldman, M. D'arcy *et al.*, "Openscholar: Synthesizing scientific literature with retrieval-augmented llms," *arXiv preprint arXiv:2411.14199*, 2024.
- [21] U. Katz, M. Levy, and Y. Goldberg, "Knowledge navigator: Llm-guided browsing framework for exploratory search in scientific literature," in *Findings of the Association for Computational Linguistics: EMNLP 2024*, 2024, pp. 8838–8855.
- [22] Litmaps. (2026) Litmaps for researchers. [Online]. Available: <https://www.litmaps.com/about/for-researchers>
- [23] ResearchRabbit. (2026) ResearchRabbit: AI tool for smarter, faster literature reviews. [Online]. Available: <https://www.researchrabbit.ai/>
- [24] C. Chen, "CiteSpace II: Detecting and visualizing emerging trends and transient patterns in scientific literature," *Journal of the American Society for Information Science and Technology*, vol. 57, no. 3, pp. 359–377, Feb. 2006.
- [25] X. Wang, S. Zhang, and Y. Liu, "ITGInsight—discovering and visualizing research fronts in the scientific literature," *Scientometrics*, vol. 127, no. 11, pp. 6509–6531, Nov. 2022.
- [26] L. Schwab, "arxiv.py: Python wrapper for the arxiv api." [Online]. Available: <https://pypi.org/project/arxiv/>
- [27] Zubov, Roman, "arxivql: arxiv query language and tools." [Online]. Available: <https://pypi.org/project/arxivql/>
- [28] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks," in *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019, pp. 3980–3990.

- [29] M. A. Mersha, M. G. Yigezu, and J. Kalita, "Semantic-driven topic modeling using transformer-based embeddings and clustering algorithms," *arXiv preprint arXiv:2410.00134*, 2024.
- [30] M. Grootendorst, "BERTopic: Neural topic modeling with a class-based TF-IDF procedure," *arXiv preprint arXiv:2203.05794*, 2022.
- [31] S. Inc., "Streamlit: A faster way to build and share data apps," 2019, open-source Python library for building interactive web applications. [Online]. Available: <https://streamlit.io>
- [32] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," *arXiv preprint arXiv:1903.10676*, 2019.
- [33] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," *arXiv preprint arXiv:1802.03426*, 2018.
- [34] Z. Zhang, "The singular value decomposition, applications and beyond," *CoRR*, vol. abs/1510.08532, 2015. [Online]. Available: <http://arxiv.org/abs/1510.08532>
- [35] J. H. Ward Jr, "Hierarchical grouping to optimize an objective function," *Journal of the American statistical association*, vol. 58, no. 301, pp. 236–244, 1963.
- [36] J. C. Bezdek, R. Ehrlich, and W. Full, "Fcm: The fuzzy c-means clustering algorithm," *Computers & Geosciences*, vol. 10, no. 2, pp. 191–203, 1984. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0098300484900207>
- [37] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
- [38] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Communications in Statistics-theory and Methods*, vol. 3, no. 1, pp. 1–27, 1974.
- [39] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE transactions on pattern analysis and machine intelligence*, no. 2, pp. 224–227, 1979.
- [40] S. Terragni, E. Fersini, B. G. Galuzzi, P. Tropeano, and A. Candelieri, "OCTIS: Comparing and Optimizing Topic models is Simple!" in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*. Online: Association for Computational Linguistics, 2021, pp. 263–270.
- [41] M. Röder, A. Both, and A. Hinneburg, "Exploring the space of topic coherence measures," in *Proceedings of the eighth ACM international conference on Web search and data mining*, 2015, pp. 399–408.