ReSearchy: Guarding Novelty, Guiding Ideas

Shilong Li
sli148@illinois.edu
Univeristy of Illinois, Urbana-Champaign
Champaign, Illinois, USA

Xuanming Zhang
xz130@illinois.edu
Univeristy of Illinois, Urbana-Champaign
Champaign, Illinois, USA

Yixuan Li
yixuan19@illinois.edu
Univeristy of Illinois, Urbana-Champaign
Champaign, Illinois, USA

Ya-Ting Pai
yatingp2@illinois.edu
Univeristy of Illinois, Urbana-Champaign
Champaign, Illinois, USA

Abstract

Academic research is often hampered by the time-consuming nature of literature reviews and the risk of redundant work due to the difficulty of identifying relevant studies that use different terminology. This paper presents ReSearchy, a tool designed to address these challenges by providing an integrated solution for identifying potential overlaps between a researcher's work and existing literature. ReSearchy allows researchers to input their research idea or abstract and receive a list of semantically similar papers, with a unified interface to compare their input against selected papers and highlight overlapping content. This hybrid approach, combining semantic matching using language models with efficient vector search, enables early identification of potential redundancy and facilitates focused research. We detail the system architecture, implementation using Python and Javascript, and the evaluation methodology employing a synthetic dataset to assess the tool's ability to retrieve relevant abstracts. ReSearchy aims to streamline the research process, promoting efficiency and originality in academic contributions. The code for ReSearchy is available at: https://github.com/averypai/ReSearchy

Keywords

Information Retrieval, Semantic Matching, Research Tool, Overlap Detection, Content Comparison, Natural Language Processing

ACM Reference Format:
Shilong Li, Xuanming Zhang, Yixuan Li, and Ya-Ting Pai. 2025. ReSearchy: Guarding Novelty, Guiding Ideas. In *Proceedings of CS510 Final Project Report*. ACM, New York, NY, USA, 5 pages.

1 Introduction

The pursuit of novel research is a cornerstone of academic and scientific advancement. However, a significant challenge faced by researchers, including both seasoned professors and students, is the unintentional pursuit of ideas that have already been explored or published. Discovering substantial overlap with existing work late in the research process can lead to wasted time, resources, and effort, ultimately hindering the pace of innovation. Traditional literature review methods, while essential, can be inefficient and may fail to uncover conceptually similar research, especially when there are differences in terminology or expression. This inefficiency not only leads to duplicated efforts but also diverts valuable academic resources from addressing genuine knowledge gaps.

To address this critical pain point, we introduce ReSearchy, a platform designed to empower researchers by "Guarding Novelty and Guiding Ideas." ReSearchy aims to provide an early warning system for potential research overlap by leveraging advanced semantic matching techniques. Users can input their research ideas, abstracts, or preliminary findings, and ReSearchy will search and compare this input against a comprehensive database of existing academic literature, focusing initially on open-access sources like arXiv.

The core contribution of ReSearchy lies in its integrated approach to idea overlap detection and content comparison. When a researcher submits their concept, the system not only retrieves semantically similar papers but also provides a highlighted comparison indicating the nature of the overlap. This immediate and nuanced feedback allows researchers to quickly assess the novelty of their ideas, identify areas for refinement, or pivot their focus towards more innovative contributions early in the ideation process. By facilitating a more efficient and informed initial research phase, ReSearchy strives to help researchers focus their efforts on truly novel investigations, thereby accelerating the discovery of new knowledge and making the academic process more effective.

This paper will detail the motivation behind ReSearchy, its system architecture, the methodologies employed for development and evaluation, and a discussion of its potential impact on the research community. We will also outline future directions for the platform.

2 Related Work

The challenge of efficiently navigating and synthesizing existing research has led to the development of various tools and approaches. Traditional literature review methods often prove inefficient, and important works can be missed due to variations in terminology.

Our proposed system offers a streamlined workflow where a user inputs their specific research concept and receives not only a list of semantically similar existing papers but also an integrated content comparison with highlighted sections of similarity. This direct comparison feature, tailored for evaluating the novelty of an initial idea against existing publications, distinguishes ReSearchy's

CS510 Final Project Report, Champaign, IL

approach from broader research discovery platforms, with the goal of helping researchers quickly gauge potential redundancy and refine their unique contributions before significant time investment.

2.1 Semantic Similarity and Information Retrieval

A core component of ReSearchy is the use of semantic similarity to identify related work. Research in semantic similarity focuses on measuring the degree to which two text fragments are related in meaning, going beyond simple keyword matching. Techniques like word embeddings and contextual embeddings, generated by models such as BERT [1], have significantly advanced this field. Information retrieval systems also aim to provide relevant documents to users, but traditional systems often rely on keyword-based queries. ReSearchy's hybrid approach, combining semantic search with efficient vector search using tools like Milvus [4], builds upon this research to improve the accuracy and efficiency of literature retrieval.

2.2 Literature Review Support Tools

Some tools aim to assist researchers in conducting literature reviews. These tools may offer features like:

- Citation analysis: Identifying influential papers based on citation networks.
- Keyword extraction: Automatically identifying important terms in a set of documents.
- Document summarization: Providing concise summaries of research papers.

While these tools offer valuable support, ReSearchy's focus on proactively identifying potential overlap in research ideas and providing detailed content comparisons distinguishes it.

2.3 Plagiarism Detection Software

- Turnitin [3]: This is one of the most widely used plagiarism detection tools in education. It compares submitted documents against a vast database of online content, previously submitted student papers, and academic publications. It provides a similarity score and highlights matching text.
- Grammarly [2]: While Grammarly is primarily known as a writing assistance tool, its plagiarism checker compares text to a large database to identify potential instances of plagiarism.

Although the goal of ReSearchy is not plagiarism detection, there are some overlaps in the techniques used for content comparison. Plagiarism detection software analyzes documents to identify sections of text that are similar to other sources. ReSearchy utilizes similar text comparison techniques to highlight overlapping content between a user's input and retrieved papers. However, ReSearchy's purpose differs significantly from tools like Turnitin. While Turnitin's core function is to verify originality by detecting textual matches for the purpose of identifying potential plagiarism, ReSearchy is designed to aid researchers in the early stages of the research process. ReSearchy focuses on identifying potential overlaps between a researcher's ideas or abstract and existing work, and highlighting textual similarities to guide research and prevent redundant work. ReSearchy achieves this by using semantic matching to find conceptual similarities, going beyond Turnitin's primary focus on textual matches. ReSearchy is a tool for discovery and guidance, not for policing academic misconduct.

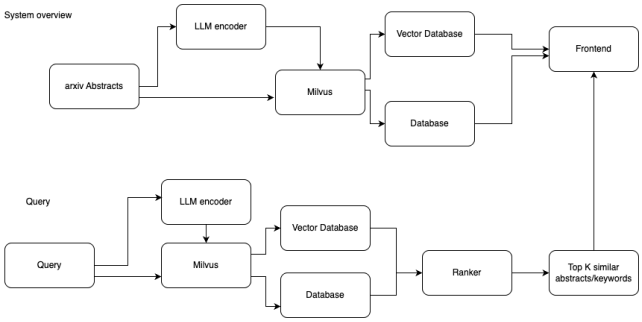

Figure 1: Architecture

3 Methodology

This section outlines the systematic approach for ReSearchy design and development, focusing on identifying conceptual overlaps between user research ideas and existing academic literature. We detail the system architecture, datasets, and implementation choices.

3.1 Overview

As shown in Figure 1, our system is designed to retrieve top-K relevant arXiv abstracts given a user query. It leverages Milvus and encoders to represent both documents and queries in the same embedding space. The overall architecture consists of two main workflows: indexing and querying.

Algorithm 1 Indexing Pipeline

Require: arXiv abstracts $\mathcal{A} = \{a_1, a_2, \ldots, a_n\}$
  for each $a_i \in \mathcal{A}$ **do**
    $v_i \leftarrow \text{LLMEncode}(a_i)$
    $\textsc{Milvus.Insert}(v_i)$
  end for

Algorithm 2 Query Pipeline

Require: Query q
  $v_q \leftarrow \text{LLMEncode}(q)$
  $C \leftarrow \textsc{Milvus.Search}(v_q)$
  $R \leftarrow \textsc{Ranker}(v_q, M)$
  return Top-K ranked abstracts or keywords

In the indexing workflow, arXiv abstracts are encoded into both dense and sparse vector representations. The dense embeddings capture semantic similarity using an encoder, while the sparse vectors retain exact lexical information. Both representations are stored in the same collection within Milvus, which supports hybrid search across multiple vector fields.

In the querying workflow, the input query is encoded into dense vector using the same encoder setup. Both the input and its dense

vector are used in a hybrid search query within Milvus, which combines the similarity scores of both representations to retrieve the most relevant document candidates. A weight ranker module then reorder the results. Finally, top-K most relevant abstracts are presented through the frontend interface.

3.2 Dataset

We use arXiv abstracts, each paired with its metadata (title, authors, categories, etc.). The data is preprocessed by formatting it consistently. These are then passed to the encoder for vectorization.

3.3 Implementation

Our implementation includes an encoder for vector generation, and a ranking module to refine retrieved results.

3.3.1 Document Encoding. Each arXiv abstract is passed through an encoder to obtain two types of vector representations:

- Dense embeddings generated using an encoder to capture semantic similarity.
- Sparse embeddings derived using a term-weighted encoder that preserves lexical specificity.

These embeddings are inserted into the same collections within Milvus, which we configure to support hybrid vector indexing. Milvus is used as the core database due to its efficient support for high-dimensional ANN search across both dense and sparse vectors.

3.3.2 Query Handling. At inference time, a user query undergoes the same encoding process to produce both dense and sparse representations. These are passed into Milvus to retrieve top-N candidate documents using hybrid search strategy.

3.3.3 Ranking and Fusion. Retrieved candidates are re-ranked with a weight ranker. The top-K most relevant abstracts or keywords are returned to the frontend.

3.3.4 Frontend Integration. The final results are served to the frontend application, which supports keyword highlighting. This enables a smooth exploration of semantically or lexically similar scientific content.

4 Evaluation

4.1 Evaluation Design

To evaluate the effectiveness of the ReSearchy hybrid retrieval system, we constructed a controlled benchmark dataset. The evaluation pool consists of 1,000 unrelated research papers, mainly sampled from the fields of non-computational linguistics (non-CL), such as systems, security and linguistics, to ensure minimal overlap with the topics of the target query.

We selected 20 real summaries from the CL domain as evaluation queries. For each query, five semantically similar variants are generated using the GPT-based rewriting technique, marked from level 1 to level 5, and the semantic similarity gradually decreases. However, due to overgeneralization and semantic drift in the level 5 variant, only the first four levels (level 1 to level 4) are retained as positive base truth values. This results in a total of 80 relevant documents in all queries. During the evaluation period, each query was matched with a pool of 1,004 documents (1 original + 4 positive variants + 1,000 distractors). The original summary used for querying is not included in the pool.

For each query, we use the BGE-M3 encoder to generate dense and sparse vector representations. Milvus' hybrid vector search function is used for storing and searching these embeddings. The retrieval is performed using weightedRanker with the same weight (with a dense score of 0.5 and a sparse score of 0.5). The top-K retrieval size is 4 to evaluate the system, reflecting the basic true positive number of each query.

4.2 Evaluation Metrics

We report standard retrieval metrics to assess accuracy and ranking quality.

- **Precision@4**: Proportion of retrieved documents that are relevant among the top 4 results.
- **Recall@4**: Proportion of relevant documents that appear in the top 4 results.
- **NDCG@4**: Normalized Discounted Cumulative Gain, which evaluates both relevance and ranking order.
- **MRR**: Mean Reciprocal rank of the first relevant result.

These metrics are computed independently for each query and reported individually in the 16 evaluation cases.

4.3 Results

The results are showed in **Table 1**. The hybrid retrieval system achieved perfect scores in **Precision@4**, **Recall@4**, and **MRR** in all 20 queries, indicating that all relevant documents were successfully retrieved and at least one was always ranked first. Minor differences in **NDCG@4** reflect slight reordering among the retrieved ground-truth variants.

Table 1: Retrieval Metrics for First 10 Queries and Overall Average (K=4)

Query ID	Precision@4	Recall@4	NDCG@4	MRR
2301.00320v1	1.0000	1.0000	1.0000	1.0000
2301.00321v1	1.0000	1.0000	0.9627	1.0000
2301.00397v1	1.0000	1.0000	0.9362	1.0000
2301.00399v1	1.0000	1.0000	0.9362	1.0000
2301.00418v1	1.0000	1.0000	0.9362	1.0000
2301.00422v1	1.0000	1.0000	0.9362	1.0000
2301.00429v1	1.0000	1.0000	0.9627	1.0000
2301.00539v1	1.0000	1.0000	0.9362	1.0000
2301.00604v1	1.0000	1.0000	0.9362	1.0000
2301.00628v2	1.0000	1.0000	0.9362	1.0000
		...		
Average	**1.0000**	**1.0000**	**0.9589**	**1.0000**

These results demonstrate that ReSearchy's dense-sparse hybrid retrieval approach is highly effective for identifying semantically similar content, even in the presence of substantial distractor noise. The tool consistently identifies all relevant variants, supporting its role as a reliable assistant in early-stage literature review and idea refinement.

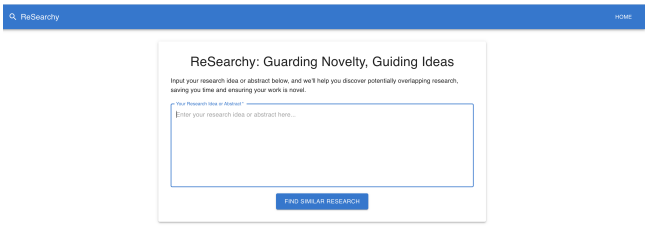

Figure 2: Interface for inputting research ideas or abstracts

5 Instructions to Use

ReSearchy provides a concise workflow to help researchers identify possible overlaps between their research ideas and existing literature. This section briefly describes how to use the system and its main features.

5.1 System Access

ReSearchy is an open source tool. The code and deployment instructions are available in the GitHub repository: https://github.com/averypai/ReSearchy. The repository provides a detailed guide for local deployment, including Milvus backend configuration and frontend installation instructions.

5.2 User Interface and Workflow

The user's interaction process with ReSearchy includes the following three steps:

5.2.1 Input Research Idea. On the homepage (see figure 2), users can enter their research ideas or abstracts into the designated text area. The interface is designed to be simple so that users can focus on the research content itself.

5.2.2 Review Search Results. After submission, the system will display a list of documents that are semantically similar to the input content and sort them by relevance (see Figure 3). Each result includes the following information:

- Concept similarity percentage
- Paper title and author
- Publication year (if available)
- Abstract snippet

For each result, the user can choose from two available actions:

- "Compare" – opens a side-by-side view to closely examine the similarities between the user input and the selected paper
- "View Paper" – links directly to the full publication for further reading and context

Users can sort results by relevance or publication year using the drop-down menu in the upper right corner.

5.2.3 Analyze Content Comparison. Clicking the "Compare" button for any search result will take you to the comparison view (see figure 4), which provides the following features:

- Display user input and corresponding paper abstract side by side
- Highlight the overlap in concepts or terms in orange

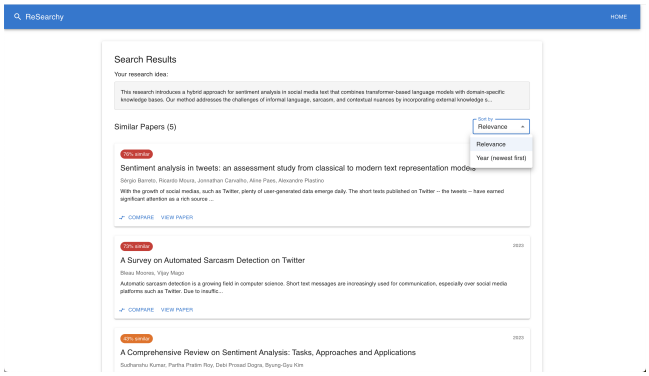

Figure 3: Search results page sorted by semantic similarity

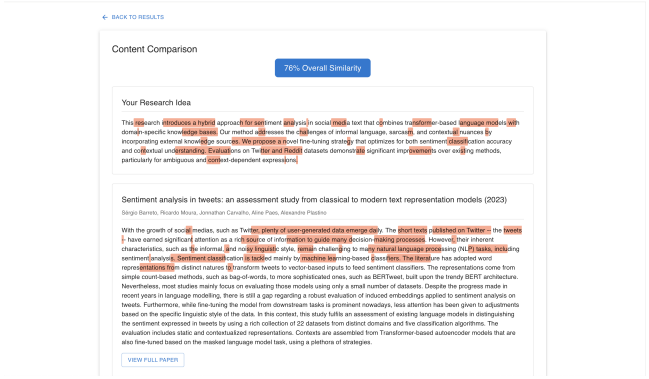

Figure 4: Compare views to highlight semantically overlapping content

- Display the overall approximate size percentage at the top of the page

This comparative view allows users to:

- Identify specific areas of conceptual overlap
- Click "View Full Paper" to access the full publication in a new tab
- Return to the results list via the "Back to Results" link

This integrative approach enables researchers to quickly assess conceptual similarity and seamlessly transition to in-depth reading when appropriate.

5.3 Usage Notes

When using ReSearchy, it is recommended to pay attention to the following points:

- **Input Quality**: The system works best with well-structured, clearly described studies. Complete abstracts or detailed study structures generally result in more accurate matches.
- **Highlight content interpretation**: The current highlighting mechanism can accurately mark the semantic overlap, and its matching accuracy will be further optimized in the future. Users should regard the highlighted areas as potential overlaps rather than complete duplications.

- **Database Scope**: The current version only indexes some arXiv abstracts, and will be gradually expanded to more academic sources and journals in the future.
- **Research Context**: ReSearchy is intended as a research aid to help identify potential overlaps, not to make final judgments on research novelty. System results should be used as a starting point for further literature searches.

By using the above process, researchers can discover the connection between their ideas and existing research earlier, thereby more effectively clarifying research directions and avoiding duplication of investment.

6 Conclusion

In conclusion, this project has introduced ReSearchy, a platform designed to address the ubiquitous and often costly challenge of unintentional idea overlap in academic research. Recognizing that significant investments of time and resources can be nullified by the late discovery of pre-existing similar work, ReSearchy offers a proactive solution. By leveraging advanced semantic matching techniques, powered by pre-trained language models, and initially drawing from open-access databases like arXiv, our platform provides researchers with an early and insightful mechanism for detecting conceptual similarities between their nascent ideas and the vast body of published literature.

A key innovation of ReSearchy lies in its seamlessly integrated system for idea overlap detection and content comparison. This cohesive function empowers users to input their research concepts or abstracts and, in turn, receive an immediate and intuitive analysis. This includes not only the identification of potentially overlapping papers but also a direct, highlighted comparison detailing the specific areas of similarity. This granular level of feedback is designed to transform traditional literature review workflows from a reactive search process into a proactive discovery and refinement tool, making the path to novel research more transparent and efficient.

The development of ReSearchy as a web tool, with planned future extensions such as plugin integration for popular writing platforms, is strategically aimed at enhancing research productivity. By enabling researchers to swiftly identify and navigate away from inadvertent duplication, ReSearchy ensures that intellectual efforts are more effectively concentrated on exploring genuine knowledge gaps. This targeted focus promises not only to conserve valuable time and institutional resources, but also to accelerate the overall advancement of scholarly knowledge. ReSearchy thus serves as a practical guide, steering the research community towards more impactful and truly original contributions. The project development and codebase can be followed at https://github.com/averypai/ReSearchy.

7 Contributions

Shilong Li was responsible for the frontend development of the system. He designed and implemented the user interface, enabling users to input queries and view retrieved results effectively. His work included integrating the frontend with backend APIs, building components for displaying top-K search results, and ensuring a smooth and responsive user experience.

Xuanming Zhang focused on backend development. He implemented the core retrieval pipeline, connecting the encoder and Milvus vector database. He also developed the RESTful API endpoints that allowed seamless communication between the frontend and backend, ensuring that queries were properly encoded, executed, and returned to the user interface.

Yixuan Li configured Milvus for hybrid search. He defined the schema for storing both dense and sparse embeddings in the same collection, managed the insertion of encoded vectors, and configured search parameters to support efficient hybrid retrieval. His work enabled the system to leverage both semantic and lexical information during search.

Ya-Ting Pai was responsible for data preparation and overall system workflow design. She preprocessed the arXiv abstract dataset, ensuring it was ready for embedding and storage. She also designed the end-to-end workflow, including the indexing and querying pipelines, and coordinated the integration of various system components to ensure consistency and functionality throughout the project.

References

[1] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805 [cs.CL] https://arxiv.org/abs/1810.04805

[2] Grammarly. [n. d.]. Grammarly website. https://www.grammarly.com/ Accessed: 2025-05-07.

[3] Turnitin. [n. d.]. Turnitin website. https://www.turnitin.com/ Accessed: 2025-05-07.

[4] Zilliz. 2019–. Milvus: An open-source vector database. (GitHub repository). https://github.com/milvus-io/milvus Accessed: 2024-05-07.