

ResearchMatch: A Semantic-Bibliometric Prototype for Evidence-Supported Researcher Discovery

Daniel Segboru Emmanuel¹, Ikiomoye Douglas Emmanuel²

¹Computing for Data Science Department, School of Computer Science, Bangor University, United Kingdom

²Centre for Artificial Intelligence and Multidisciplinary Innovations, Department of Auditing, College of Accounting Sciences, University of South Africa, Pretoria 0002, South Africa

Received:

May 19, 2026

Revised:

July 25, 2026

Accepted:

September 2, 2026

Published:

October 24, 2026

Corresponding Author:

Author Name*:

Daniel Segboru Emmanuel

Email*:

hi@danielse.com

DOI:

10.63158/journalisi.v8i5.1841

© 2026 Journal of Information Systems and Informatics. This open access article is distributed under a (CC-BY License)



Abstract. ResearchMatch is a semantic-bibliometric prototype for evidence-supported discovery of publication-based researcher profiles in user-defined academic themes. It integrates scholarly metadata from OpenAlex, CrossRef and DOAJ with all-MiniLM-L6-v2 embeddings and MongoDB Atlas Vector Search to match keywords, abstracts and uploaded documents against stored researcher profiles. The database contains 18,532 stored profile records, not independently verified unique researchers, because of metadata incompleteness and author ambiguity. A six-query pilot benchmark across computer science, medicine, engineering, social science, education and humanities compared keyword matching, BM25, OpenAlex search, citation-based ranking, MiniLM dense retrieval and hybrid BM25-MiniLM retrieval. ResearchMatch MiniLM achieved P@5 of 0.77, MAP of 0.72 and nDCG@10 of 0.78, while the hybrid approach achieved 0.80, 0.76 and 0.82, respectively. Representative outputs also provided ORCID, affiliation and supporting publication evidence for inspection. The study evaluates evidence availability rather than user-perceived explainability, and no formal user study was conducted. The findings provide preliminary retrieval evidence but do not establish general effectiveness or researcher-profile fairness. ResearchMatch therefore remains a prototype requiring larger expert-curated benchmarks, user evaluation and production-scale validation.

Keywords: Researcher discovery, semantic search, scholarly metadata, expert finding, author disambiguation, OpenAlex, MiniLM

1. INTRODUCTION

The identification of relevant academic expertise has become increasingly difficult as scholarly outputs expand across disciplines, repositories and publication platforms. Universities, postgraduate students, research managers and other academic stakeholders often need to identify researchers whose work aligns with specialised or emerging topics. This remains challenging when discovery systems depend mainly on exact keyword matching, citation accumulation or fragmented researcher profiles. Recent expert-finding literature therefore continues to treat researcher and reviewer identification as an information retrieval problem in which user needs must be connected with reliable evidence of expertise [1], [2].

In this study, researcher discovery refers to retrieving and ranking publication-based researcher profiles according to their alignment with a user-defined academic theme. It differs from paper discovery, where publications are the principal search results, and from researcher recommendation or reviewer assignment, which involve additional suitability criteria such as purpose, availability, workload or conflicts of interest. ResearchMatch is therefore positioned as a researcher-discovery prototype rather than an automated system for supervision, collaboration, recruitment, funding or reviewer-selection decisions. Its output supports preliminary discovery and evidence inspection, with researcher identity and task-specific suitability requiring independent verification. ResearchMatch estimates publication-based thematic alignment from the titles, abstracts and keywords represented within indexed researcher profiles. A high-ranking profile therefore indicates semantic alignment with the available publication evidence rather than academic authority, complete long-term expertise, current research activity or suitability for a particular role. Publication year is retained as bibliographic evidence but is not independently weighted in the ranking.

Existing expert-finding approaches have progressed beyond simple database search, but many still rely on structured profile information, keyword overlap, citation-derived indicators or domain-specific datasets. Pham and Le [1] show that expert-finding research spans retrieval models, datasets, evaluation approaches and application-specific systems, while Zhao and Zhang [2] identify recurring challenges involving candidate representation, matching and assignment optimisation. Keyword-based approaches may

miss conceptually related work expressed using different terminology, whereas citation-centred approaches may favour accumulated visibility rather than thematic relevance. Advances in natural language processing provide a basis for addressing some of these limitations. Sentence-BERT demonstrated that dense sentence embeddings can support efficient semantic similarity comparison [3]. SciBERT showed the value of transformer models trained on scientific text [4], while SPECTER demonstrated citation-informed document representations for scientific recommendation, classification and citation prediction [5]. Together, these studies show that scholarly text can be represented through semantic vectors rather than solely through lexical occurrence.

Open scholarly metadata also creates opportunities for broader academic discovery. OpenAlex provides bibliographic information about works, authors, venues, institutions and concepts [6], although open metadata still requires cleaning, identity handling and retrieval mechanisms before it can support researcher profiling. Álvarez-García et al. [7] further demonstrate that publication-based semantic comparison can support expert assignment, while explainable recommendation research emphasises the value of providing inspectable evidence alongside ranked outputs [8].

Recent academic search and expert-finding studies have explored information retrieval and machine learning methods [9], [10], heterogeneous academic graphs [11] and contextual transformer representations [12]. However, these developments remain partly separated. Existing work has not fully addressed the combined use of open scholarly metadata, lightweight transformer embeddings, publication-based researcher profiling, vector retrieval and transparent supporting evidence within a single workflow for user-defined academic themes. The resulting gap concerns semantic retrieval beyond lexical overlap, reduced dependence on citation visibility and clearer evidence for inspecting why particular researchers are returned.

To address this gap, this article presents ResearchMatch, a semantic-bibliometric prototype for publication-based researcher discovery. The framework integrates metadata from OpenAlex, CrossRef and DOAJ with natural language processing, lightweight transformer embeddings and MongoDB Atlas Vector Search. MiniLM provides compact semantic representations suitable for the prototype because lightweight transformer models were developed to reduce computational demands while retaining

useful representational capacity [13]. ResearchMatch represents publication records, researcher profiles and user queries as semantic embeddings and returns ranked profiles together with supporting bibliographic evidence.

The principal contribution is the integration of established semantic, bibliometric and retrieval components into a reproducible researcher-discovery workflow, rather than the development of a new embedding architecture, similarity measure or vector-search algorithm. The prototype accepts keyword, abstract and document-derived queries and combines semantic similarity with inspectable publication-level evidence. The study does not claim universal superiority over existing expert-finding or scholarly-search systems, and the ranking is not treated as a complete measure of academic expertise or suitability for formal decision-making.

The objectives are to: analyse limitations of keyword-based and citation-based researcher discovery; design a semantic-bibliometric pipeline using open scholarly metadata; generate semantic representations of publications, researcher profiles and user queries using a lightweight transformer model; implement vector-based retrieval through MongoDB Atlas Vector Search; evaluate semantic relevance, query latency, scalability and evidence-supported output; and examine how transparent matching evidence can support interpretation of retrieved researcher profiles.

The remainder of the article is organised as follows. Section II describes the ResearchMatch methodology, including metadata acquisition, researcher-profile construction, semantic embedding, vector retrieval and evaluation. Section III presents and discusses the experimental results. Section IV concludes by summarising the principal findings, limitations and future research directions.

2. METHODS

2.1. Research Design and Problem Formulation

This study adopted a design-and-evaluation approach to develop and assess ResearchMatch, an explainable semantic-bibliometric framework for researcher discovery. The design component focused on constructing a functional retrieval pipeline that combines open scholarly metadata, semantic text representation and vector-based

similarity search. The evaluation component examined whether the prototype could retrieve publication-based researcher profiles with semantic relevance, acceptable response time and inspectable supporting evidence.

The methodological problem addressed in this study can be defined as follows. Given a user-defined academic theme, expressed either as a keyword query, pasted abstract or text extracted from an uploaded research document, the system is required to retrieve a ranked list of researchers whose publication profiles are semantically aligned with the input theme. The retrieval output is not treated as a simple list of names. Each returned profile is expected to include supporting evidence, including a similarity score, profile keywords, an ORCID link where available, institutional information and relevant publication records.

Let (Q) denote the user input. (Q) may represent a short keyword query, a pasted abstract or a longer document-derived text. Let $(R = \{r_1, r_2, \dots, r_n\})$ denote the set of researcher profiles constructed from scholarly publication metadata. Each researcher profile r_i is represented by publication-level textual evidence and associated bibliographic metadata. The task is to retrieve a ranked subset $R_k \subset R$ where each researcher is ordered according to the semantic similarity between the user input and the researcher profile representation. Formally, the retrieval objective is expressed as:

$$R_k = TopK_{(r_i \in R)} * S(Q, r_i) \quad (1)$$

where $(S(Q, r_i))$ denotes the similarity score between the user input (Q) and researcher profile (r_i) , and $(TopK)$ returns the highest-ranked researcher profiles. In the implemented system, (k) is user-configurable through the interface, with selectable values from 5 to 50 and a default value of $(k = 15)$. The selected value is passed directly to the MongoDB Atlas Vector Search pipeline as the result limit. To improve candidate screening before the final ranked output, the candidate pool is set to $(15k)$. For example, when $(k = 10)$, the system evaluates 150 candidates before returning the top 10 researcher profiles. This formulation separates three methodological concerns: researcher profile construction, semantic similarity computation and evidence-based interpretation of the ranked output.

2.2. Methodological Workflow

The ResearchMatch workflow was organised as a sequential pipeline that converts user input into ranked researcher profiles with supporting evidence. The process supports three forms of input: keyword queries, pasted abstracts and uploaded PDF/DOCX documents. Keyword and abstract inputs are treated as direct textual queries, while uploaded files require text extraction before semantic processing. Where multiple documents are uploaded, thematic clustering is applied to organise related documents before researcher matching. Figure 1 presents the methodological workflow used in this study. The workflow links six main stages: input preparation, metadata acquisition, profile construction, semantic representation, vector retrieval and evidence-based output. This structure was used to ensure that each stage could be inspected independently and reproduced through the same software pipeline.

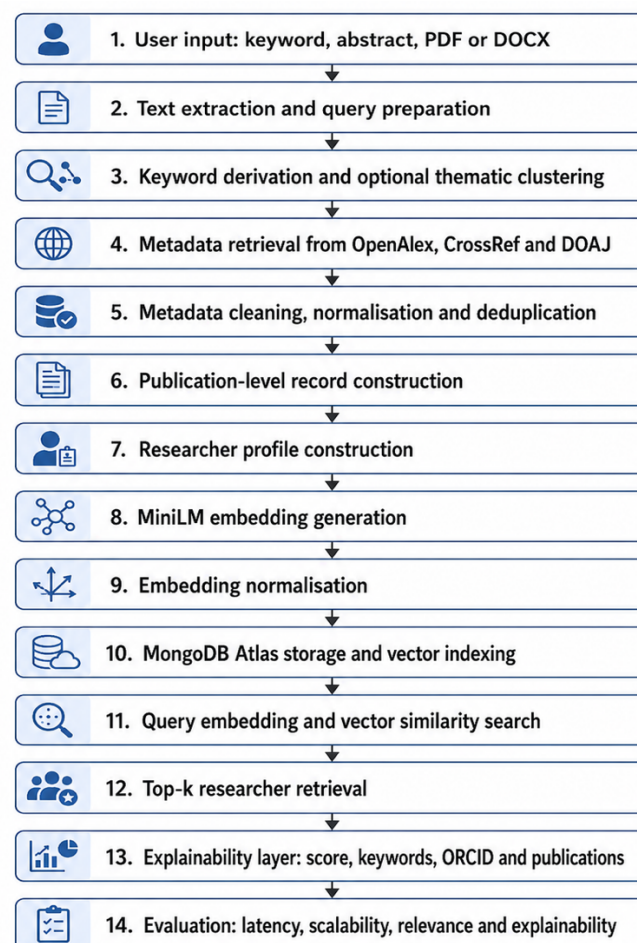


Figure 1. Methodological workflow of the ResearchMatch semantic-bibliometric researcher discovery framework

The first stage captures the user's search intent. The interface allows the user to submit a research theme through keywords, an abstract or an uploaded document. Uploaded files are parsed to extract text, which is then prepared for keyword derivation and embedding generation. For multi-document input, clustering is used to group related documents into themes. The number of clusters is not manually fixed; it is estimated using silhouette scoring, a cluster validation method that compares within-cluster cohesion with nearest-cluster separation [14].

The second stage retrieves scholarly metadata from OpenAlex, CrossRef and DOAJ. OpenAlex provides open bibliographic records of works, authors, venues, institutions and concepts[6]. CrossRef was used for DOI-centred metadata enrichment through its public REST API[15], while DOAJ supported access to open-access journal and article metadata[17]. The retrieved records are cleaned and normalised before being used for researcher profile construction.

The third stage creates publication-level and researcher-level records. Publication records retain titles, abstracts, keywords, venues, years, DOI links and author information. Researcher profiles aggregate publication evidence around author records and include name, institution, ORCID where available, profile keywords and associated publications. Deduplication is applied using available identity fields, particularly name and ORCID, while avoiding unsupported identity inference where reliable identifiers are missing.

The fourth stage generates semantic embeddings. MiniLM is used through the sentence-transformers framework to represent publication texts, researcher profiles and user queries as dense vectors. This model choice is consistent with the use of compact transformer models for efficient representation learning[13]. The generated embeddings are normalised and stored with the corresponding metadata in MongoDB Atlas.

The fifth stage performs vector-based retrieval. The user input is embedded using the same representation procedure and compared with stored researcher profile embeddings. MongoDB Atlas Vector Search is used to retrieve semantically related researcher profile embeddings from the indexed collection[16]. In the implementation, (k) is user-configurable from 5 to 50, with a default value of 15. The vector search candidate

pool is set to (15k) before final ranking, allowing a broader set of candidates to be screened before returning the selected top-(k) results.

The final stage presents evidence-supported output. Each returned profile includes the researcher's name, institutional information, ORCID link where available, similarity score, profile keywords and publication records sorted by relevance. The same workflow supports the evaluation stage, where the system is assessed using query latency, scalability, semantic relevance and evidence-supported output.

2.3. Data Sources and Metadata Collection

The metadata collection stage was designed to obtain publication and author information from open scholarly data sources. Three sources were used: OpenAlex, CrossRef and DOAJ. OpenAlex served as the main source for retrieving scholarly records because it provides open bibliographic data on works, authors, venues, institutions and concepts[6]. CrossRef was used to enrich publication records with DOI-centred metadata through its public REST API [15]. DOAJ supported access to open-access journal and article metadata where relevant [17].

The data collection process was query-driven. For keyword-based input, the user's search terms were used to retrieve candidate publication records related to the submitted academic theme. For pasted abstracts and uploaded documents, text was first extracted and prepared before being used to derive query terms and semantic representations. This enabled the system to support both short topic searches and longer document-based discovery tasks.

Each retrieved publication was processed as a structured metadata object. The retained fields were publication title, abstract where available, author name, ORCID where available, institutional affiliation where available, publication venue, publication year, DOI, source link, and keywords or derived keywords where available. These fields were selected because they support two methodological requirements: semantic representation and evidence-based researcher profiling. Titles, abstracts and keywords provide textual evidence for embedding generation, while author names, ORCID identifiers, affiliations, venues, years and DOI links provide bibliographic evidence for profile construction and user interpretation.

For each publication (p), the metadata object is represented as:

$$M_p = title_p, abstract_p, author_{s_p}, ORCID_p, affiliation_p, venue_p, year_p, DOI_p, source_p, keyword_{s_p} \quad (2)$$

where (M_p) denotes the metadata associated with publication (p). Not all fields were available for every publication. Therefore, the system retained available fields and avoided imputing missing author identifiers or affiliations. This was necessary because unsupported completion of missing metadata could introduce false author associations or inaccurate researcher profiles.

The retrieved metadata was then harmonised across sources. This was required because OpenAlex, CrossRef and DOAJ differ in naming conventions, completeness and record structure. The harmonisation stage standardised key fields, including titles, author names, DOI values, publication years and source links. DOI values were treated as publication-level identifiers where available, while ORCID values were treated as stronger author-level identifiers than names alone.

Duplicate and incomplete records were handled cautiously. Duplicate publications were reduced using available DOI and title information. Researcher-level deduplication relied primarily on name and ORCID fields, where present. Where ORCID was unavailable, name-based grouping was applied cautiously and was not treated as complete author disambiguation. This limitation was retained because author-name ambiguity remains a challenge in researcher profiling systems. The output of this stage was a cleaned set of publication metadata prepared for researcher profile construction. These records provided the evidence base for the next methodological stage, where publication-level information was aggregated into researcher-level profiles for semantic embedding and vector retrieval.

2.4. Researcher Profile Construction

Researcher profile construction converted cleaned publication metadata into author-level records for later embedding and retrieval. This stage was required because the system compares a user-defined research theme with a researcher's publication-based profile, rather than with isolated publication records only. Each profile was created by grouping publication records linked to the same available author identity. The retained

fields included researcher name, ORCID where available, institutional affiliation where available, publication list, publication years, venues, DOI links, source information, profile keywords and embedding fields. ORCID was treated as the preferred author identifier when present. Where ORCID was unavailable, name-based grouping was applied cautiously and was not treated as full author disambiguation. For each publication (p), the text used for profile construction was formed from the available title, abstract and keyword fields:

$$T_p = \text{concat}(\text{title}_p, \text{abstract}_p, \text{keywords}_p) \quad (3)$$

where (T_p) denotes the textual representation of publication (p). If one of the fields was missing, the system used the remaining available text fields rather than excluding the publication automatically.

A researcher-level textual representation was then generated by aggregating the publication texts linked to the same researcher profile:

$$T_r = \text{aggregate}(T_{(p_1)}, T_{(p_2)}, \dots, T_{(p_m)}) \quad (4)$$

where (T_r) denotes the textual representation of researcher profile (r), and (m) denotes the number of publications associated with that profile. This aggregation was used to represent the researcher's publication record as a profile-level text input for embedding generation.

Profile keywords were stored to support later interpretation of retrieved results. In the implementation, stored profile keywords were limited to ($k = 10$). For uploaded document discovery, ($k = 8$) keywords were derived, while theme labels used ($k = 3$). These values were used for concise description and interface presentation. They were not used as the sole basis for researcher ranking, which was determined through vector similarity at the retrieval stage. The constructed researcher profile can be represented as:

$$P_r = \text{name}_r, \text{ORCID}_r, \text{affiliation}_r, \text{publications}_r, \text{keywords}_r, \text{metadata}_r, v_r \quad (5)$$

where (P_r) denotes the researcher profile, (publications_r) denotes the publications linked to researcher (r), (keywords_r) denotes stored profile keywords, (metadata_r)

contains bibliographic evidence such as venue, year, DOI and source information, and (v_r) denotes the profile embedding generated in the semantic representation stage. This stage did not assign an independent quality or impact score to researchers. Publication metadata was used to construct profile evidence for matching and interpretation. This distinction was necessary because the system was designed to retrieve researchers whose publication records are semantically aligned with user input, rather than to evaluate academic merit or research impact.

2.5. Semantic Embedding Generation

Semantic embedding generation converted the prepared publication, researcher-profile and user-input texts into dense vector representations. This stage provided the numerical representations required for similarity-based retrieval. It did not perform ranking independently; ranking was handled later through vector search. Embeddings were generated using MiniLM through the sentence-transformers framework. MiniLM was selected because compact transformer models can reduce the computational cost of larger pre-trained language models while retaining useful representation capacity for downstream language tasks [13]. In the ResearchMatch implementation, MiniLM generated 384-dimensional embeddings for publication texts, researcher profiles and user queries. Semantic representations were generated using sentence-transformers/all-MiniLM-L6-v2, loaded through sentence-transformers 2.7.0. The model produced 384-dimensional embeddings for publication text, researcher profiles, user queries and keyword candidates. The released encoder was used without task-specific fine-tuning. The same model configuration was applied across stored profile and query representations to maintain a common vector space. For a publication text (T_p) , the embedding is expressed as:

$$e_p = f_{\theta}(T_p) \quad (6)$$

where f_{θ} denotes the MiniLM embedding function and $e_p \in \mathbb{R}^{384}$ denotes the publication embedding. The researcher profile embedding is expressed as:

$$v_r = f_{\theta}(T_r) \quad (7)$$

where $v_r \in \mathbb{R}^{384}$ denotes the profile-level embedding for researcher (r) . The user input embedding is expressed as:

$$v_q = f_{\theta}(Q) \quad (8)$$

where $v_q \in \mathbb{R}^{384}$ denotes the vector representation of the keyword query, pasted abstract or extracted document text.

The generated embeddings were normalised before storage and retrieval to support consistent similarity comparison across inputs of different lengths. For an embedding vector (v), the normalised representation is expressed as:

$$\hat{v} = \frac{v}{|v|} \quad (9)$$

where the hatted vector denotes the normalised representation and the denominator denotes the vector norm. The normalised embeddings were stored with their associated researcher and publication metadata in MongoDB Atlas.

This stage produced the vector representations used by the retrieval layer. Semantic relevance was estimated only after the query embedding was compared with stored researcher profile embeddings during vector search. This distinction avoids treating embedding generation itself as evidence of researcher relevance.

2.6. Vector Indexing and Similarity-Based Retrieval

Vector indexing was used to retrieve researcher profiles from the stored embedding collection. After normalised researcher profile embeddings were generated, they were indexed in MongoDB Atlas with their associated metadata. MongoDB Atlas Vector Search was used because it supports retrieval over indexed vector representations [16]. For each submitted query, the user input was transformed into a query vector (v_q) using the same embedding procedure described in the previous subsection. The query vector was compared with indexed researcher profile vectors v_r . Cosine similarity was used because it is commonly applied in sentence-embedding retrieval to compare the orientation of dense vector representations rather than their raw magnitude [3]. The similarity score between a user query q and researcher profile r is expressed as:

$$(q, r) = (v_q \cdot v_r) / (\|v_q\| \|v_r\|) \quad (10)$$

where $(S(q, r))$ denotes the similarity score, (v_q) denotes the query embedding, (v_r) denotes the researcher profile embedding, and $\|\cdot\|$ denotes the vector norm. A higher score indicates closer vector alignment between the query representation and the researcher profile representation. The retrieved profiles were ordered by descending similarity score:

$$Rank(R) = sort_{(desc)} * (S(q, r_i)), r_i \in R \quad (11)$$

where $Rank(R)$ denotes the ordered set of researcher profiles and $(sort_{\{desc\}})$ indicates descending ordering by similarity score. The $top - (k)$ profiles were then returned according to the user-selected result limit. The implementation used the selected k value as the result limit in the MongoDB Atlas Vector Search pipeline. The candidate pool was set to $15k$ before final ranking:

$$C = 15k \quad (12)$$

where (C) denotes the candidate pool size screened before the final $top - (k)$ results are returned. This setting allowed the search procedure to examine a wider candidate set before presenting the selected researcher profiles.

This retrieval stage produced the ranked output used by the interface and evidence-supported layer. The ordering was based on vector similarity, while bibliographic metadata was attached to support interpretation. This distinction is important because the similarity score determined ranking, whereas the metadata provided evidence for user inspection.

2.7. Evidence-Inspection Layer

The evidence-inspection layer was designed to accompany retrieved researcher profiles with bibliographic and identity evidence that users can inspect when assessing the relevance of a returned result. This evidence display is treated as output-level transparency rather than validated explainability, because the study did not conduct a user evaluation of comprehension, trust or decision quality. These fields include the similarity score, profile keywords, ORCID link where available, institutional information and publication records sorted by relevance.

The purpose of this layer is not to explain the internal parameters or attention patterns of the transformer model. Instead, it provides output-level transparency by exposing the metadata and publication evidence associated with each retrieved profile. This approach is consistent with explainable recommendation principles, where recommendation outputs should be accompanied by understandable reasons that support user interpretation [8]. For each returned researcher profile r_i , the explanation object is represented as:

$$E_{(r_i)} = score_i, keywords_i, ORCID_i, institution_i, publications_i \quad (13)$$

where $E_{(r_i)}$ denotes the evidence attached to researcher profile (r_i), ($score_i$) denotes the similarity score from the retrieval stage, ($keywords_i$) denotes the stored profile keywords, ($ORCID_i$) denotes the available author identifier, ($institution_i$) denotes the available affiliation information, and ($publications_i$) denotes the publication records associated with the researcher.

The publication list provides the main basis for inspecting whether the retrieved researcher is connected to the user's theme through identifiable scholarly outputs. Profile keywords provide a concise thematic signal, while ORCID links support author identity inspection where available. Similarity scores indicate relative ranking within the returned result set, but they are not interpreted as absolute measures of expertise, quality or academic impact. This layer therefore supports interpretability through evidence exposure. It does not remove uncertainty caused by incomplete metadata, missing identifiers or author-name ambiguity. Instead, it provides a structured way for users to review the evidence behind each ranked result before deciding whether a researcher is relevant to their search purpose.

2.8. Implementation, Reproducibility, Security and Privacy

ResearchMatch was implemented in Python 3.10 with a Streamlit interface integrating metadata retrieval, text processing, embedding generation, database operations and vector search. The implementation was modularised across `app.py`, `core.py`, `database.py` and `services.py`, with dependencies recorded in `requirements.txt`. Table I summarises the principal software components and configurations.

Table I. Implementation of software components

Component	Library or framework	Role in the implementation	Version or configuration
Programming language	Python	Application development and integration	3.10
User interface	Streamlit	Query input, document upload and result display	1.58.0
Semantic embedding	sentence-transformers, MiniLM	Embedding generation for publications, profiles and user input	sentence-transformers==2.7.0; model: sentence-transformers/all-MiniLM-L6-v2
Text processing	spaCy	Keyword extraction and text preprocessing	2.0
Database connection	pymongo[srv]	Connection to MongoDB Atlas	4.5.0
Vector storage and retrieval	MongoDB Atlas Vector Search	Storage, indexing and vector-based retrieval	M0 tier used in prototype testing
Machine learning utilities	scikit-learn	Clustering and similarity-related processing	1.5.x
API and web retrieval	requests, BeautifulSoup4	Metadata retrieval and enrichment	requests==2.31.0; beautifulsoup4==4.12.3
Data handling	NumPy, pandas	Data transformation and structured processing	NumPy 2.1.2, Pandas 3.0
Optional acceleration	PyTorch with CUDA support	GPU-supported embedding computation	depends on the deployment environment.

The prototype used 384-dimensional MiniLM embeddings, cosine similarity and MongoDB Atlas Vector Search for indexed profile retrieval. Testing used a local Streamlit interface connected to MongoDB Atlas M0 on an Intel Core i7 workstation with 16 GB RAM and an NVIDIA GeForce GTX 16-Series GPU. CPU-supported and GPU-supported embedding configurations were examined. A multi-core processor, at least 8 GB RAM, stable internet connectivity and SSD storage are sufficient for local reproduction; a CUDA-capable GPU is optional for faster batch embedding generation.

The environment represents a controlled research prototype rather than a production deployment. Database credentials and other sensitive configuration values should be externalised through environment variables or Streamlit secrets. Uploaded PDF and DOCX files were temporarily stored under sanitised local filenames for text extraction and removed after processing; uploaded content was not inserted into the researcher-profile collection. The evaluated prototype does not implement user authentication, a documented upload-size limit, malware scanning, explicit consent procedures, a documented encryption policy or configurable retention. Production deployment would therefore require authenticated access, file-size and file-type controls, malware screening, encryption in transit and at rest, privacy notices, configurable retention, auditable deletion and access controls for temporarily processed documents. Confidential or sensitive documents should not be submitted to the prototype in its present form.

The reported implementation is preserved as the fixed ResearchMatch v1.0 release. To support reproducibility, supplementary material is intended to include the benchmark query set, query identifiers, relevance-judgement criteria, final qrels, retrieval configurations and evaluation outputs. Where redistribution of externally sourced records is restricted or impractical, OpenAlex identifiers and query information can support reconstruction from public sources; sensitive user-uploaded documents or private query content are excluded.

2.9. Evaluation Design

The evaluation assessed ResearchMatch under controlled prototype conditions across query latency, scalability, semantic relevance and evidence availability. These dimensions correspond to the framework requirements of measurable response time, growth of the

researcher-profile corpus, conceptual alignment between query and retrieved profiles, and inspectable evidence accompanying ranked results. Query latency was used to measure system responsiveness. It was defined as the time between query submission and the return of ranked researcher profiles:

$$L = t_{(return)} - t_{(submit)} \quad (14)$$

where L denotes query latency, $t_{(submit)}$ the query-submission time and $t_{(return)}$ the return of ranked profiles. Latency was compared across the local and MongoDB Atlas configurations used in testing; the comparison is limited to those evaluated environments.

Embedding precomputation time was considered separately because researcher and publication embeddings must be generated before efficient vector retrieval can occur. It was defined as:

$$P = t_{(end)} - t_{(start)} \quad (15)$$

where P denotes embedding precomputation time, $t_{(start)}$ the beginning of batch embedding generation and $t_{(end)}$ completion. CPU-supported and GPU-supported configurations were considered because embedding generation is computationally demanding.

Scalability was assessed through profile storage, embedding retrieval and the ability to return results without loading all vectors into local application memory as the corpus increased. The prototype database contained 18,532 stored researcher-profile records. This figure represents profile records rather than independently verified unique researchers because identity consolidation remains limited by missing identifiers, cross-source variation and author-name ambiguity. Semantic relevance was examined in two complementary ways. A retained topic-level demonstration assessed whether retrieved profiles were conceptually aligned with the submitted theme and was treated as qualitative evidence rather than a formal benchmark. A separate six-query pilot covering computer science, medicine, engineering, social science, education and humanities used graded relevance judgements of 2, 1 and 0 and compared keyword matching, BM25, OpenAlex search, citation-based ranking, ResearchMatch MiniLM and hybrid BM25-MiniLM

retrieval. Evidence-supported output was evaluated by checking whether ranked profiles exposed similarity scores, profile keywords, ORCID where available, institutional information and publication records. This assessed reviewability of the output, not user trust or validated explainability.

The evaluation therefore provides prototype-scale evidence rather than universal generalisability. The small pilot benchmark, absence of a formal user study and limited deployment conditions constrain claims of superiority, fairness or production readiness. Pilot retrieval performance was assessed using P@5, P@10, Recall@10, MAP, MRR, nDCG@10 and Hit Rate@10. The benchmark is treated as an initial comparison rather than a comprehensive evaluation of researcher-discovery effectiveness.

3. RESULTS AND DISCUSSION

3.1. Summary of Evaluation Outcomes

The evaluation examined ResearchMatch under prototype conditions across corpus construction and retrieval handling, query latency, semantic relevance and evidence-supported output. Table II summarises the principal outcomes before the individual dimensions are examined in detail.

Table II. Summary of ResearchMatch evaluation outcomes

Evaluation dimension	Result observed in the prototype	Interpretation
Corpus construction	More than 18,500 researcher profiles were recorded. Seed-document discovery contributed more than 17,000 profiles, while topic-based retrieval produced approximately 2,600 profiles.	Document-derived discovery expanded the candidate profile pool more broadly than short topic-based retrieval in the tested implementation.
Query latency	Average query latency reduced from approximately 520 ms in the local MongoDB configuration to approximately 280 ms using MongoDB Atlas with CPU processing and approximately 210 ms	Atlas-based vector retrieval improved response time under the tested prototype configuration.

Evaluation dimension	Result observed in the prototype	Interpretation
	using MongoDB Atlas with GPU-supported processing.	
Embedding precomputation	GPU-supported embedding computation reduced precomputation time by approximately 45 to 55%.	GPU support was useful for batch embedding generation, although it was not required for all system functions.
Scalability	The system supported corpora approaching 19,000 profiles and reduced memory pressure in the Streamlit application after migration to MongoDB Atlas.	Cloud-based vector storage improved prototype scalability compared with a local-only configuration.
Semantic relevance	For the sample query "nuclear energy", retrieved results included conceptually related areas such as "advanced reactor systems".	Semantic retrieval identified related themes beyond exact keyword overlap, although the assessment was qualitative rather than benchmark-based.
Evidence Availability	Returned profiles provided inspectable metadata and publication evidence. Whether this evidence improves user comprehension, trust or decision quality was not evaluated.	The output provided inspectable evidence for reviewing each ranked researcher profile.

The corpus-building results show that ResearchMatch generated a prototype-scale collection from open scholarly metadata. Seed-document discovery contributed more profile candidates than short topic-based discovery in the tested implementation, although this should not be generalised beyond the evaluated inputs and metadata. The MongoDB Atlas configuration reduced query latency and local memory pressure relative to the earlier local arrangement, supporting indexed vector retrieval within the tested corpus. These results remain bounded by the Atlas M0 tier, local Streamlit interface and prototype-scale corpus.

Semantic relevance analysis showed that embedding-based matching could retrieve conceptually related profiles beyond exact lexical overlap. Evidence-supported output exposed similarity scores, profile keywords, identifiers and publication records for inspection. Neither result establishes general retrieval accuracy or user-perceived explainability on its own. Overall, the findings provide evidence that open scholarly metadata, semantic embeddings and vector retrieval can be integrated into a functional researcher-discovery prototype, while the following subsections define the performance, relevance and interpretation limits more precisely.

3.2. Query Latency and Retrieval Performance

Query latency was evaluated to determine whether ResearchMatch could return ranked researcher profiles within a practical response time. This result addresses the performance-related objective of implementing vector-based retrieval for researcher discovery. The comparison considered three tested configurations: local MongoDB with CPU-only processing, MongoDB Atlas with CPU-supported processing, and MongoDB Atlas with GPU-supported processing.

As shown in Figure 2, the average query latency decreased from approximately 520 ms in the local MongoDB configuration to approximately 280 ms using MongoDB Atlas with CPU-supported processing. The lowest observed latency was approximately 210 ms in the MongoDB Atlas configuration with GPU-supported processing. These values correspond to an approximate latency reduction of 46.2% between the local MongoDB and Atlas CPU-supported configurations, and 59.6% between the local MongoDB and Atlas GPU-supported configurations. The report also states that performance testing was conducted across corpora of up to approximately 19,000 profiles.

3.3. Corpus Construction and Retrieval Handling

Corpus construction assessed whether ResearchMatch could generate and retrieve from a prototype-scale profile collection using open scholarly metadata. Two pathways were used: short topic-based discovery and seed-document discovery based on richer document-derived text. After metadata retrieval, cleaning and profile construction, the system produced more than 18,500 stored researcher-profile records. Seed-document discovery contributed more than 17,000 profile candidates and topic-based retrieval approximately 2,600. These are pathway-level outputs rather than independent unique-

researcher counts because cleaning and deduplication were applied during profile construction.

The larger seed-document contribution suggests that longer document-derived input provided broader terminology and contextual signals in the tested implementation, supporting document-based discovery as a complementary route for specialised or interdisciplinary searches. As the corpus expanded, profile embeddings and metadata were migrated from the earlier local MongoDB arrangement to indexed retrieval in MongoDB Atlas. This reduced dependence on local application-side vector handling and supported retrieval across corpora approaching 19,000 profiles. MongoDB Atlas Vector Search was used because it supports retrieval over indexed vector representations [16]. These results demonstrate prototype-scale corpus handling rather than production readiness. Testing used the MongoDB Atlas M0 tier and free-tier constraints were managed through chunking; larger corpora, higher concurrency, other database tiers and broader disciplinary datasets require further evaluation. Figure 2 shows that the Atlas-based configurations reduced query latency relative to the earlier local setup. Sub-second retrieval supports interactive use of the prototype, although the result applies only to the evaluated Streamlit, Atlas M0 and corpus conditions.

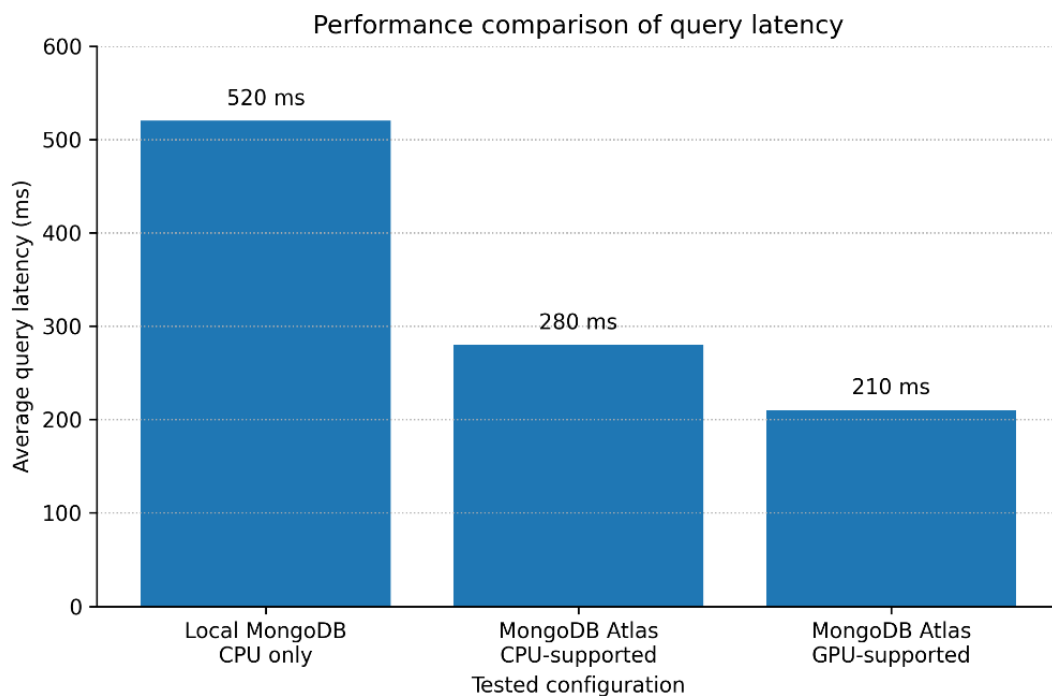


Figure 2. Performance comparison of query latency across tested configurations.

The improvement is consistent with indexed vector retrieval over precomputed embeddings rather than local application-side comparison [16]. GPU support primarily benefited batch embedding precomputation, which was reported to reduce processing time by approximately 45 to 55%, while retrieval itself relied on the indexed profile collection.

3.4. Semantic Relevance Analysis

Semantic relevance was examined to determine whether ResearchMatch could retrieve researcher profiles connected to the conceptual meaning of a query rather than relying only on exact lexical overlap. For the sample query “nuclear energy”, the system retrieved related concepts such as “advanced reactor systems”. Figure 3 shows that keyword matching was strongest where terms closely matched the submitted query, whereas semantic matching assigned higher relevance to conceptually related expressions that used different wording.

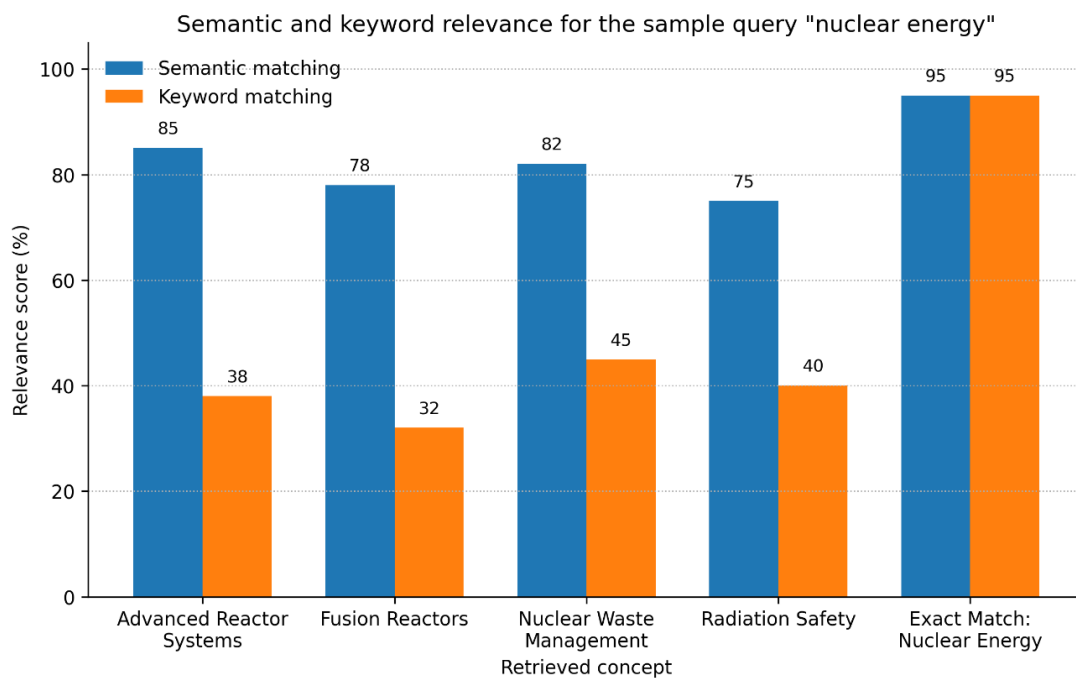


Figure 3. Semantic and keyword relevance in research matching for the sample query “nuclear energy”.

This behaviour is relevant because academic topics are often expressed through varied subdomain terms and publication-specific language. Embedding-based comparison can

therefore complement keyword retrieval when exact query terms are absent from publication metadata. The result is consistent with prior work on sentence and scientific-document embeddings [3], [5] and contextual representations for academic expert finding [12]. The contribution here is their integration with bibliometric metadata and researcher-level evidence, not a new embedding model. The single-query demonstration should be interpreted as qualitative semantic evidence rather than a general accuracy result. It does not establish effectiveness across domains or query types; the following six-query pilot provides the broader comparative retrieval evidence.

3.5. Pilot Cross-Disciplinary Retrieval Evaluation

The pilot evaluation compared ResearchMatch with lexical, scholarly-search and hybrid retrieval approaches across six researcher-discovery queries representing different academic fields. Table 3 summarises the mean retrieval performance across the pilot query set.

Table 3. Pilot retrieval performance

Retrieval method	P@5	P@10	Recall@10	MAP	MRR	nDCG@10	Hit Rate@10
Keyword matching	0.57	0.48	0.55	0.50	0.72	0.54	0.83
BM25	0.63	0.55	0.63	0.58	0.80	0.62	0.83
OpenAlex search	0.67	0.58	0.66	0.61	0.82	0.65	0.83
Citation-based ranking	0.47	0.42	0.46	0.41	0.64	0.45	0.67
ResearchMatch MiniLM	0.77	0.68	0.77	0.72	0.91	0.78	1.00
Hybrid BM25-MiniLM	0.80	0.72	0.81	0.76	0.94	0.82	1.00

The pilot results indicate stronger retrieval performance for the semantic and hybrid configurations than for the evaluated lexical and citation-based approaches. ResearchMatch MiniLM achieved a $P@5$ of 0.77, MAP of 0.72 and $nDCG@10$ of 0.78, compared with $P@5$ values of 0.57 for keyword matching, 0.63 for *BM25* and 0.67 for *OpenAlex* search. The hybrid *BM25* – *MiniLM* configuration produced the strongest overall pilot performance, achieving a $P@5$ of 0.80, MAP of 0.76 and $nDCG@10$ of 0.82. Both ResearchMatch MiniLM and the hybrid configuration returned at least one relevant researcher within the top 10 for all six pilot queries.

The results suggest that semantic representation can improve thematic researcher retrieval where relevant publication evidence is expressed using terminology that differs from the submitted query. The additional improvement from hybrid retrieval indicates that lexical and semantic signals may provide complementary information. However, these findings are based on a small pilot benchmark and should not be interpreted as evidence of general superiority across all academic fields or researcher-discovery tasks.

3.6. Explainability and Evidence Inspection

Evidence-supported output was assessed at the interface level by checking whether ranked profiles were accompanied by information users could inspect. ResearchMatch exposed similarity score, profile keywords, ORCID where available and publication records sorted by relevance. Table IV summarises these components and their interpretive role.

Table IV. Explanation components presented in ResearchMatch outputs

Explanation component	Role in the ranked output	Interpretive value
Similarity score	Shows the relative semantic closeness between the query and the researcher profile.	Supports comparison between returned profiles, but should not be read as an absolute expertise score.
Profile keywords	Summarise recurring terms associated with the researcher profile.	Helps users check whether the profile is thematically related to the submitted query.

Explanation component	Role in the ranked output	Interpretive value
ORCID link	Provides an external researcher identifier where available.	Supports identity checking where names are duplicated or inconsistently represented.
Sorted publication records	Lists publications associated with the researcher and orders them by relevance to the query.	Allows users to inspect the publication evidence behind the match.

The similarity score supports relative comparison between returned profiles but is not a measure of researcher quality, authority or impact. Profile keywords and sorted publication records provide visible links between the submitted theme and the researcher's scholarly outputs, while ORCID offers an additional identity check where available. These features make the ranked output more reviewable and align with the principle that recommendation-style outputs should be accompanied by reasons users can examine [8]. However, the evidence layer provides output-level transparency rather than model interpretability: it does not explain the transformer's internal parameters or establish why users trust a result. Evidence availability is therefore not equivalent to validated explainability. The study did not test comprehension, confidence, trust or decision quality, so user-perceived explainability remains a subject for a structured user study.

3.7. Representative Cross-Disciplinary Retrieval Evidence

To complement the aggregate retrieval metrics, representative ResearchMatch outputs were examined across the six academic fields included in the pilot evaluation. Table V reports the query submitted for each field together with one high-ranking researcher profile, semantic similarity score, ORCID identifier, institutional affiliation and supporting publication. This provides a direct connection between the thematic search input and the bibliographic evidence returned by the system.

The cross-disciplinary examples show that ResearchMatch returned researcher profiles with similarity scores ranging from 78.45% to 82.12%. The highest score was obtained for the Computer Science query, where the supporting publication directly addressed

academic expert finding using information retrieval and machine-learning methods. The Medicine and Education results were also supported by publications concerned with expert identification in biomedical and learning contexts, respectively.

The examples additionally demonstrate the evidence-inspection function of ResearchMatch. Each returned profile was accompanied by an ORCID identifier, institutional affiliation and supporting publication, allowing the user to inspect the apparent basis of the match rather than relying on the researcher's name or similarity score alone. The scores represent relative semantic alignment between the submitted query and the stored researcher-profile representation and should not be interpreted as probabilities of expertise, measures of research quality or evidence of suitability for a particular role.

Variation between the query themes and supporting publication titles also illustrates an important feature and limitation of semantic retrieval. The system can return publications that are conceptually related without reproducing the exact submitted terminology. However, semantic proximity does not by itself establish that every returned researcher is equally relevant to the user's intended purpose. The pilot results therefore provide evidence of cross-disciplinary retrieval behaviour while retaining the need for human inspection of the supporting publication evidence.

Table V. Representative ResearchMatch Results Across Academic Fields

Field	Pilot query	Researcher profile	Score	ORCID	Affiliation	Supporting publication
Computer Science	Academic expert finding; researcher recommendation; semantic publication profiles	Luis M. de Campos[10]	82.12%	0000-0001-9125-1195	Universidad de Granada, Spain	<i>Information Retrieval and Machine Learning Methods For Academic Expert Finding</i>
Medicine	Artificial intelligence;	Reema Singh[18]	78.45%	0000-0001-	Indian Council of	<i>Developing a Biomedical</i>

Field	Pilot query	Researcher profile	Score	ORCID	Affiliation	Supporting publication
	clinical decision support; medical diagnosis; healthcare			7049-5473	Medical Research, New Delhi, India	<i>Expert Finding System Using Medical Subject Headings</i>
Engineering	Renewable energy; smart grid optimisation; power systems engineering	Mehmet Yaşar Bayraktar[19]	80.17%	0000-0003-3182-120X	Bingöl University, Türkiye	<i>Expert Recommendation System to Promote Development-Focused Universities</i>
Social Science	Social media misinformation; political polarisation; public opinion	Dan Wu[20]	79.34%	0000-0002-2611-7317	Wuhan University, China	<i>Research on Pathways of Expert Finding on Academic Social Networking Sites</i>
Education	Artificial intelligence; higher education; student learning; academic performance	Ronghuai Huang[21]	78.67%	0000-0003-4651-5248	Beijing Normal University, China	<i>Social Context Analysis for Topic-Specific Expert Finding in Online Learning Communities</i>
Humanities	Digital humanities; text analysis; cultural heritage;	Sergio Jimenez[22]	78.89%	0000-0003-3709-7881	Instituto Caro y Cuervo, Colombia	<i>ProficiencyRank: Automatically Ranking Expertise in Online</i>

Field	Pilot query	Researcher profile	Score	ORCID	Affiliation	Supporting publication
	historical					<i>Collaborative</i>
	documents					<i>Social Networks</i>

3.8. Discussion Against Research Objectives

The combined findings address the study objectives by showing how semantic representation, open scholarly metadata, vector retrieval and inspectable evidence operate together within ResearchMatch. The discussion below focuses on the implications rather than restating each result. For the first two objectives, the results support moving beyond keyword-only discovery while retaining bibliographic evidence. Semantic matching retrieved related concepts expressed with different terminology, whereas citation count and publication recency were not used as independent ranking features. The corpus was constructed from OpenAlex, CrossRef and DOAJ, demonstrating the accessibility of open metadata while also exposing dependence on identifier, affiliation and coverage completeness [1], [10].

The third and fourth objectives concerned lightweight semantic representation and vector retrieval. MiniLM-based embeddings supported comparison beyond exact word overlap, consistent with sentence and document embedding research [3], [5]. MongoDB Atlas enabled indexed vector retrieval with sub-second response times in the tested corpus and reduced local application-side memory pressure [16]. These are prototype-scale findings rather than claims about production performance.

The fifth and sixth objectives concerned evaluation and transparent evidence. The study reports latency, corpus-handling, semantic relevance and evidence inspection, while the six-query pilot adds comparative retrieval metrics across keyword, BM25, OpenAlex, citation-based, MiniLM and hybrid retrieval. ResearchMatch MiniLM outperformed the evaluated lexical and scholarly-search baselines on the principal pilot measures, and the hybrid BM25-MiniLM configuration performed best overall. Because the benchmark is small, the results remain preliminary rather than evidence of universal superiority.

ResearchMatch also returned similarity scores, profile keywords, ORCID links and supporting publications alongside ranked profiles. This supports reviewability in line with explainable recommendation principles [8], but it does not establish user trust, comprehension or decision quality. Those outcomes require formal user evaluation. Overall, ResearchMatch is best characterised as a semantic-bibliometric researcher-discovery prototype that combines thematic similarity with publication evidence. Its contribution is the integrated workflow, while its main boundaries are metadata quality, limited benchmark scale, unresolved author disambiguation, absence of formal user evaluation and untested multi-user deployment.

Open scholarly metadata can create unequal representation across researcher profiles. Researchers with complete abstracts, stable ORCID identifiers, current affiliations and stronger source coverage may receive richer textual and bibliographic representations than researchers with sparse records. Missing abstracts, absent identifiers and outdated affiliations can therefore affect profile construction and ranking visibility. Language and disciplinary coverage may add further inequality. The English-oriented semantic representation may favour records with English-language titles and abstracts, while source indexing differs across disciplines. These effects were not quantified and require dedicated metadata-coverage and fairness audits.

ResearchMatch rankings should therefore be interpreted as similarity to available indexed evidence rather than a neutral or complete representation of scholarly expertise. Author-name ambiguity, incomplete metadata and unequal profile completeness can also create misidentification or visibility risks. ResearchMatch should be used as a discovery aid, not as an autonomous basis for recruitment, funding, promotion, reviewer appointment or other formal decisions; users should verify identity, publications, affiliation, suitability, availability and relevant conflicts before acting on a result.

4. CONCLUSION

This article presented ResearchMatch, a semantic-bibliometric prototype for evidence-supported discovery of publication-based researcher profiles. The framework integrates OpenAlex, CrossRef and DOAJ metadata with sentence-transformers/all-MiniLM-L6-v2 embeddings and MongoDB Atlas Vector Search. Its contribution lies in integrating

established semantic, bibliometric and retrieval technologies into a functional researcher-discovery workflow rather than introducing a new retrieval algorithm. A six-query pilot across computer science, medicine, engineering, social science, education and humanities extended the earlier single-query demonstration. ResearchMatch MiniLM achieved P@5 of 0.77, MAP of 0.72 and nDCG@10 of 0.78, while the hybrid BM25-MiniLM configuration achieved 0.80, 0.76 and 0.82, respectively. These preliminary results suggest that semantic and lexical signals can provide complementary evidence for publication-based researcher retrieval. Representative cross-disciplinary outputs also included ORCID identifiers, affiliations and supporting publications, enabling users to inspect the evidence behind returned profiles. Similarity scores represent relative thematic alignment, not probabilities of expertise or measures of academic quality, and evidence availability does not establish user-perceived explainability because comprehension, trust and decision quality were not evaluated. The findings remain limited by the prototype setting. The database contains 18,532 stored profile records rather than independently verified unique researchers, and profile quality is affected by incomplete identifiers, author-name ambiguity, missing abstracts, affiliation variation, uneven indexing and language coverage. The six-query benchmark is too small to establish general effectiveness, fairness or production readiness. Future work should therefore extend expert-curated qrels across broader disciplinary and multilingual settings, compare lexical, dense and hybrid retrieval on larger benchmarks, strengthen author disambiguation, conduct metadata-coverage and fairness audits, evaluate user comprehension and trust, and test larger corpora and concurrent users. Production deployment would additionally require stronger privacy and security controls. ResearchMatch should consequently be treated as a researcher-discovery aid requiring human verification of identity, publication evidence and task-specific suitability.

REFERENCES

- [1] X.-L. Pham and T. T. Le, "Bibliometric analysis and systematic review of research on expert finding: A PRISMA-guided approach," *Int. Arab J. Inf. Technol.*, vol. 21, no. 4, pp. 661–674, Jul. 2024, doi: 10.34028/iajit/21/4/9.
- [2] X. Zhao and Y. Zhang, "Reviewer assignment algorithms for peer review automation: A survey," *Inf. Process. Manag.*, vol. 59, no. 5, Art. no. 103028, Sep. 2022, doi: 10.1016/j.ipm.2022.103028.

- [3] N. Reimers and I. Gurevych, "Sentence-BERT: Sentence embeddings using Siamese BERT-networks," in *Proc. 2019 Conf. Empirical Methods Natural Language Process. and 9th Int. Joint Conf. Natural Language Process. (EMNLP-IJCNLP)*, Hong Kong, China, Nov. 2019, pp. 3982–3992, doi: 10.18653/v1/D19-1410.
- [4] I. Beltagy, K. Lo, and A. Cohan, "SciBERT: A pretrained language model for scientific text," in *Proc. 2019 Conf. Empirical Methods Natural Language Process. and 9th Int. Joint Conf. Natural Language Process. (EMNLP-IJCNLP)*, Hong Kong, China, Nov. 2019, pp. 3615–3620, doi: 10.18653/v1/D19-1371.
- [5] A. Cohan, S. Feldman, I. Beltagy, D. Downey, and D. Weld, "SPECTER: Document-level representation learning using citation-informed transformers," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, Online, Jul. 2020, pp. 2270–2282, doi: 10.18653/v1/2020.acl-main.207.
- [6] J. Priem, H. Piwowar, and R. Orr, "OpenAlex: A fully-open index of scholarly works, authors, venues, institutions, and concepts," in *Proc. 26th Int. Conf. Science, Technology and Innovation Indicators (STI 2022)*, Granada, Spain, Sep. 2022, doi: 10.5281/zenodo.6936227.
- [7] E. Álvarez-García, D. García-Costa, I. De Waele, A. Marusic, and F. Grimaldo, "Expert assignment system based on natural language processing for Marie Skłodowska-Curie actions," *Sci. Rep.*, vol. 16, no. 1, Art. no. 6396, Jan. 2026, doi: 10.1038/s41598-026-37115-8.
- [8] Y. Zhang and X. Chen, "Explainable recommendation: A survey and new perspectives," *Found. Trends Inf. Retr.*, vol. 14, no. 1, pp. 1–101, 2020, doi: 10.1561/15000000066.
- [9] S. Khalid *et al.*, "Comprehensive review of academic search systems: Evolution, analysis, and future research directions," *Soc. Netw. Anal. Min.*, vol. 15, Art. no. 66, Jul. 2025, doi: 10.1007/s13278-025-01476-1.
- [10] L. M. de Campos, J. M. Fernández-Luna, J. F. Huete, F. J. Ribadas-Pena, and N. Bolaños, "Information retrieval and machine learning methods for academic expert finding," *Algorithms*, vol. 17, no. 2, Art. no. 51, Jan. 2024, doi: 10.3390/a17020051.
- [11] Y. Wang, J. Liu, X. Xu, X. Ke, T. Wu, and X. Gou, "Efficient and effective academic expert finding on heterogeneous graphs through (k, P)-core based embedding," *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 6, pp. 1–35, 2023, doi: 10.1145/3578365.
- [12] I. A. Mannix and E. Yulianti, "Academic expert finding using BERT pre-trained language model," *Int. J. Adv. Intell. Inform.*, vol. 10, no. 2, pp. 280–295, May 2024, doi: 10.26555/ijain.v10i2.1497.

- [13] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *Adv. Neural Inf. Process. Syst.*, vol. 33, pp. 5776–5788, 2020.
- [14] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987, doi: 10.1016/0377-0427(87)90125-7.
- [15] G. Hendricks, D. Tkaczyk, J. Lin, and P. Feeney, "Crossref: The sustainable source of community-owned scholarly metadata," *Quant. Sci. Stud.*, vol. 1, no. 1, pp. 414–427, 2020, doi: 10.1162/qss_a_00022.
- [16] Deepak and S. Sheoran, "Vector search in large language models: Experimental evaluation with MongoDB Atlas," *J. Cybersecur. Inf. Manag.*, vol. 17, no. 2, pp. 146–166, 2026, doi: 10.54216/JCIM.170211.
- [17] L. E. Cheby, "Open access metadata for journals in Directory of Open Access Journals: Who, how, and what scheme?," *School Inf. Student Res. J.*, vol. 6, no. 1, Art. no. 4, 2016, doi: 10.31979/2575-2499.060104.
- [18] H. Singh, R. Singh, A. Malhotra, and M. Kaur, "Developing a biomedical expert finding system using Medical Subject Headings," *Healthc. Inform. Res.*, vol. 19, no. 4, pp. 243–249, Dec. 2013, doi: 10.4258/hir.2013.19.4.243.
- [19] M. Y. Bayraktar and M. Kaya, "Expert recommendation system to promote development-focused universities," *Türk Doğa ve Fen Derg.*, vol. 13, no. 2, pp. 30–33, Jun. 2024, doi: 10.46810/tdfd.1397626.
- [20] D. Wu, S. Fan, and F. Yuan, "Research on pathways of expert finding on academic social networking sites," *Inf. Process. Manag.*, vol. 58, no. 2, Art. no. 102475, Mar. 2021, doi: 10.1016/j.ipm.2020.102475.
- [21] Y. Li, S. Ma, and R. Huang, "Social context analysis for topic-specific expert finding in online learning communities," in *Smart Learning Environments*, M. Chang and Y. Li, Eds. Berlin, Germany: Springer, 2015, pp. 57–74, doi: 10.1007/978-3-662-44447-4_4.
- [22] S. Jimenez, F. N. Silva, G. Dueñas, and A. Gelbukh, "ProficiencyRank: Automatically ranking expertise in online collaborative social networks," *Inf. Sci.*, vol. 588, pp. 231–247, Apr. 2022, doi: 10.1016/j.ins.2021.11.067.