Rethinking Information Retrieval in a Re-Decentralised Web: Exploring the Feasibility and Quality of Search Across Personal Online Datastores

MOHAMMAD BAHRANI, University of Southampton, Southampton, United Kingdom of Great Britain and Northern Ireland
MOHAMED RAGAB, University of Southampton, Southampton, United Kingdom of Great Britain and Northern Ireland
HELEN OLIVER, Birkbeck University of London, London, United Kingdom of Great Britain and Northern Ireland
THANASSIS TIROPANIS, University of Southampton, Southampton, United Kingdom of Great Britain and Northern Ireland
ADRIANE CHAPMAN, University of Southampton, Southampton, United Kingdom of Great Britain and Northern Ireland
ALEXANDRA POULOVASSILIS, Birkbeck University of London, London, United Kingdom of Great Britain and Northern Ireland
GEORGE ROUSSOS, Birkbeck University of London, London, United Kingdom of Great Britain and Northern Ireland

Traditional information retrieval (IR) models, such as keyword-based and vector-based techniques, have long been used in centralized systems. However, the Web's re-decentralization, with its focus on data ownership and privacy, calls for a re-evaluation of these methods in these settings. While standards for decentralized search enhance privacy to some extent, they also introduce computational overhead, black-box decision-making, and infrastructure complexity. Despite these challenges, traditional IR techniques remain largely unexplored in such environments. This paper presents an innovative application of traditional IR models in the decentralized Web by adapting them for Personal Online Data Stores (PODs), where search parties have varying access rights. We explore their role in source selection, document ranking, and result merging, extending them to meet decentralized search demands. Using `Solid` PODs and a synthetic medical dataset, we evaluate these models in a privacy-sensitive environment. Our findings demonstrate that extended IR methods provide an effective balance of performance, interpretability, and efficiency. These approaches hold strong potential as privacy-preserving alternatives for decentralized search on a re-decentralized Web. Notably, our top-performing model achieved competitive results in top-item retrieval compared to centralized search systems, maintaining high relevance scores under both limited and full data access conditions.

Authors' Contact Information: Mohammad Bahrani, University of Southampton, Southampton, United Kingdom of Great Britain and Northern Ireland; e-mail: m.bahrani@soton.ac.uk; Mohamed Ragab, University of Southampton, Southampton, England, United Kingdom of Great Britain and Northern Ireland; e-mail: ragab.mohamed@soton.ac.uk; Helen Oliver, Birkbeck University of London, London, England, United Kingdom of Great Britain and Northern Ireland; e-mail: h.oliver@bbk.ac.uk; Thanassis Tiropanis, University of Southampton, Southampton, England, United Kingdom of Great Britain and Northern Ireland; e-mail: t.tiropanis@soton.ac.uk; Adriane Chapman, University of Southampton, Southampton, England, United Kingdom of Great Britain and Northern Ireland; e-mail: adriane.chapman@soton.ac.uk; Alexandra Poulovassilis, Birkbeck University of London, London, England, United Kingdom of Great Britain and Northern Ireland; e-mail: a.poulovassilis@bbk.ac.uk; George Roussos, Birkbeck University of London, London, England, United Kingdom of Great Britain and Northern Ireland; e-mail: g.roussos@bbk.ac.uk.

This work is licensed under a Creative Commons Attribution 4.0 International License.
© 2025 Copyright held by the owner/author(s).

ACM 1559-114X/2025/11-ART
https://doi.org/10.1145/3777445

Just Accepted

CCS Concepts: • **Information systems → Information retrieval**; **Search engine architectures and scalability**; *Search engine indexing*; *Distributed retrieval*; • **Computing methodologies** → *Artificial intelligence*; • **Computer systems organization** → *Distributed architectures.*

Additional Key Words and Phrases: Decentralized AI, Personal Online Data stores, Decentralized Web, Decentralized search

1 Introduction

This paper introduces the first application of standardizing traditional Information Retrieval (IR) techniques for search in Personal Online Data Stores (PODs), proposing IR models tailored for privacy-preserving settings based on both keyword and vector-based approaches.

The re-decentralization of the Web, driven by the increasing demand for data ownership and privacy, necessitates a re-evaluation of IR techniques, particularly in the context of PODs. Decentralized search faces significant challenges, including privacy risks, computational overhead, black-box decision-making, and infrastructure complexity. However, traditional IR, grounded in decades of research, has proven effective in centralized settings and, with proper adaptation, can provide scalable and privacy-conscious solutions for search in the re-decentralized Web.

Our work builds on the Efficient Search over Personal Repositories – Secure and Sovereign (ESPRESSO) project [35, 38] [1], which introduced a structured infrastructure comprising access control-aware indexes and multi-level metainformation (at the POD, POD hosting server, and overlay network levels). This architecture has been demonstrated to be both feasible and efficient for decentralized, privacy-preserving search tasks.

The ESPRESSO platform offers a solid foundation for developing and experimenting with new methods. Its three-layered architecture of indexes and meta-information—spanning the POD, POD hosting server, and overlay network levels—provides the flexibility to explore various approaches to source selection at different granularities. This layered structure enables a comprehensive re-evaluation of traditional IR techniques within a decentralized and privacy-preserving context. However, the current indexing scheme in ESPRESSO does not model the entire dataset; user indexes include only the portions of the data they have access to, which leads to reduced ranking quality. Additionally, the previous work was only an initial step toward applying traditional IR in PODs and lacked a comprehensive approach.

In this work, we introduce a generalizable and transparent multi-level ranking framework for search across and within PODs, implemented on top of the ESPRESSO infrastructure. The framework supports both keyword-based and vector-based ranking across multiple stages, including source selection, results ranking, and merging. Crucially, it operates independently of the number of decentralization layers involved. Our approach addresses the challenge of balancing effective information retrieval with strict access controls that constrain index completeness and impact model performance.

For keyword-based IR, we propose methods that leverage user-specific indexes—distributed indexes securely stored within users' PODs—along with dataset metadata. Unlike centralized IR systems that aggregate data from all sources, our approach constructs separate indexes for each user, organized at various levels (e.g., per document, POD, POD server, and network of servers). This structure facilitates secure decentralized search, treating each search step as a distinct textual document ranking task.

For vector-based ranking, we investigate both sparse and dense vector retrieval methods. In the sparse vector approach, we transform queries and documents into vectors that capture statistics derived from user-specific indexes. This method allows the retention of knowledge about the

[1] https://espressoproject.org/.

Just Accepted

entire dataset while ensuring sensitive information is kept secure and private to the parties that have access to it. For dense vector-based retrieval, we propose a novel technique for fine-tuning embeddings. These embeddings are trained and refined using user-specific indexes, ensuring that patterns and relationships remain within the privacy constraints of the user.

This study evaluates the effectiveness of the proposed techniques by comparing overall ranking accuracy and top-k retrieval performance against an ideal per-user centralized baseline, defined as a single index built only over the documents each user can access, with a proven centralized search method applied. Despite challenges such as handling additional sources (e.g., PODs beyond just a server or node), user and access control management, and decentralization, our approach achieves results comparable to the centralized baseline.

The contribution of this paper is threefold. First, we introduce a generalizable multi-level framework for source selection, ranking, and result merging in federated search, as detailed in Section 3. Building on this framework, we propose a family of keyword-based privacy-preserving ranking models in Section 4 for federated search across and within PODs. Additionally, we address the challenge that vector-based approaches in federated machine learning (prediction) often require access to global statistics for effective training—posing privacy risks and potentially causing traffic and system overload. In Section 5, we propose solutions to mitigate these privacy concerns while preserving the effectiveness of vector-based models. Our experimental results demonstrate that keyword-based models can achieve comparable performance to an ideal per-user centralized baseline operating on the same access-controlled subset. Additionally, our fine-tuning approach for word embeddings yields high-quality results when data access exceeds 50%.

Just Accepted

2 Challenges and Framework for Privacy-Preserving Retrieval in PODs

This section presents the ESPRESSO framework for applying IR across and within PODs for both keyword-based and vector-based ranking, and outlines the key challenges of adapting traditional IR techniques to this decentralized context.

2.1 Challenges in POD-Based IR

In decentralized ecosystems, data is stored in PODs controlled by individual users. Access permissions are managed through mechanisms such as Access Control Lists (ACLs) [42, 43], which define who can read, write, or share specific data. These controls ensure that data remains user-governed and securely accessible across different applications.

A key challenge in privacy-preserving search is ensuring that retrieval remains effective while restricting access to only authorized documents. In ecosystems where access is controlled by ACLs, indexes built solely from an individual's accessible subset may fail to capture meaningful relationships across the broader dataset. This limitation can lead to lower-quality rankings and reduced retrieval effectiveness. Federated learning approaches, such as *FedAvg* [27], attempt to mitigate this by continuously updating models, but they introduce significant computational overheads.

2.2 Hybrid Retrieval Approach

To address this, we propose a hybrid approach that integrates user-specific indexes with general dataset metadata, all securely managed by POD owners. While searchers can only retrieve documents they are authorized to view, their user-specific indexes lack broader corpus-level insights, such as term distributions and global document statistics. To overcome this, we leverage the ESPRESSO overlay network, which organizes logically structured tables to provide non-sensitive global statistics, enhancing retrieval effectiveness without compromising privacy.

This gives rise to two distinct families of privacy-preserving retrieval models that can be applied at each stage of the multi-level ranking process—server-level, POD-level, and document-level. **User-specific** models rely exclusively on the data accessible to an individual at a given level, while **global** models leverage aggregated statistical insights from the overlay network to improve ranking effectiveness. In the following sections, we formalize both model types within the paradigms of keyword-based and vector-based ranking.

Notation	**Definition**	**Source**	**Data Structure**
Public (global) IR Features			
$\mathrm{df}_{\mathrm{global}}$	Document frequency	**Overlay Network**	**Relational Database**
$\lvert c\rvert_{\mathrm{global}}$	Collection length		
μ_{global}	Average entity length		
$\mathrm{IDF}_{\mathrm{global}}$	Inverse document frequency		
$\mathrm{tf}_{\mathrm{global}}$	Global term frequency		
$\mathrm{N}_{\mathrm{global}}$	Document count		
$\mathrm{D}(\varphi)_{\mathrm{global}}$	Entity length normalization factor		
Private (user-specific) IR Features			
$\mathrm{tf}_{\mathrm{user}}$	Term frequency	**POD** —> *for document ranking* **ESPRESSO POD** —> *for source selection*	**Index**
$\mathrm{df}_{\mathrm{user}}$	Document frequency		
$\lvert c\rvert_{\mathrm{user}}$	Collection length		
$\lvert\varphi\rvert_{\mathrm{user}}$	Entity length		
μ_{user}	Average entity length		
$\mathrm{IDF}_{\mathrm{user}}$	Inverse document frequency		
$\mathrm{D}(\varphi)_{\mathrm{user}}$	Entity length normalization factor		

Table 1. Notations used in public (global) and private (user-specific) IR models, distinguishing between retrieval statistic sources (POD vs. Overlay Network) and their corresponding data structures (Relational Database Tables and Document-based Index).

As detailed in our previous papers [35, 38], the ESPRESSO infrastructure comprises SOLID-based search and indexing applications, a metadata manager, and an overlay network. It utilizes GaianDB[2] [3] as a federated storage layer for metadata about servers. More than just a P2P overlay network, GaianDB enables distributed data storage and retrieval, following the "Store Locally, Query Anywhere" principle. It propagates queries efficiently, retrieving results via the shortest paths. Fig.1 illustrates the ESPRESSO framework's design and its core components that support search across PODs.

2.2.1 Notations and Data Representation. Table 1 provides an overview of the notations used in our models, clearly distinguishing between user-specific and global metadata. To illustrate this distinction, the table introduces user and global subscripts within the listed features. For example, $\mathrm{df}_{\mathrm{user}}$ represents the document frequency of a term within the user-specific index, whereas $\mathrm{df}_{\mathrm{global}}$ denotes the term's document frequency across the entire dataset. As shown in the table, the user-specific index is stored in the user's POD, while metadata for POD selection—relevant to the query—is also user-specific and physically resides in the ESPRESSO POD on the user's server. The

[2] IBM GaianDB https://github.com/gaiandb/gaiandb

Just Accepted

Just Accepted

ESPRESSO POD is a dedicated POD on each server that maintains metadata specific to that server. However, non-sensitive metadata about the entire dataset is stored in the overlay network's logical tables, which are updated by the ESPRESSO POD manager on each server using data from the ESPRESSO POD.

2.2.2 Keyword-based and Vector-based Approach. Unlike traditional IR systems, which rank results based on a single specified collection and index — whether using probabilistic (keyword-based) or vector-based similarity models - our approach ensures that retrieval features originate from a privacy-preserving physical space where features could be decoupled from the index. This approach offers several advantages, including:

(1) **Reduced latency**: It avoids the overhead related to cross-index merging which is computationally expensive.
(2) **Rare query-term handling**: The occurrence of terms can vary across different indexes. A user-specific index enables a tailored analysis of the requester's user data, providing more relevant and specialized handling of rare query terms.
(3) **Full IR model but still privacy-preserving**: By leveraging global collection length and other features, we can still build a comprehensive model with strict access control (no need to query other indexes). Employing a single, unified model throughout the search process (source selection and ranking) significantly enhances interpretability and transparency. Moreover, this generalizable model is not restricted by the number or type of sources, such as PODs, or servers, making it adaptable across different levels of the system architecture.
(4) **Optimized resource usage**: By querying user-specific indexes instead of the entire index, the system consumes fewer resources (e.g., memory and computational power). This is especially beneficial in ecosystems like `SOLID` [3] [43], where HTTP requests can become costly [38]. Additionally, traditional federated search ranking, which typically involves multiple result merging steps, is streamlined by performing the merge only once.

Vector-based IR represents documents and queries as mathematical vectors, using techniques such as TF-IDF [40] or word embeddings [29]. These vectors enable similarity measurement through methods like cosine similarity or Euclidean distance, with neural networks further refining these representations. However, these vectors pose privacy risks in PODs search due to the exposure of sensitive data during training and the potential leakage of sensitive patterns or relationships. To address this, we introduce user-specific vectors, which leverage metadata from the overlay network's logical tables and propose embeddings that are trained and fine-tuned using user-specific indexes. These vectors are securely stored within each user's POD. Unlike traditional federated learning, which relies on gradient updates and embedding aggregation, our approach updates global statistics from the overlay network while preserving user-specific vectors. While federated search ensures privacy through techniques like differential privacy [11], homomorphic encryption [16], and secure aggregation [5, 24], these methods can be complex, costly, and still pose some privacy risks. Separating sensitive data from general metadata for search could offer a more privacy-preserving alternative.

3 Generalizable Multi-Level Ranking Structure

In this section, we introduce a generalizable multi-level ranking structure that supports entity-based ranking across the key stages of federated search: source selection, result ranking, and merging. This structure is designed to address the challenges of interpretability and complexity inherent in federated search systems.

[3] https://solidproject.org/

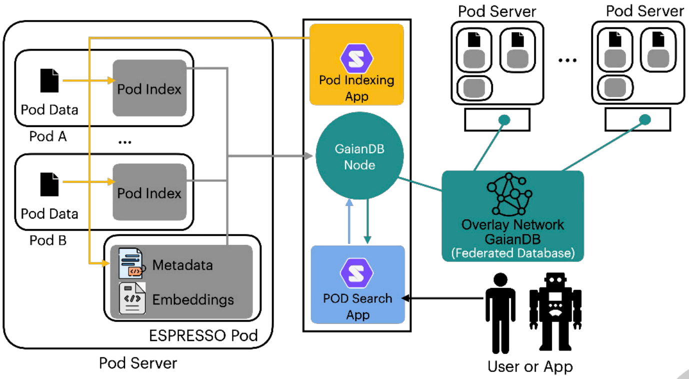

Fig. 1. ESPRESSO Framework Architecture

User-Specific Indexing and Metadata Design. A user-specific index refers to an index containing only the subset of data accessible to a given user, as determined by ACLs. We employ three types of metadata: system-level metadata (statistical features used for server selection), server-level metadata (statistical features for POD selection within each server), and document indexes (used for document ranking at the POD level). A detailed discussion and flow of the metadata generation is provided below.

- **System-level Metadata:** The first step is to extract system-level metadata, such as term frequency distributions across servers, in a user-specific manner. These metadata are stored in overlay network logical tables and are continuously updated by reading the ESPRESSO POD at each server through the relevant network node.
- **Server-level Metadata:** After system-level metadata is gathered, we identify relevant servers and rank them based on specific criteria. Each server contains multiple PODs, and we retrieve the corresponding PODs from each selected server. The ESPRESSO POD maintains the user-specific indexes on that server, which are securely stored as zipped archives.
- **Document Indexes:** Once the relevant PODs have been selected and retrieved, the next step is to focus on retrieving documents from within these PODs. Each POD contains multiple user-specific indexes, which are structured similarly to the POD-level index. These indexes store information about documents and are organized according to ACLs to ensure appropriate access control.

Multi-level Ranking. We use here a flexible method for defining ranking entities, φ, which can represent different levels of granularity: documents, PODs (sets of documents), or server-level aggregations of documents. We use the symbol φ to denote an *entity* to be ranked (or selected), with each entity being defined at different levels of granularity depending on the task. At each stage of ranking (document, POD, server), we generalize the ranking process by defining φ appropriately. The mathematical formulation of φ at each stage is expressed as follows:

$$\varphi = \begin{cases} d, & \textit{Document ranking: individual document} \\ \bigcup_{i=1}^{N_P} d_i, & \textit{POD ranking: aggregation of all documents in a POD} \\ \bigcup_{i=1}^{N_S} \left(\bigcup_{j=1}^{N_i} d_{ij} \right), & \textit{Server ranking: aggregation of all documents across PODs in a server} \end{cases} \quad (1)$$

The specific definition of φ depends on the level of ranking, as detailed below:

Just Accepted

- **Document ranking:** Let d be a single document. At the document level, φ represents an individual textual document stored in a POD, i.e.,

$$\varphi = d \tag{2}$$

- **POD ranking:** To rank PODs, we construct a single representative document by aggregating all individual documents within a POD into a single textual document. This allows us to frame the problem of POD ranking as a textual document ranking task. Specifically,

$$\varphi = \bigcup_{i=1}^{N_P} d_i \tag{3}$$

where N_P is the number of documents in a POD, and d_i denotes the i^{th} document in the POD.

- **Server ranking:** Let $P = \{P_1, P_2, \ldots, P_N\}$ represent the set of PODs hosted on the server accessible to the user, where P_i denotes the individual POD i. Each P_i contains a set of documents $\{d_{i1}, d_{i2}, \ldots, d_{iN_i}\}$, where N_i is the number of documents in P_i.
Similar to POD-level ranking, we define a large document generated by aggregating the textual contents from all documents accessible to the user across the PODs hosted on a server S. Specifically, let N_S be the number of PODs in S, then, φ is defined as the concatenation of all documents from each POD in server S:

$$\varphi = \bigcup_{i=1}^{N_S} \left(\bigcup_{j=1}^{N_i} d_{ij} \right) \tag{4}$$

4 Privacy-preserving keyword-based IR

In this section, we introduce the user-specific and global approaches for keyword-based IR and formulate these variants for BM25 [39] and Language Modeling (LM) [33], as detailed in Section 4.1 and section 4.2. While we focus on these two models, the approach is generalizable to other keyword-based models as well.

4.1 User-Specific and Global BM25 Formulation

The BM25 weight $\mathrm{W}_{\mathrm{BM25}}$ for a term t in an entity φ belongs to the set of documents accessible to the searcher's user $\mathcal{P}_{\text{user}}$ ($\varphi \in \mathcal{P}_{\text{user}}$) is defined as:

$$\mathrm{W}_{\mathrm{BM25,Type}}(t, \varphi \mid \mathcal{P}_{\text{user}}) = \mathrm{IDF}_{\mathrm{Type}}(t) \cdot \frac{\mathrm{tf}_{\mathrm{Type}}(t, \varphi) \cdot (k_1 + 1)}{\mathrm{tf}_{\mathrm{Type}}(t, \varphi) + k_1 \cdot \mathrm{D}_{\mathrm{Type}}(\varphi)} \tag{5}$$

Here, k_1 is the saturation parameter for scaling term frequency, and b determines the level of document length normalization. $\mathrm{IDF}_{\mathrm{Type}}(t)$ represents the Inverse Document Frequency of term t (as defined in Equation 7). The *Type* parameter, *Type* $\in$ {user, global}, indicates whether the collection-wide statistics are specific to the searcher's user (from the index) or computed globally from the entire dataset (from logical tables) (see Table 1). The function $\mathrm{D}(\varphi)$ accounts for length normalization and is defined as follows:

$$\mathrm{D}_{\mathrm{Type}}(\varphi) = 1 - b + b \cdot \frac{|\varphi|}{\mu_{\varphi,\mathrm{Type}}} \tag{6}$$

where $\mu_{\varphi,\text{user}}$ and $\mu_{\varphi,\text{global}}$ represent the user-specific and global average entity lengths, respectively. If $\mathrm{D}_{\mathrm{Type}}(\varphi)$ is computed using user-specific statistics, the weight model is denoted as $\mathrm{W}_{\mathrm{BM25,user}}$. Conversely, if global statistics are used, it is denoted as $\mathrm{W}_{\mathrm{BM25,global}}$. The IDF function determines term importance based on its distribution across entities:

Just Accepted

$$\mathrm{IDF}_{\mathrm{Type}}(t) = \log\left(\frac{\mathrm{N}_{\mathrm{Type}} - \mathrm{df}_{\mathrm{Type}}(t) + 0.5}{\mathrm{df}_{\mathrm{Type}}(t) + 0.5}\right) + 1 \tag{7}$$

where $\mathrm{N}_{\mathrm{Type}}$ is the total number of entities, and $\mathrm{df}_{\mathrm{Type}}(t)$ is the number of entities containing term t. These values depend on the selected *Type*, results in two variants: $\mathrm{IDF}_{\mathrm{user}}(t)$ and $\mathrm{IDF}_{\mathrm{global}}(t)$.

The Retrieval Score Value (RSV) for a query Q and an entity φ is computed as:

$$\mathrm{RSV}_{\mathrm{BM25,Type}}(Q, \varphi \mid \mathcal{P}_{\mathrm{user}}) = \sum_{t \in Q} W_{\mathrm{BM25,Type}}(t, \varphi) \tag{8}$$

Depending on whether the weighting function is computed using user-specific or global statistics, RSV exists in two variants:

$$\mathrm{RSV}_{\mathrm{BM25,user}}(Q, \varphi) \quad \text{and} \quad \mathrm{RSV}_{\mathrm{BM25,global}}(Q, \varphi). \tag{9}$$

4.2 User-Specific and Global Language Model (LM) Formulation

The language model (LM) weight for a term t in an entity φ, given a query Q, is computed as:

$$\mathrm{W}_{\mathrm{LM,Type}}(t, \varphi \mid \mathcal{P}_{\mathrm{user}}, Q) = \mathrm{tf}(t, Q) \cdot \log\left((1 - \lambda_d) + \lambda_d \cdot \frac{\alpha_d(t \mid \mathrm{Type})}{\alpha_c(t \mid \mathrm{Type})}\right) \tag{10}$$

where λ_d is the smoothing parameter controlling the balance between the entity-specific model and the background model, and $\mathrm{tf}(t, Q)$ represents the frequency of term t in query Q. The entity-level language model, $\alpha_d(t \mid \mathrm{Type})$, and the background model probability, $\alpha_c(t \mid \mathrm{Type})$, are defined as:

$$\alpha_d(t \mid \mathrm{Type}) = \frac{\mathrm{tf}_{\mathrm{Type}}(t, \varphi)}{|\varphi|}, \quad \alpha_c(t \mid \mathrm{Type}) = \frac{\mathrm{df}_{\mathrm{Type}}(t)}{\sum_{t'} \mathrm{df}_{\mathrm{Type}}(t')} \tag{11}$$

where $\mathrm{df}_{\mathrm{Type}}(t)$ denotes the number of entities containing term t. The *Type* parameter determines whether the background model statistics are user-specific ($\alpha_c(t \mid \mathrm{user})$) or computed globally ($\alpha_c(t \mid \mathrm{global})$). Accordingly, the weighting function is denoted as $\mathrm{W}_{\mathrm{LM,user}}$ or $\mathrm{W}_{\mathrm{LM,global}}$.

The Retrieval Score Value (RSV) for a query Q and entity φ is then computed as:

$$\mathrm{RSV}_{\mathrm{LM,Type}}(Q, \varphi \mid \mathcal{P}_{\mathrm{user}}) = \sum_{t \in Q} W_{\mathrm{LM,Type}}(t, \varphi \mid \mathcal{P}_{\mathrm{user}}, Q) \tag{12}$$

where $\varphi \in \mathcal{P}_{\mathrm{user}}$. Similar to BM25, two scoring variants exist: (i) user-specific LM, when the background model is user-specific ($\alpha_c(t \mid \mathrm{user})$), denoted as $\mathrm{RSV}_{\mathrm{LM,user}}(Q, \varphi)$, and (ii) global LM, when the background model is global ($\alpha_c(t \mid \mathrm{global})$), denoted as $\mathrm{RSV}_{\mathrm{LM,global}}(Q, \varphi)$.

5 Privacy-preserving vector-based IR

In traditional IR, vector-based approaches transform documents and queries into numerical representations, enabling similarity computations through standard methods such as cosine similarity or inverted Euclidean distance [41]. These representations fall into two main categories: sparse representations, which use high-dimensional, mostly zero-valued vectors (e.g., TF-IDF), and dense representations, which map text into lower-dimensional, continuous-valued vectors (e.g., word embeddings, transformer-based embeddings). However, conventional vector-based methods expose sensitive information, and are thus not suitable for privacy-preserving search in PODs. In this section, we introduce novel methodologies and solutions for achieving privacy-preserving vector-based IR that ensures confidentiality.

Just Accepted

Just Accepted

5.1 Sparse vector-based retrieval

To make sparse vector-based IR privacy-preserving, we adopt the approach proposed for keyword-based IR, as discussed in Section 4. This approach integrates both user-specific and global metadata from Table 1 into the model (which, in this case, is a vector transformer). To evaluate the sparse vector-based models in this paper, we use TF-IDF vectorization for both documents and queries. The equation below shows the TF-IDF weight for term t in document φ:

$$\mathrm{W}_{\mathrm{TF.IDF}}(t, \varphi) = \mathrm{TF}_{\mathrm{user}}(t, \varphi) \cdot \mathrm{IDF}_{\mathrm{global}}(t) \tag{13}$$

Given a vocabulary of V terms, a document φ (or query q) is represented as a TF-IDF vector:

$$\mathbf{v}_{\mathrm{TF.IDF}}(\varphi) = (\mathrm{W}_{\mathrm{TF.IDF}}(t_1, \varphi), \mathrm{W}_{\mathrm{TF.IDF}}(t_2, \varphi), ..., \mathrm{W}_{\mathrm{TF.IDF}}(t_V, \varphi)) \in \mathbb{R}^V \tag{14}$$

Given this vectorization, equations 15 and 16 present the corresponding RSV formulas for ranking the entity φ in response to the query Q based on cosine similarity and inverse Euclidean distance.

$$\mathrm{RSV}_{\mathrm{TF.IDF,cosine}}(Q, \varphi \mid \mathcal{P}_{\mathrm{user}}) = \frac{\mathbf{v}_{\mathrm{TF.IDF}}(Q) \cdot \mathbf{v}_{\mathrm{TF.IDF}}(\varphi)}{\|\mathbf{v}_{\mathrm{TF.IDF}}(Q)\|\|\mathbf{v}_{\mathrm{TF.IDF}}(\varphi)\|} \tag{15}$$

$$\mathrm{RSV}_{\mathrm{TF.IDF,Euclidean}}(Q, \varphi \mid \mathcal{P}_{\mathrm{user}}) = -\|\mathbf{v}_{\mathrm{TF.IDF}}(Q) - \mathbf{v}_{\mathrm{TF.IDF}}(\varphi)\| \tag{16}$$

5.2 Dense vector-based retrieval

Dense vector representations, such as those derived from word embeddings, inherently pose a risk of leaking information about other users. This is due to factors such as the exposure of sensitive data during training and the potential leakage of sensitive patterns or relationships that could compromise privacy. To address this, we propose fine-tuning word embeddings using user-specific indexes stored in PODs, ensuring that they remain localized within users' PODs to preserve privacy.

While our approach is applicable to any word embedding model, in this work, we focus on Word2Vec [29] as a case study. Algorithm 1 details the process of training user-specific embeddings and leveraging cosine similarity and Euclidean distance to rank documents in a privacy-preserving manner.

6 Motivating Scenario, Experimental Design, and Evaluation

In this section, we present a motivating scenario that serves as the foundation for our experiments, followed by a detailed description of the experimental setup, results and analysis.

6.1 Motivating scenario

A federated health data network, where patients retain ownership of their medical records across multiple hospitals, allows researchers to query patient data while ensuring patients' privacy and consent. For a detailed description of this use case, we refer readers to our previous work [36].

Algorithm 1 Fine-Tuning Word2Vec and Ranking Documents with user-Specific Embeddings

```
**Part 1: Fine-Tuning Word2Vec for user-Specific Embeddings**
**Input:** Pre-trained Word2Vec model $M_{\text{base}}$, Set of user-specific index ZIP files $\mathcal{Z}$, List of users $\mathcal{W}$
**Output:** Fine-tuned Word2Vec models $\{M_w\}$ for each user $w$
**for all** $z \in \mathcal{Z}$ **do**
    Extract documents for users: $\mathcal{D}_z \leftarrow \text{ExtractDocuments}(z)$
    **for all** $w \in \mathcal{W}$ **do**
        $\mathcal{D}_w \leftarrow \{d \in \mathcal{D}_z \mid \text{AccessibleBy}(d, w)\}$
        $\mathcal{D}_w^{\text{tokens}} \leftarrow \{\text{Preprocess}(d) \mid d \in \mathcal{D}_w\}$
        **Filtering:** Remove numbers, dates, and stopwords
        $M_w \leftarrow \text{Word2Vec}(\text{vector\_size} = 300, \text{window} = 5, \text{min\_count} = 5, \text{sg} = 1, \text{epochs} = 10)$
        $\text{BuildVocab}(M_w, \mathcal{D}_w^{\text{tokens}})$
        **for all** word $\in M_w$ **do**
            **if** word $\in M_{\text{base}}$ **then**
                Copy vector from $M_{\text{base}}$
            **end if**
        **end for**
        $\text{Train}(M_w, \mathcal{D}_w^{\text{tokens}}, \text{epochs} = 5)$
        $\text{Save}(M_w, \text{"user\_"}w\text{"\_Model.bin"})$
    **end for**
**end for**
**Return** $\{M_w\}$

**Part 2: Ranking Documents Using Fine-Tuned Embeddings**
**Input:** Query $Q$, List of documents $\mathcal{D}$, user $w$, Word2Vec repository $R$
**Output:** Ranked list of documents $R_{\text{sorted}}$
$M_w \leftarrow \text{LoadWord2VecModel}(R, w)$
$Q_{\text{tokens}} \leftarrow \text{Preprocess}(Q, M_w)$
$Q_{\text{vec}} \leftarrow \text{Vectorize}(Q_{\text{tokens}}, M_w)$
$R_{\text{results}} \leftarrow [\,]$
**for all** $d \in \mathcal{D}$ **do**
    $d_{\text{vec}} \leftarrow \text{Vectorize}(\text{Preprocess}(d), M_w)$
    Compute similarity scores:
    $\text{cosineSim}(Q, d) \leftarrow \text{CosineSimilarity}(Q_{\text{vec}}, d_{\text{vec}})$
    $\text{invEDist}(Q, d) \leftarrow \text{InverseEuclideanDistance}(Q_{\text{vec}}, d_{\text{vec}})$
    Append $(d, \text{cosineSim}, \text{invEuclidDist})$ to $R_{\text{results}}$
**end for**
Sort $R_{\text{results}}$ by cosineSim and invEuclidDist
**Return** ranked list $R_{\text{sorted}}$
```

Table 2. Parameters of the Synthetic Decentralized Patient Data Dataset.

Parameter	Description
Number of Solid Servers	50 servers
Number of PODs/Server	9500 PODs
Number of Documents/POD	1 document per POD (medical history)
Data Size per POD	~5KB to ~750KB
Access Control	10% of PODs grant access
users Used	Access levels: (5%, 10%, 25%, 50%, 100%)
POD Index Size	~4KB to ~8KB (via user)
Server Index Size	~2.5MB to ~65MB (based on user)
Overlay Network Index Size	~123MB to ~3GB (based on user access)

6.2 Implementation

We employ the ESPRESSO framework in our experiments here. Our earlier work [35, 38], provides validation of the performance and efficiency of ESPRESSO. This prior work outlines the challenges, design principles, and a first prototype system for decentralized search. The IR experimental setup

Just Accepted

presented here builds on a companion manuscript currently in revision for resubmission [34]. We briefly summarise the aspects most pertinent to our contributions.

Building on this foundation, here we investigate new approaches to standardize the integration of centralized IR techniques, indexing, and embeddings within and across PODs. The following sections describe the specific components and methods used in our implementation, including the use of SOLID-based PODs, indexing, decentralized search, and the training of embeddings.

- **SOLID-based PODs Implementation** In ESPRESSO, we use SOLID [43] for managing access control, leveraging its WebIds and ACLs. Data, along with its ACL, is stored within each user's POD, corresponding to their WebId, and hosted on SOLID servers, adhering to SOLID protocols.
- **Decentralized search** The ESPRESSO Search App is a Solid-based application based on Node.js and Axios library to execute authenticated HTTP GET requests, ensuring appropriate WebId-based authorization before retrieving index and embedding files from PODs.
- **Indexing** We used Apache Lucene [17] to generate the indexes at different levels of granularity (network-level, POD-level and data-level) for each WebId based on PODs' ACLs. Building a WebID-specific Lucene index scales approximately linearly with the number of accessible documents. In our experiments, indexing time ranged from a few minutes at 5% access to just under two hours at 100% access per WebID. We expect that index updates will follow a similar trend, though conducting performance evaluations of update operations remains an area for future work.
- **Training embeddings** For each user, we fine-tuned a pre-trained Gensim Word2Vec model [29] using the documents accessible to that user, thereby generating user-specific embeddings for privacy-preserving retrieval within Solid PODs. The text data was pre-processed by removing numbers, dates, and stop-words. The embeddings were fine-tuned using a 300-dimensional skip-gram model with a context window of 5. Pre-trained vectors were leveraged where applicable, and training was extended for five additional epochs. The resulting fine-tuned embeddings were then stored within the respective PODs. Fine-tuning time scaled with the number of tokens and epochs; in our setup this ranged from a few minutes at 5% access to roughly half a day at 100% per WebID, reported as indicative wall-clock times rather than systematic benchmarks.

6.3 Experimental setup

The experiments are designed to evaluate the retrieval quality of our approach in comparison to centralized search. The analysis of the results provides insights into the overall performance of the models, helps identify when and why each model should be used, and reveals any associated limitations.

- **Environment:** A cluster of 50 Solid servers was set up to represent a network of general health practitioners (GPs) in a metropolitan area. Each GP provides a Solid server for patients to store their confidential medical records in individual PODs. The servers run on Virtual Machines (VMs) with Red Hat Enterprise Linux 8.7, 2.4GHz processors, 8GB RAM, and 125GB storage. A separate VM hosts the search app, featuring a 32-core processor, 132GB RAM, and 1TB storage. The ESPRESSO system utilizes Community Solid Server (v6.0) and a customized GaianDB (v2.1.8).
- **Synthetic Setup for Decentralized Patient Data Retrieval:** We generated a synthetic dataset of decentralized documents representing patient data using *Synthea*[4], an open-source

[4] https://github.com/synthetichealth/synthea

Just Accepted

tool that creates synthetic patient data, including medical histories. Each data item corresponds to a complete medical history of a single patient. For our experiments, we populated the PODs with one data item per patient. Each document is associated with a unique patient and is stored within a single POD, mimicking the real-world scenario where NHS patient data is securely managed within individual PODs. The documents are distributed across the 50 servers, with each server hosting 9,500 PODs, reflecting the average size of a UK GP practice. Specifically, the average UK GP practice serves around 9,500 patients [5], and Cornwall, for example, has 57 GP practices [6]. In total, we sampled 475,000 documents from this synthetic dataset. The parameters for the decentralized patient data, along with server and index configurations at different levels, are provided in Table 2.

- **WebId Selection and Access Distribution:** Each document was assigned to a WebId from a list of 250, simulating doctors accessing patient files. To explore different access scenarios, we created virtual search parties, each granted access to randomly selected portions of the data across all servers. In the experiments, four WebIds were chosen, each with a different access level: one with 5%, another with 10%, a third with 25%, and the fourth with 50% of the dataset. This setup enables a comprehensive analysis of how varying access levels influence retrieval performance under different privacy-preserving conditions.
- **Data Sparsity:** While each POD in our setup contains one complete document, data sparsity naturally emerges through access control. Specifically, WebId-specific indexes include only the subset of documents a user is permitted to access. As access levels vary, the evaluation captures conditions ranging from sparse (with relatively small indexes) to complete data availability, thereby exposing the models to missing and fragmented data.
- **Query-set Generation:** Experiments were conducted using 30 search queries of varying lengths, generated via Latent Dirichlet Allocation (LDA) [4] with 50 topics.
- **Centralized Relevance Benchmark:** Our evaluation here aims to assess the quality of the proposed models by comparing them to a centralized IR approach. To establish user-specific relevance judgments, we create a separate centralized index for each selected WebId, based on the documents they have access to (e.g., 5%, 10%, 25%, and 50% access levels). Each index consolidates the relevant documents for that WebId into a single, centralized directory. For each WebId, we execute queries over their corresponding centralized index, rank the results using the BM25 relevance score, and consider the top-ranked documents as the gold standard (i.e., the true relevant answers). This process results in a unique gold standard for each WebId based on the documents they have access to.
  This setup provides a benchmark for measuring the accuracy of our privacy-preserving models by evaluating how closely their results align with the gold standards generated from the centralized IR baseline for each user.

Just Accepted

7 Evaluation: privacy-preserving Keyword-based IR:

For evaluation, we selected a combination of standard IR metrics that together provide a comprehensive view of model accuracy and quality: P@k (Precision at rank k), reporting the fraction of relevant results in the top-k documents; nDCG (Normalized Discounted Cumulative Gain, measuring overall ranking quality) and nDCG@k (its cutoff version at rank k), rewarding higher placement of relevant documents; and MAP (Mean Average Precision), capturing the overall averaged precision across all queries.

[5] Data taken from https://www.gponline.com/fifth-gp-practices-closed-merged-nhs-england-formed/article/1790429

[6] Data taken from https://cios.icb.nhs.uk/health/primary-care/

Table 3 presents the evaluation results for keyword-based global models (BM25 and LM) across varying server selection cut-offs (top S servers, ranging from 1 to 5) for top-5 PODs in each server and different access levels. The models are evaluated at cutoff points k = 5, 10, and 20 using these retrieval metrics.

The results reveal a linear correlation between performance and S, which is intuitive—retrieval improves as more servers are selected. At lower access levels, $\text{BM25}_{\text{global}}$ and $\text{LM}_{\text{global}}$ yield similar performance. However, as access increases, $\text{BM25}_{\text{global}}$ significantly outperforms $\text{LM}_{\text{global}}$ at both 50% and 100% access. This advantage stems from BM25's *term frequency saturation* (which prevents overemphasis on frequent terms) and *IDF weighting* (which enhances the differentiation of important terms). Further analysis of the behavior of these two models is presented in figure 4.

At lower access levels (e.g., 5% and 10%), top-5 and top-10 retrieval achieve excellent results, with values such as 0.915 for nDCG@5 for BM25 at 5% access.

Figures 2 and 3 illustrate the linear performance of two variants of BM25: BM25_{user} and $\text{BM25}_{\text{global}}$, evaluated against top-$S$ servers and various metrics, including nDCG@5, nDCG@10, nDCG@20, P@5, P@10 and MAP, for access levels of 5% and 10%. We report this representative subset of evaluation metrics for clarity; complete results, including P@20, are provided in Tables 3 and 4. As expected, the global variant generally outperforms the user-specific variant. However, at $S = 1$, $\text{BM25}_{\text{user}}$ produces results that are very close to those of $\text{BM25}_{\text{global}}$. This can be explained by the fact that with strict server cut-offs, the number of available documents is limited, and in such cases, there is less need for global term distribution information across the entire dataset.

Just Accepted

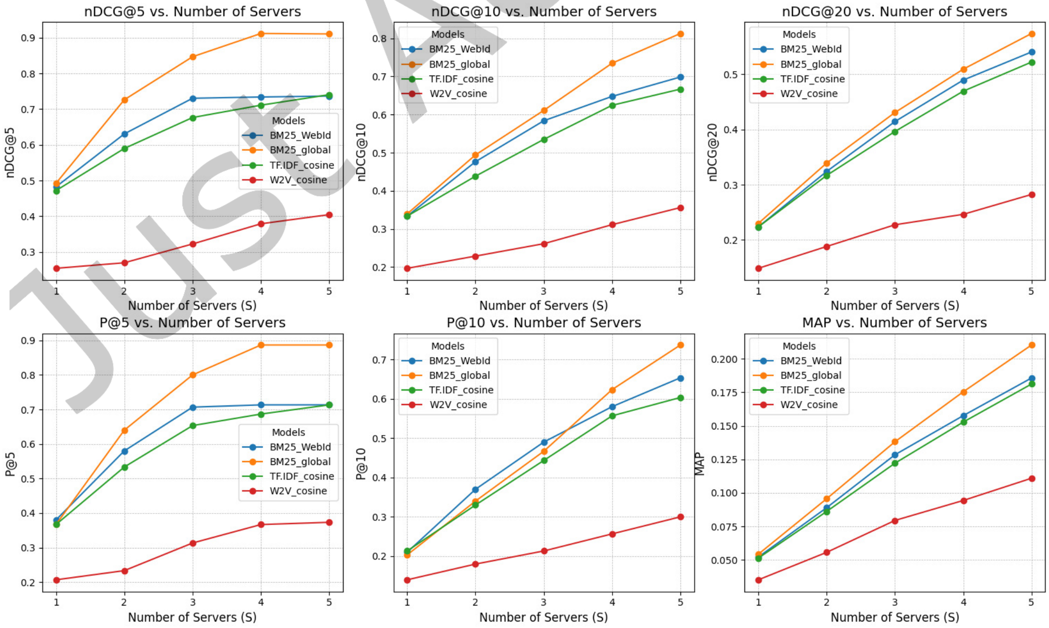

Fig. 2. Performance of top server selection with 5% access across models: Comparing ranking-based selection vs. centralized search. Higher values indicate better performance.

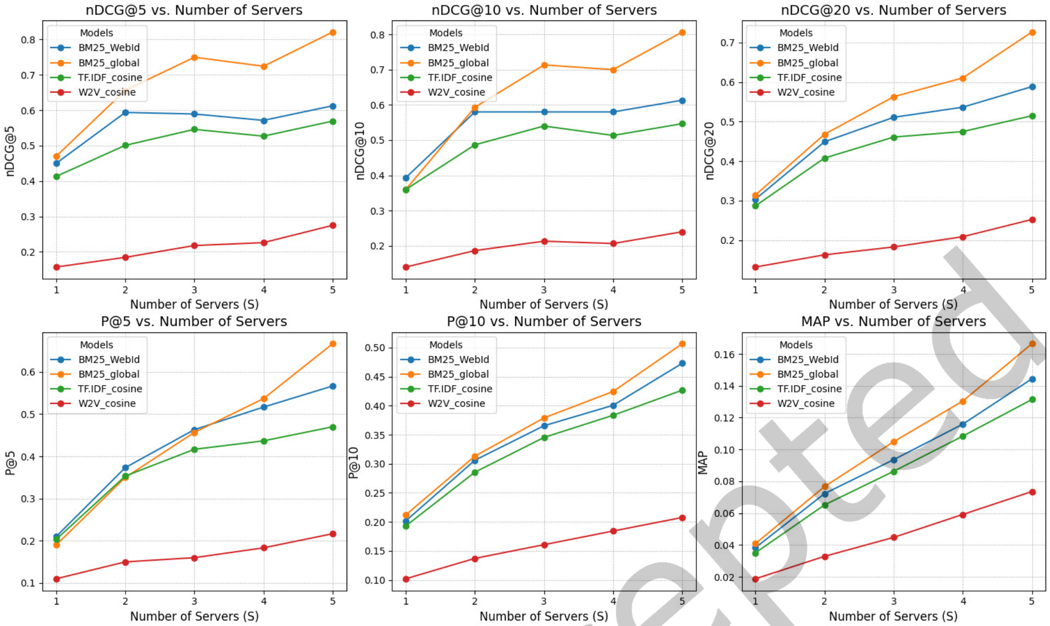

Fig. 3. Performance of top server selection with 10% access across models: Comparing ranking-based selection vs. centralized search. Higher values indicate better performance.

Table 3. Evaluation of keyword-based global models by Source Selection: Performance comparison of ranking vs. centralized search (BM25 without source selection) by varying the number of top S servers (top-5 PODs). Higher values indicate better performance. Bold font indicates the best result for each evaluation metric within its respective access level category.

Keyword-based Evaluation: Top Servers Performance Across Global Models									
Model	**Access (%)**	**Top (S)**	**nDCG@5**	**P@5**	**nDCG@10**	**P@10**	**nDCG@20**	**P@20**	**MAP**
BM25$_{global}$	5%	S=1	0.519	0.413	0.342	0.206	0.229	0.103	0.054
	5%	S=2	0.767	0.693	0.540	0.393	0.361	0.196	0.101
	5%	S=3	0.866	0.826	0.650	0.516	0.451	0.278	0.143
	5%	S=4	**0.922**	**0.900**	**0.766**	0.666	0.542	0.375	0.181
	5%	S=5	0.915	0.893	0.827	**0.756**	**0.605**	**0.450**	**0.216**
LM$_{global}$	5%	S=1	0.514	0.413	0.339	0.206	0.227	0.103	0.053
	5%	S=2	0.730	0.673	0.523	0.393	0.350	0.196	0.097
	5%	S=3	0.796	0.780	0.614	0.506	0.432	0.278	0.135
	5%	S=4	0.824	0.800	0.729	0.656	0.521	0.375	0.171
	5%	S=5	0.825	0.793	0.758	0.693	0.582	0.448	0.202
BM25$_{global}$	10%	S=1	0.508	0.413	0.331	0.207	0.217	0.103	0.042
	10%	S=2	0.696	0.647	0.518	0.410	0.339	0.205	0.083
	10%	S=3	0.775	0.746	0.615	0.523	0.420	0.283	0.114
	10%	S=4	0.823	0.813	0.714	0.650	0.509	0.378	0.151
	10%	S=5	**0.846**	**0.840**	**0.764**	**0.713**	**0.560**	**0.435**	**0.178**
LM$_{global}$	10%	S=1	0.488	0.413	0.319	0.207	0.209	0.103	0.039
	10%	S=2	0.609	0.573	0.488	0.410	0.320	0.205	0.075
	10%	S=3	0.659	0.620	0.566	0.497	0.399	0.283	0.103
	10%	S=4	0.716	0.687	0.644	0.590	0.486	0.378	0.137
	10%	S=5	0.715	0.700	0.651	0.607	0.525	**0.435**	0.158
BM25$_{global}$	25%	S=1	0.402	0.320	0.261	0.160	0.169	0.080	0.029
	25%	S=2	0.569	0.507	0.403	0.300	0.260	0.150	0.054
	25%	S=3	0.728	0.693	0.536	0.433	0.356	0.230	0.081
	25%	S=4	0.770	0.747	0.604	0.517	0.412	0.288	0.102
	25%	S=5	**0.824**	**0.813**	**0.688**	**0.620**	**0.472**	**0.348**	**0.126**

Just Accepted

Model	Access (%)	Top (*S*)	nDCG@5	P@5	nDCG@10	P@10	nDCG@20	P@20	MAP
$\text{LM}_{\text{global}}$	25%	*S*=1	0.364	0.320	0.236	0.160	0.153	0.080	0.025
	25%	*S*=2	0.456	0.453	0.352	0.300	0.227	0.150	0.045
	25%	*S*=3	0.573	0.573	0.460	0.410	0.316	0.230	0.066
	25%	*S*=4	0.572	0.560	0.510	0.473	0.368	0.288	0.083
	25%	*S*=5	0.591	0.580	0.543	0.513	0.417	**0.348**	0.099
$\text{BM25}_{\text{global}}$	50%	*S*=1	0.452	0.360	0.294	0.180	0.189	0.090	0.032
	50%	*S*=2	0.642	0.573	0.447	0.327	0.288	0.163	0.059
	50%	*S*=3	0.750	0.707	0.554	0.443	0.364	0.230	0.083
	50%	*S*=4	0.790	0.760	0.629	0.540	0.418	0.287	0.101
	50%	*S*=5	**0.810**	**0.780**	**0.686**	**0.613**	**0.464**	**0.337**	**0.112**
$\text{LM}_{\text{global}}$	50%	*S*=1	0.428	0.360	0.278	0.180	0.179	0.090	0.029
	50%	*S*=2	0.552	0.500	0.414	0.327	0.267	0.163	0.051
	50%	*S*=3	0.626	0.587	0.495	0.417	0.336	0.230	0.069
	50%	*S*=4	0.583	0.540	0.522	0.470	0.375	0.287	0.079
	50%	*S*=5	0.592	0.547	0.536	0.487	0.412	0.335	0.092
$\text{BM25}_{\text{global}}$	100%	*S*=1	0.341	0.260	0.222	0.130	0.143	0.065	0.023
	100%	*S*=2	0.497	0.413	0.347	0.240	0.224	0.120	0.041
	100%	*S*=3	0.635	0.573	0.446	0.333	0.296	0.178	0.062
	100%	*S*=4	0.705	0.667	0.519	0.417	0.352	0.232	0.079
	100%	*S*=5	**0.780**	**0.740**	**0.615**	**0.520**	**0.419**	**0.291**	**0.102**
$\text{LM}_{\text{global}}$	100%	*S*=1	0.310	0.260	0.201	0.130	0.130	0.065	0.020
	100%	*S*=2	0.409	0.393	0.297	0.240	0.191	0.120	0.032
	100%	*S*=3	0.473	0.473	0.369	0.323	0.251	0.178	0.045
	100%	*S*=4	0.485	0.480	0.416	0.380	0.299	0.232	0.058
	100%	*S*=5	0.549	0.540	0.493	0.463	0.359	0.288	0.076

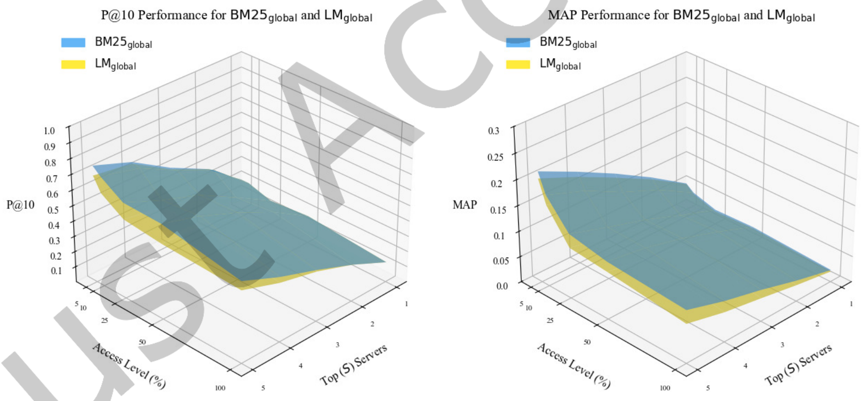

Fig. 4. MAP and P@10 Performance comparison of global keyword-based models (LM and BM25) across varying access control levels and different server selection cut-offs. Both LM and BM25 show similar results at lower access levels, but as access increases, BM25 outperforms LM, demonstrating better performance with higher access control.

8 Evaluation: privacy-preserving vector-based IR:

Table 4 presents the performance of various vector-based models, including sparse vectorization models $\text{TF.IDF}_{\text{cosine}}$, $\text{TF.IDF}_{\text{Euclidean}}$) and dense vectorization models (i.e., $\text{W2V}_{\text{cosine}}$, $\text{W2V}_{\text{Euclidean}}$), across different servers (ranging from 1 to 5) for the top-5 PODs on each server and varying access levels. The experimental results indicate that the W2V models, which leverage fine-tuned

Just Accepted

embeddings based on user-specific indexes, outperform TF. IDF-based models in terms of similarity only when the access level is high. This is particularly promising for scenarios where users have access to 50% or more of the data, suggesting that fine-tuning is a robust approach in these cases. Conversely, when access levels are lower, sparse retrieval methods, such as TF.IDF, remain more effective. This is likely because at lower access levels the fine-tuned model tends to overfit and drift away from the original pre-trained semantics, where the risk of catastrophic forgetting is higher. This explanation is consistent with intuition and with prior work on fine-tuning with small corpora [10, 22], where limited training material often results in reduced generalization. Approaches such as employing pre-trained initialization with limited updates, stronger regularization, or capacity control could help mitigate this effect. Further experimentation will be required to confirm this hypothesis, and we leave such investigations to future work. By contrast, with around 50% or more access, the index provides enough data to capture meaningful semantic relations, supporting adaptation while still retaining useful general semantics. These findings are further explored in figure 5. Notably, for very low access levels (e.g., 5% and 10%), user-specific fine-tuning is not beneficial, as shown in figures 2 and 3, where W2V performs poorly compared to TF.IDF. Additionally, sparse TF.IDF models exhibit performance trends similar to those of keyword-based global models, with performance fluctuations corresponding to changes in the number of servers (S) and access level.

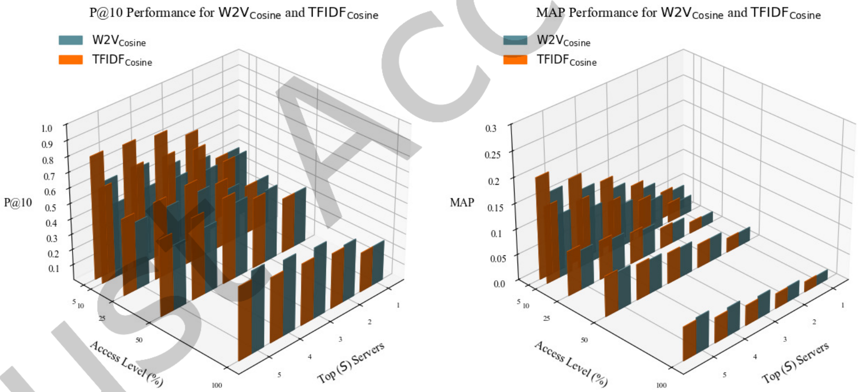

Fig. 5. MAP and P@10 Performance comparison of vector-based models with cosine similarity (W2V and TFIDF) across varying access control levels and different server selection cut-offs. W2V (fine-tuned) performs better with higher access control, showing particularly strong results at the 50% access level and outperforming TF-IDF at access levels above 50%.

Just Accepted

Table 4. Evaluation of vector-based models by Source Selection: Performance comparison of ranking vs. centralized search (BM25 without source selection) by varying the number of top *S* servers (top-5 PODs). Higher values indicate better performance. Bold font indicates the best result for each evaluation metric within its respective access level category.

Vector-based Evaluation: Top Servers Performance Across Models

Model	Access (%)	Top (*S*)	nDCG@5	P@5	nDCG@10	P@10	nDCG@20	P@20	MAP
TF.IDF$_{\text{cosine}}$	5%	*S*=**1**	0.522	0.413	0.344	0.207	0.231	0.103	0.054
	5%	*S*=**2**	0.714	0.660	0.519	0.393	0.348	0.197	0.097
	5%	*S*=**3**	0.769	0.747	0.613	0.5133	0.429	0.278	0.134
	5%	*S*=**4**	0.797	0.780	0.706	0.637	0.515	0.375	0.169
	5%	*S*=**5**	**0.821**	**0.800**	**0.757**	**0.700**	**0.585**	**0.458**	**0.200**
TF.IDF$_{\text{Euclidean}}$	5%	*S*=**1**	0.454	0.413	0.300	0.207	0.202	0.103	0.046
	5%	*S*=**2**	0.489	0.473	0.437	0.393	0.295	0.197	0.079
	5%	*S*=**3**	0.455	0.427	0.462	0.433	0.360	0.278	0.108
	5%	*S*=**4**	0.468	0.440	0.457	0.423	0.429	0.375	0.129
	5%	*S*=**5**	0.477	0.460	0.446	0.407	0.442	0.393	0.147
W2V$_{\text{cosine}}$	5%	*S*=**1**	0.450	0.413	0.297	0.207	0.200	0.103	0.047
	5%	*S*=**2**	0.481	0.453	0.440	0.393	0.297	0.197	0.080
	5%	*S*=**3**	0.519	0.513	0.457	0.413	0.366	0.278	0.109
	5%	*S*=**4**	0.588	0.593	0.504	0.450	0.450	0.375	0.137
	5%	*S*=**5**	0.608	0.607	0.552	0.507	0.479	0.402	0.161
W2V$_{\text{Euclidean}}$	5%	*S*=**1**	0.450	0.413	0.297	0.207	0.200	0.103	0.047
	5%	*S*=**2**	0.480	0.453	0.439	0.393	0.296	0.197	0.080
	5%	*S*=**3**	0.515	0.507	0.459	0.417	0.365	0.278	0.109
	5%	*S*=**4**	0.584	0.587	0.504	0.450	0.450	0.375	0.137
	5%	*S*=**5**	0.605	0.600	0.552	0.507	0.479	0.402	0.161
TF.IDF$_{\text{cosine}}$	10%	*S*=**1**	0.480	0.413	0.313	0.207	0.205	0.103	0.039
	10%	*S*=**2**	0.597	0.587	0.476	0.410	0.312	0.205	0.074
	10%	*S*=**3**	0.643	0.633	0.560	0.513	0.389	0.283	0.101
	10%	*S*=**4**	**0.684**	0.660	0.614	0.563	0.477	0.378	0.132
	10%	*S*=**5**	0.672	**0.653**	**0.620**	**0.580**	**0.513**	**0.432**	**0.152**
TF.IDF$_{\text{Euclidean}}$	10%	*S*=**1**	0.435	0.413	0.284	0.207	0.186	0.103	0.035
	10%	*S*=**2**	0.456	0.440	0.434	0.410	0.285	0.205	0.065
	10%	*S*=**3**	0.418	0.427	0.405	0.400	0.331	0.283	0.079
	10%	*S*=**4**	0.406	0.413	0.394	0.387	0.395	0.378	0.099
	10%	*S*=**5**	0.401	0.400	0.393	0.380	0.397	0.378	0.115
W2V$_{\text{cosine}}$	10%	*S*=**1**	0.427	0.413	0.279	0.207	0.183	0.103	0.033
	10%	*S*=**2**	0.437	0.413	0.430	0.410	0.283	0.205	0.063
	10%	*S*=**3**	0.477	0.460	0.445	0.420	0.350	0.283	0.083
	10%	*S*=**4**	0.487	0.487	0.456	0.437	0.418	0.378	0.107
	10%	*S*=**5**	0.499	0.480	0.463	0.430	0.430	0.387	0.123
W2V$_{\text{Euclidean}}$	10%	*S*=**1**	0.431	0.413	0.281	0.207	0.185	0.103	0.034
	10%	*S*=**2**	0.441	0.413	0.433	0.410	0.284	0.205	0.063
	10%	*S*=**3**	0.474	0.453	0.444	0.417	0.350	0.283	0.083
	10%	*S*=**4**	0.492	0.493	0.454	0.433	0.418	0.378	0.107
	10%	*S*=**5**	0.503	0.487	0.464	0.430	0.428	0.383	0.123
TF.IDF$_{\text{cosine}}$	25%	*S*=**1**	0.342	0.320	0.222	0.160	0.143	0.080	0.023
	25%	*S*=**2**	0.416	0.433	0.332	0.300	0.214	0.150	0.041
	25%	*S*=**3**	**0.517**	**0.520**	0.447	0.417	0.305	0.230	0.063
	25%	*S*=**4**	0.506	0.507	0.481	0.467	0.351	0.288	0.077
	25%	*S*=**5**	0.480	0.500	**0.478**	**0.487**	**0.384**	**0.348**	**0.088**
TF.IDF$_{\text{Euclidean}}$	25%	*S*=**1**	0.332	0.320	0.216	0.160	0.139	0.080	0.022
	25%	*S*=**2**	0.337	0.307	0.321	0.300	0.207	0.150	0.037
	25%	*S*=**3**	0.366	0.360	0.324	0.303	0.268	0.230	0.048
	25%	*S*=**4**	0.333	0.340	0.313	0.307	0.297	0.288	0.055
	25%	*S*=**5**	0.303	0.307	0.308	0.310	0.288	0.280	0.063
W2V$_{\text{cosine}}$	25%	*S*=**1**	0.326	0.320	0.212	0.160	0.137	0.080	0.022
	25%	*S*=**2**	0.372	0.360	0.327	0.300	0.211	0.150	0.039
	25%	*S*=**3**	0.456	0.440	0.400	0.367	0.293	0.230	0.056
	25%	*S*=**4**	0.465	0.440	0.422	0.390	0.340	0.288	0.069
	25%	*S*=**5**	0.438	0.440	0.412	0.403	0.353	0.323	0.077
W2V$_{\text{Euclidean}}$	25%	*S*=**1**	0.326	0.320	0.211	0.160	0.136	0.080	0.022
	25%	*S*=**2**	0.373	0.360	0.328	0.300	0.212	0.150	0.039
	25%	*S*=**3**	0.456	0.440	0.400	0.367	0.294	0.230	0.057
	25%	*S*=**4**	0.475	0.453	0.426	0.393	0.341	0.288	0.069
	25%	*S*=**5**	0.444	0.453	0.412	0.403	0.350	0.318	0.077
TF.IDF$_{\text{cosine}}$	50%	*S*=**1**	0.422	0.360	0.274	0.180	0.177	0.090	0.029
	50%	*S*=**2**	0.497	0.460	0.392	0.327	0.253	0.163	0.047
	50%	*S*=**3**	0.566	0.560	0.438	0.380	0.313	0.230	0.062

Just Accepted

Model	Access (%)	Top (*S*)	nDCG@5	P@5	nDCG@10	P@10	nDCG@20	P@20	MAP
	50%	*S*=**4**	**0.538**	**0.533**	0.452	0.410	0.351	0.287	0.070
	50%	*S*=**5**	0.515	0.500	**0.465**	**0.433**	**0.375**	**0.323**	**0.080**
TF.IDF$_{\text{Euclidean}}$	50%	*S*=**1**	0.375	0.360	0.244	0.180	0.157	0.090	0.023
	50%	*S*=**2**	0.325	0.333	0.321	0.327	0.207	0.163	0.034
	50%	*S*=**3**	0.336	0.333	0.308	0.297	0.260	0.230	0.044
	50%	*S*=**4**	0.336	0.327	0.300	0.280	0.298	0.287	0.052
	50%	*S*=**5**	0.311	0.307	0.294	0.283	0.282	0.272	0.057
W2V$_{\text{cosine}}$	50%	*S*=**1**	0.417	0.360	0.271	0.180	0.175	0.090	0.028
	50%	*S*=**2**	0.506	0.473	0.391	0.327	0.253	0.163	0.047
	50%	*S*=**3**	0.513	0.493	0.434	0.387	0.307	0.230	0.060
	50%	*S*=**4**	0.470	0.433	0.440	0.403	0.345	0.287	0.067
	50%	*S*=**5**	0.450	0.420	0.450	**0.433**	0.357	0.310	0.076
W2V$_{\text{Euclidean}}$	50%	*S*=**1**	0.417	0.360	0.271	0.180	0.175	0.090	0.028
	50%	*S*=**2**	0.504	0.473	0.390	0.327	0.252	0.163	0.047
	50%	*S*=**3**	0.518	0.493	0.437	0.387	0.310	0.230	0.060
	50%	*S*=**4**	0.477	0.440	0.439	0.400	0.346	0.287	0.067
	50%	*S*=**5**	0.455	0.427	0.451	**0.433**	0.360	0.313	0.076
TF.IDF$_{\text{cosine}}$	100%	*S*=**1**	0.318	0.260	0.206	0.130	0.133	0.065	0.021
	100%	*S*=**2**	0.378	0.360	0.288	0.240	0.186	0.120	0.030
	100%	*S*=**3**	0.395	0.400	0.338	0.317	0.234	0.178	0.040
	100%	*S*=**4**	0.409	0.420	0.360	0.343	0.277	0.232	0.050
	100%	*S*=**5**	0.469	0.473	0.411	0.390	0.334	**0.290**	0.066
TF.IDF$_{\text{Euclidean}}$	100%	*S*=**1**	0.264	0.260	0.171	0.130	0.111	0.065	0.015
	100%	*S*=**2**	0.189	0.173	0.226	0.240	0.146	0.120	0.019
	100%	*S*=**3**	0.206	0.193	0.192	0.180	0.190	0.178	0.027
	100%	*S*=**4**	0.217	0.227	0.190	0.187	0.219	0.232	0.031
	100%	*S*=**5**	0.227	0.240	0.205	0.200	0.212	0.213	0.040
W2V$_{\text{cosine}}$	100%	*S*=**1**	0.301	0.260	0.195	0.130	0.126	0.065	0.020
	100%	*S*=**2**	0.390	0.373	0.292	0.240	0.188	0.120	0.031
	100%	*S*=**3**	0.462	0.460	0.366	0.323	0.249	0.178	0.045
	100%	*S*=**4**	0.479	0.487	0.406	0.377	0.294	0.232	0.056
	100%	*S*=**5**	**0.512**	**0.527**	**0.453**	**0.433**	**0.345**	**0.290**	**0.070**
W2V$_{\text{Euclidean}}$	100%	*S*=**1**	0.301	0.260	0.195	0.130	0.126	0.065	0.020
	100%	*S*=**2**	0.396	0.380	0.293	0.240	0.189	0.120	0.031
	100%	*S*=**3**	0.453	0.460	0.361	0.323	0.246	0.178	0.044
	100%	*S*=**4**	0.471	0.487	0.401	0.377	0.290	0.232	0.055
	100%	*S*=**5**	0.500	0.520	0.443	0.427	0.342	**0.290**	0.069

Just Accepted

9 Evaluation: sensitivity analysis by query type

Averaged measures such as MAP or nDCG cannot reveal why and when a query performs well, and may risk being influenced by a few outliers. To validate the robustness of our findings, we conduct a per-query sensitivity analysis focusing on two representative models, **BM25$_{\text{global}}$** and **W2V$_{\text{cosine}}$**. Queries are grouped into *short* (1–3 keywords), *medium* (4–5), and *long* (6+). For each group and server-selection cut-off (1–5), we report the difference (BM25$_{\text{global}}$ − W2V$_{\text{cosine}}$) across six metrics (nDCG@5/10/20, P@5/10, MAP), averaging over access-control levels (Fig. 6).

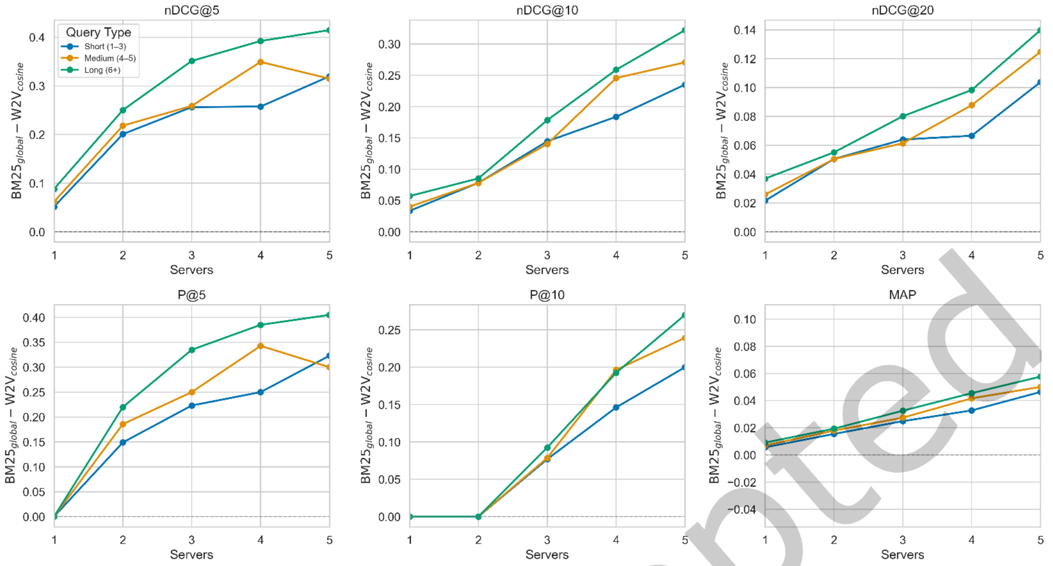

Fig. 6. Difference between **$BM25_{global}$** and **$W2V_{cosine}$** evaluation scores across query types (short, medium, long) and varying numbers of server selections (1–5). Results are averaged over access-control levels. The zero line marks parity between models; positive values (as observed here) indicate stronger performance for $BM25_{global}$.

Just Accepted

Across all conditions the gap is strictly positive, indicating that $BM25_{global}$ consistently outperforms $W2V_{cosine}$. The gap widens with more servers and correlates with query length, especially for *long* queries at early cut-offs (P@5 and nDCG@5). MAP gaps are smaller in magnitude, as expected, but consistently increase with server count. These results confirm that $BM25_{global}$'s advantage is robust across query types and not driven by a handful of queries.

10 Related Work

In this section, we situate ESPRESSO within the context of related research: first in federated search, then in decentralized AI. Finally, we discuss the crisis of confidence that traditional IR is experiencing with the emergence of generative AI. We argue that traditional IR is a fundamentally sound approach to decentralized search that is consistent with the values of data sovereignty that ESPRESSO aims to support.

10.1 Federated Search

Federated search environments are divided into two categories: cooperative or uncooperative [14].

An example of a cooperative environment is an enterprise database, in which all the resources are owned by one entity. In a cooperative environment, the data owner provides term statistics and content information; STARTS [18] is an example of a data representation technique for a cooperative environment.

In an uncooperative environment, data owners do not provide information about statistics or content, because resources are owned by different entities and there is no standard protocol for providing information about them. In uncooperative environments, of which the current Web is considered a real-world example, resources are represented using query-based techniques over samples [6].

Much of the current literature on federated search assumes an uncooperative environment [15]. This is because, in real-world web settings, most databases and search systems (e.g., Amazon, Bing, PubMed) are uncooperative [13] and federated search is often employed in these contexts to provide aggregated results across such independent data sources.

ESPRESSO can treat each POD as a cooperative environment because all the documents in it are owned by the POD owner. This allows ESPRESSO to create an accurate representation of the data in the POD and store it within the POD itself. Accuracy is balanced with privacy by representing the data at different levels of granularity at each of ESPRESSO's indexing levels - POD, server, and global network.

10.2 Decentralized AI

As touched upon in section 2.2.2, the privacy-preserving techniques required by federated search and AI incur complexity and cost [23] which are to be balanced against the performance of traditional IR. Zheng et al. [48] address the issue of cost in their approach to decentralized recommendation, which relies on local gradient calculation and global gradient passing. Their approach also addresses the reality that data items in real-world search are neither completely open nor completely private, but will be selectively shared with users other than the data owner. Looking at the problem from the point of view of the search environment: while search is not among the use cases considered by Meurisch, Bayrak and Mühlhäuser [28], they propose a privacy-preserving platform for personalized AI services that is specifically designed for decentralized personal online datastores, including Solid. This approach strictly confines personal data processing within the POD [12], training the personalized model locally from cloud-based general models, and allowing for limited model sharing between 'similar' users. The approach follows the principle of separating sensitive data from global data (see section 2.2.2) as a means to preserve privacy in decentralized AI, but does not address use cases like ESPRESSO's, which require some level of integration between the global and the local in order for data items to be selectively discoverable by users other than the data owner. This integration is achieved in ESPRESSO by its layered retrieval approach (section 2.2).

10.3 Privacy-preserving IR

Privacy-preserving IR came to prominence in 2014 [45] [44]. A diverse field of inquiry emerged from this initial call to action. At the same time, decentralized personal data stores were emerging, including Mydex [30], Hub-of-All-Things [31], and in 2018, Solid. In 2019, the FACTS-IR workshop identified 'the implementation of confidentiality-aware end user search' as a requirement for safeguarding against information leakage [32]. It was in 2024 that Soboroff wrote: "Overall, the area of privacy for document owners is under-studied, possibly because the dominant paradigm in Web search is centralized search engines." [46]

A decentralized retrieval model by Mahmoud et al. [26] uses friend-of-a-friend connections to build a log for each user of trusted potential networks, and the logs are built into topic models. A similar approach, leveraging access control lists, could inform ESPRESSO's server and POD profiling for query routing.

Recently, the ascendancy of AI, particularly generative AI, has led some practitioners to question whether traditional IR is still relevant [2]. Hersh, quoting Harman [19] refocuses the argument on the core task of IR, which is to guide the search party to relevant existing resources [21].

In [2], Verbene underlines the inherent transparency of linking to a source and leaving the search party to decide whether or not to follow the link, and to interpret the linked content for themselves. Traditional IR is a human-in-the-loop approach, consistent with the principles of data sovereignty that ESPRESSO search aims to support. ESPRESSO is the first approach to query processing in decentralized, federated, or Linked Data environments that takes differential access control into

Just Accepted

account [35]. In this context, generative AI is a consideration for future work, and analytical AI is of more immediate relevance to ESPRESSO. Whatever developments follow in later iterations, ESPRESSO's current goal is to accomplish the core IR task of providing relevant search results, and to do so with the lowest complexity and cost that is practicable at each experimental stage.

11 Conclusions & Future Work

This section is organized into two parts: we begin by discussing directions for future work, followed by the conclusions of the study.

11.1 Future Work

Our research will extend this work on privacy-preserving decentralized search to the following future directions:

- Exploring user-specific fine-tuned word embeddings in applying advanced neural networks to decentralized search, in order to augment privacy-preserving retrieval tasks with Large Language Models (LLMs).
- Conducting further analysis on this paper's finding of access-level thresholds where fine-tuning word embeddings leads to stronger results, and confirming that privacy-preserving fine-tuning for users with lower access risks overfitting and drift from pre-trained semantics.
- Extending our framework to work with real-world heterogeneous personal data in domains (such as Health and Well-being [20, 37]) that typically entail multi-modal datasets. This would allow us to evaluate ranking model robustness under varied data types and distributions.
- Extending our use case scenarios [37] to the usage of multi-modal datasets in practical implementations. This could be in the context of Human Digital Twins (HDTs) for healthcare, where our framework has the potential to address the problem of privacy-preserving data analysis and decision-making in network infrastructure implementations [8]. There is also a potential contribution to effective data privacy schemes for prevention of unauthorized access in generative AI-driven HDTs [9], as well as for protection of sensitive individual data over mobile networks when producing AI-generated content for HDTs [7].
- Benchmarking the performance of our source selection approach against more sophisticated Learning-to-Rank techniques, such as the LTRRS framework proposed by Wu et al. [47], which supports resource ranking by partitioning collections in cooperative environments and incorporates topic relevance, term statistics, query-independent features, and document sampling as ranking signals.
- Exploring the integration of advanced privacy-enhancing technologies—such as differential privacy, secure computation, along with utilizing homomorphic encryption techniques for large-scale personal data sharing systems [1], empowering those techniques on the level of P2P overlay networks such as GaianDB or Gnutella [25].
- Investigating the scalability of the framework in large, globally distributed POD networks; including optimization of overlay network routing, POD index compression and aggregations, and privacy-preserving caching strategies to reduce query latency and bandwidth usage.
- Last but not least, studying the performance implications of our proposed privacy-preserving ranking models on different decentralized systems, such as *Dataswyft* [7], so that we can evaluate their generalizability, identify system-specific challenges, and inform the design of adaptable retrieval strategies across diverse re-decentralized Web platforms.

[7] Dataswyft: https://dataswyft.com/

Just Accepted

11.2 Conclusions

Decentralized AI is emerging as a promising approach for enabling search across personal online datastores (PODs). However, it introduces challenges for information retrieval due to the distributed nature of data, the rareness of relevant terms across PODs, limited access rights, and the overhead of black-box models. To address these issues, we investigate efficient, privacy-preserving IR methods—such as keyword- and vector-based retrieval—tailored for decentralized personal repositories. Our work improves the quality of decentralized search and highlights the benefits of traditional IR techniques in this context, supported by experiments demonstrating strong retrieval performance. This direction is relevant to many domains where data is sensitive, distributed, and user-owned, such as personal finance and education records. We focus on health data as a realistic and impactful use case, where patients act as POD owners.

We extend both keyword-based and vector-based models for search in PODs and demonstrate that traditional IR methods achieve strong performance in top-k retrieval. The results are comparable to an ideal per-user centralized baseline, where all documents accessible to each user are processed in a single index (not a global, omniscient index), indicating the effectiveness of our approach in privacy-preserving settings. Additionally, fine-tuning embeddings with user-specific indexes yields promising results for search parties with higher access control (above 50%), while for search parties with lower access control (below 50%), a sparse vector-based approach provides more reliable results. This is because access above 50% appears to provide enough coverage of the collection to support meaningful semantic learning, while smaller indexes risk overfitting and shifting away from the pre-trained model.

Our paper offers a promising avenue for data scientists to advance these methods. By leveraging the strengths of probabilistic models and more sophisticated vector-based approaches, such as pre-trained IR models, within a decentralized AI framework, this work bridges the gap between centralized search methodologies and modern federated AI systems.

Acknowledgments

This work was funded by the UK EPSRC ESPRESSO grant (EP/W024659/1, EP/W024659/1).

References

[1] Divyakant Agrawal, Amr El Abbadi, and Shiyuan Wang. 2012. Secure and privacy-preserving data services in the cloud: A data centric view. *Proceedings of the VLDB Endowment* 5, 12 (2012), 2028–2029.

[2] Leif Azzopardi, Charles L.A. Clarke, Paul Kantor, Bhaskar Mitra, Johanne R. Trippas, Zhaochun Ren, Mohammad Aliannejaddi, Negar Arabzadeh, Raman Chandrasekar, Maarten De Rijke, Panagiotis Eustratiadis, William Hersh, Jin Huang, Evangelos Kanoulas, Jasmin Kareem, Yongkang Li, Simon Liupart, Kidist Amde Mekonnen, Adam Roegiest, Ian Soboroff, Fabrizio Silvestri, Suzan Verberne, David Vos, Eugene Yang, and Yuyue Zhau. 2024. Report on the Search Futures Workshop at ECIR 2024. In *ACM SIGIR Forum*. 1–41. Issue 1. doi:10.1145/3687273.3687288

[3] Graham Bent, Patrick Dantressangle, David Vyvyan, Abbe Mowshowitz, and Valia Mitsou. 2008. A dynamic distributed federated database. In *Proc. 2nd Ann. Conf. International Technology Alliance*.

[4] David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research* 3, Jan (2003), 993–1022.

[5] Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 1175–1191.

[6] Jamie Callan and Margaret Connell. 2001. Query-based sampling of text databases. *ACM Transactions on Information Systems (TOIS)* 19, 2 (2001), 97–130.

[7] Jiayuan Chen, Changyan Yi, Hongyang Du, Dusit Niyato, Jiawen Kang, Jun Cai, and Xuemin Shen. 2024. A revolution of personalized healthcare: Enabling human digital twin with mobile AIGC. *IEEE network* 38, 6 (2024), 234–242.

[8] Jiayuan Chen, Changyan Yi, Samuel D. Okegbile, Jun Cai, and Xuemin Shen. 2024. Networking architecture and key supporting technologies for human digital twin in personalized healthcare: a comprehensive survey. *IEEE Communications Surveys & Tutorials* 26, 1 (2024), 706–746. doi:10.1109/COMST.2023.3308717

Just Accepted

[9] Jiayuan Chen, Shi You, Yi Changyan, Hongyang Du, Jiawen Kang, and Dusit Niyato. 2024. Generative-AI-driven human digital twin in IoT healthcare: a comprehensive survey. *IEEE Internet of Things Journal* 11, 21 (2024), 34749—-34773. doi:10.1109/JIOT.2024.3421918

[10] Andrew M Dai and Quoc V Le. 2015. Semi-supervised sequence learning. *Advances in neural information processing systems* 28 (2015).

[11] Cynthia Dwork. 2006. Differential privacy. In *International colloquium on automata, languages, and programming*. Springer, 1–12.

[12] Alireza Fallah, Aryan Mokhtari, and Asuman Ozdaglar. 2020. Personalized federated learning: A meta-learning approach. *arXiv preprint arXiv:2002.07948* (2020).

[13] Adamu Garba, Shah Khalid, and Irfan Ullah. 2024. Understanding the impact of query expansion on federated search. *Multimedia Tools and Applications* 83, 4 (2024), 10393–10407.

[14] Adamu Garba, Shengli Wu, and Shah Khalid. 2023. Federated search techniques: an overview of the trends and state of the art. *Knowledge and Information Systems* 65, 12 (2023), 5065–5095.

[15] Adamu Garba, Shengli Wu, and Shah Khalid. 2023. Federated search techniques: an overview of the trends and state of the art. *Knowledge and Information Systems* 65, 12 (2023), 5065–5095.

[16] Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*. 169–178.

[17] Otis Gospodnetic, Erik Hatcher, and Michael McCandless. 2010. *Lucene in action*. Simon and Schuster.

[18] Luis Gravano, Chen-Chuan K Chang, Hector Garcia-Molina, and Andreas Paepcke. 1997. STARTS: Stanford proposal for Internet meta-searching. In *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*. 207–218.

[19] Donna Harman. 2011. *Information retrieval evaluation*. Morgan & Claypool Publishers.

[20] Konstantin Hemker, Nikola Simidjievski, and Mateja Jamnik. 2024. Healnet: Multimodal fusion for heterogeneous biomedical data. *Advances in Neural Information Processing Systems* 37 (2024), 64479–64498.

[21] William Hersh. 2024. Search still matters: information retrieval in the era of generative AI. *Journal of the American Medical Informatics Association* 31, 9 (2024), 2159–2161.

[22] Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146* (2018).

[23] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and trends® in machine learning* 14, 1–2 (2021), 1–210.

[24] Kwing Hei Li, Pedro Porto Buarque de Gusmão, Daniel J Beutel, and Nicholas D Lane. 2021. Secure aggregation for federated learning in flower. In *Proceedings of the 2nd ACM International Workshop on Distributed Machine Learning*. 8–14.

[25] Eng Keong Lua, Jon Crowcroft, Marcelo Pias, Ravi Sharma, and Steven Lim. 2005. A survey and comparison of peer-to-peer overlay network schemes. *IEEE Communications Surveys & Tutorials* 7, 2 (2005), 72–93.

[26] Mohamed Mahmoud, Khaled Rabieh, Ahmed Sherif, Enahoro Oriero, Muhammad Ismail, Erchin Serpedin, and Khalid Qaraqe. 2019. Privacy-preserving fine-grained data retrieval schemes for mobile social networks. *IEEE Trans. Dependable and Secure Comput.* (2019), 871–884. doi:10.1109/TDSC.2017.271416

[27] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.

[28] Christian Meurisch, Bekir Bayrak, and Max Mühlhäuser. 2020. Privacy-preserving AI services through data decentralization. In *Proceedings of The Web Conference 2020*. 190–200.

[29] Tomas Mikolov. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* 3781 (2013).

[30] Alan Mitchell. 2024. Beware! A new data bandwagon is rolling. (2024). https://medium.com/mydex/beware-a-new-data-bandwagon-is-rolling-5eebf8501a42

[31] IC Ng, Roger Maull, Glenn Parry, Jon Crowcroft, Kimberley Scharf, Tom Rodden, and Chris Speed. 2013. Making Value Creating Context Visible for New Economic and Business Models: Home Hub-of-all-Things (HAT) as Platform for Multi-Sided Market Powered by Internet-of-Things. In *Panel Session at The Future of Value Creation in Complex Service Systems Minitrack of Hawaii International Conference on Systems Science (HICSS), January*. 7–10.

[32] Alexandra Olteanu, Jean Garcia-Gathright, Maarten de Rijke, Michael D Ekstrand, Adam Roegiest, Aldo Lipani, Alex Beutel, Alexandra Olteanu, Ana Lucic, Ana-Andreea Stoica, et al. 2021. FACTS-IR: fairness, accountability, confidentiality, transparency, and safety in information retrieval. In *ACM SIGIR Forum*, Vol. 53. ACM New York, NY, USA, 20–43.

[33] Jay M Ponte and W Bruce Croft. 2017. A language modeling approach to information retrieval. In *ACM SIGIR Forum*, Vol. 51. ACM New York, NY, USA, 202–208.

Just Accepted

[34] Mohamed Ragab, Helen Oliver, Mohammad Bahrani, Alexandra Poulovassilis, Thanassis Tiropanis, Adriane Chapman, and George Roussos. 2025. Exploring the Performance of Scalable Keyword Search on Decentralized Data with Differential Visibility Constraints. (2025). Manuscript in revision for resubmission to VLDB.

[35] Mohamed Ragab, Yury Savateev, Reza Moosaei, Thanassis Tiropanis, Alexandra Poulovassilis, Adriane Chapman, and George Roussos. 2023. ESPRESSO: A Framework for Empowering Search on Decentralized Web. In *Web Information Systems Engineering - WISE 2023*, Vol. 14306. Springer, 360–375.

[36] Mohamed Ragab, Yury Savateev, Helen Oliver, Thanassis Tiropanis, Alexandra Poulovassilis, Adriane Chapman, and George Roussos. 2024. ESPRESSO: A Framework to Empower Search on the Decentralized Web. *Data Science and Engineering* 9, 4 (2024), 431–448.

[37] Mohamed Ragab, Yury Savateev, Helen Oliver, Thanassis Tiropanis, Alexandra Poulovassilis, Adriane Chapman, and George Roussos. 2024. Unlocking the potential of health data with decentralised search in personal health datastores. In *Companion Proceedings of the ACM Web Conference 2024*. 1154–1157.

[38] Mohamed Ragab, Yury Savateev, Helen Oliver, Thanassis Tiropanis, Alexandra Poulovassilis, Adriane Chapman, Ruben Taelman, and George Roussos. 2024. Decentralized Search over Personal Online Datastores: Architecture and Performance Evaluation. In *International Conference on Web Engineering*. Springer, 49–64.

[39] Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends® in Information Retrieval* 3, 4 (2009), 333–389.

[40] Thomas Roelleke. 2013. *Information retrieval models: Foundations and relationships*. Morgan & Claypool Publishers.

[41] Gerard Salton, Anita Wong, and Chung-Shu Yang. 1975. A vector space model for automatic indexing. *Commun. ACM* 18, 11 (1975), 613–620.

[42] Pierangela Samarati and Sabrina Capitani De Vimercati. 2000. Access control: Policies, models, and mechanisms. In *International school on foundations of security analysis and design*. Springer, 137–196.

[43] Andrei Sambra, Amy Guy, Sarven Capadisli, and Nicola Greco. 2016. Building decentralized applications for the social web. In *Proceedings of the 25th international conference companion on world wide web*. 1033–1034.

[44] Luo Si and Hui Yang. 2014. Pir 2014 the first international workshop on privacy-preserving ir: When information retrieval meets privacy and security. In *ACM SIGIR Forum*, Vol. 48. ACM New York, NY, USA, 83–88.

[45] Luo Si and Hui Yang. 2014. Privacy-preserving ir: When information retrieval meets privacy and security. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*. 1295–1295.

[46] Ian Soboroff. 2024. Privacy in Information Retrieval. In *Information Retrieval: Advanced Topics and Techniques*. 445–463.

[47] Tianfeng Wu, Xiaofeng Liu, and Shoubin Dong. 2019. LTRRS: A Learning to Rank Based Algorithm. In *Information Retrieval: 25th China Conference, CCIR 2019, Fuzhou, China, September 20–22, 2019, Proceedings*, Vol. 11772. Springer Nature, 52.

[48] Xiaolin Zheng, Zhongyu Wang, Chaochao Chen, Jiashu Qian, and Yao Yang. 2023. Decentralized graph neural network for privacy-preserving recommendation. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 3494–3504.

Received 15 April 2025; revised 16 September 2025; accepted 31 October 2025

Just Accepted