Supporting Vertical Web Search and Customized Search Applications with the Modular and Open Framework MOSAIC

Sebastian Gürtl[1,*], Alexander Nussbaumer[1,*] and Christian Gütl[1]

[1]Graz University of Technology, Rechbauerstraße 12, 8010 Graz, Austria

Abstract

Web search plays a crucial role in retrieving information, yet many existing search engines limit transparency and customization. Open-source frameworks provide alternatives but often require extensive configuration and lack direct support for openly available web indexes. This paper introduces MOSAIC, a modular open-source search framework designed for vertical web exploration. MOSAIC enables domain-specific search by integrating index partitions downloaded from the Open Web Index (OWI). Its modular architecture supports vertical search applications by allowing customization of query execution, filtering mechanisms, metadata management, and result representation. A development study and expert focus groups evaluated MOSAIC's applicability and modularity, identifying both strengths and areas for improvement. The results showed that the framework's modularity, OWI partition integration, and metadata management are key strengths. However, the study also revealed limitations in OWI partition updates, unified ranking, and documentation, which require further refinement. Furthermore, a science search application demonstrated its potential for enriching structured research repositories with web content. Future work will address improved OWI partition management, re-ranking mechanisms, and AI-assisted enhancements such as conversational search to refine usability and adaptability in vertical search applications.

Keywords

Web Search Engine, Modular Search Framework, Search Engine Customization, Open Web Index, Vertical Web Search, Information Retrieval

1. Introduction

In today's digital landscape, web search is essential for retrieving information across various domains, from academia to industry. However, the majority of widely used search engines are proprietary, which restricts transparency and customization. This lack of openness hinders the development of customized search solutions that meet specific requirements. In addition, artificial intelligence (AI)-assisted search and information retrieval systems have gained attention recently, as generative AI (GenAI) expands into web search engines and domain-specific retrieval tasks. While these advancements enable new interaction modes, concerns persist regarding the lack of transparency, the reliability of generated responses, and the difficulty of verifying sources [1]. This underscores the necessity for open-source frameworks that offer both flexibility and transparency.

Open-source search frameworks offer an alternative that focuses on these characteristics. Unlike proprietary search engines, which limit access to their index data, indexing, and ranking mechanisms, open-source solutions allow developers to customize and adapt search functionality to meet specific needs. Yet, an open-source search framework by itself is not sufficient—it also depends on comprehensive and reliable index data to function effectively [2, 3]. Without such index data, developers are often forced to rely on proprietary datasets or create their own indexes, which can be both costly and resource-intensive. A key resource in this space is the Open Web Index (OWI) [4, 5]. It offers openly accessible index data to support a wide range of search applications. By incorporating open index data such as the OWI, these frameworks enable developers to build tailored search solutions while minimizing

Second International Workshop on Open Web Search (WOWS 2025)

*Corresponding author.

sebastian.guertl@tugraz.at (S. Gürtl); alexander.nussbaumer@tugraz.at (A. Nussbaumer); c.guetl@tugraz.at (C. Gütl)
0009-0006-9008-7147 (S. Gürtl); 0000-0002-4692-5741 (A. Nussbaumer); 0000-0001-9589-1966 (C. Gütl)

reliance on proprietary search infrastructure. However, existing open-source search frameworks often require technical expertise for setup and configuration, and lack native integration with index partitions downloaded from the OWI [5].

To address these challenges, we introduce the open-source framework *Modular Search Application based on Index Fractions* (MOSAIC), designed to facilitate vertical web exploration. It provides a ready-to-use solution that can be deployed and used immediately via its API. The framework provides a configurable approach to natively incorporate OWI partitions, query processing, and result representation. In contrast to general-purpose search engines, MOSAIC focuses on domain-specific retrieval and allows search functionalities to be customized and extended with custom modules. Therefore, this paper examines its modular architecture and applicability in vertical web search exploration.

We outline the design and implementation of MOSAIC, explain its modular and extensible structure, and highlight advantages for vertical search applications. Additionally, we present findings from a development study and focus groups that illustrate MOSAIC's applicability from a developer's perspective, integration potential, and areas for improvement. Eventually, we examine future directions and open challenges of MOSAIC, focusing on integrating GenAI to improve query processing, result re-ranking, and interactive user interaction.

2. Related Work

The rise of search engines such as Google led to significant advancements in large-scale web indexing and retrieval, which enabled efficient access to vast amounts of online information [6, 7]. These search engines introduced centralized infrastructures capable of crawling, processing, and organizing unstructured web content. Before centralized search engines became dominant, distributed systems such as Harvest explored alternative architectures for indexing and retrieval and therefore offered an early approach to decentralized, modular and scalable search [8].

Building on these concepts, modern open-source search frameworks, such as Apache Solr[1], Terrier[2], OpenSearch[3], and Elasticsearch[4] have been widely adopted for a variety of indexing and search tasks. In general, these frameworks are flexible and extensible, which makes them suitable for a wide range of applications. However, they often require considerable configuration to meet domain-specific search needs [9, 10, 11]. Infret, another open-source information retrieval framework, features efficient retrieval and is particularly beneficial in educational and learning environments, but has limitations in scalability and adaptability to diverse domain-specific requirements [12, 13].

Additionally, open-source search frameworks commonly depend on proprietary or self-built indexes of web pages, which introduces challenges in acquiring and maintaining comprehensive web data. Although Common Crawl[5] provides a large-scale open web corpus, maintaining an index of its full collection is often less practical than crawling and indexing a smaller, domain-specific subset [5].

The OpenWebSearch.eu[6] project addresses this issue by providing the OWI, a structured and openly accessible web index, consisting of several partitions. Its construction follows a structured pipeline. First, OWLer, a Stormcrawler-based distributed web crawler, collects raw web data [14]. Next, the collected data is preprocessed and analyzed by the web content analysis pipeline Resilipipe, which is based on Resiliparse[7] [15]. This results in structured metadata of crawled web pages such as extracted content and language information [5]. The subsequent indexing process uses Apache Spark[8] and utilizes the information extracted from the preprocessing step. The resulting OWI partition consists of structured

[1] https://solr.apache.org
[2] https://github.com/terrier-org/terrier-core
[3] https://opensearch.org/
[4] https://elastic.co/elasticsearch
[5] https://commoncrawl.org/
[6] https://openwebsearch.eu/
[7] https://resiliparse.chatnoir.eu/
[8] https://spark.apache.org/

metadata and document representations stored in Apache Parquet[9] files, along with the inverted index in the Common Index File Format (CIFF). Regarding metadata, the columnar storage format of Parquet is well-suited for large-scale data processing [16]. The use of CIFF allows for the efficient exchange of index data, subsequently facilitating more accurate and fair comparisons between different search systems [17]. Existing search frameworks lack direct support for OWI partitions and therefore require additional effort to preprocess and integrate this data into search infrastructures. This limitation may complicate the development of open and adaptable search applications that leverage OWI partitions, particularly for those aiming to leverage web indexes without relying on proprietary solutions.

Nussbaumer et al. addressed this gap and demonstrated through a prototype application[10] that OWI partitions created by the OpenWebSearch.eu index generation pipeline can be used for searching and retrieving information [18]. However, this prototype application stored all OWI partition data in memory, thus problems occurred as soon as the OWI partitions became too large. Therefore, the primary motivation is to demonstrate the effective utilization of OWI partitions in search engines as they provide a technological foundation for custom applications while maintaining a modular and configurable framework.

In addition, recent advancements in large language models (LLMs) and GenAI have introduced new methods in search, such as retrieval-augmented generation (RAG). This approach combines traditional document retrieval with LLM-based response generation. Therefore, LLMs base their outputs on structured and verifiable data sources [19]. By combining an open and modular search architecture with AI-assisted retrieval and ranking, such a framework could help connect open search with GenAI while maintaining control over index data and search transparency.

3. The MOSAIC Framework

The web search engine framework MOSAIC provides a modular and configurable component-based architecture that overcomes the limitations of in-memory index data storage approaches. The framework follows the core idea of enabling domain-specific search by integrating modular components. It utilizes Apache Lucene[11] for indexing and query processing, and DuckDB[12] for metadata management. This enables both storage and retrieval of web pages contained in an OWI partition. The framework allows selective inclusion of one or multiple OWI partitions and provides direct access to the index data for customizable metadata filtering mechanisms. MOSAIC is an open-source software and available via an OpenWebSearch.eu project's GitLab repository[13].

3.1. Architecture and Functionalities

Figure 1 illustrates MOSAIC's sub-systems and components, and depicts the modular architecture of the framework, designed to facilitate vertical web search. The core application is implemented in Java, and available as a dockerized container via the container registry[14]. To run MOSAIC locally without additional efforts in setting up a web server environment, it includes the web server framework Quarkus[15]. The core application processes queries through Lucene and manages metadata with DuckDB. The *Lucene CIFF import* tool[16] enables the integration of an OWI partition's inverted index data into MOSAIC and removes the need for additional preprocessing. A REST API enables interaction between the core application and external applications such as a web interface, domain-specific search systems, and additional services.

[9] https://parquet.apache.org/
[10] https://opencode.it4i.eu/openwebsearcheu-public/prototype-search-application
[11] https://lucene.apache.org/
[12] https://duckdb.org/
[13] https://opencode.it4i.eu/openwebsearcheu-public/mosaic
[14] https://opencode.it4i.eu/openwebsearcheu-public/mosaic/container_registry
[15] https://quarkus.io/
[16] https://github.com/informagi/lucene-ciff

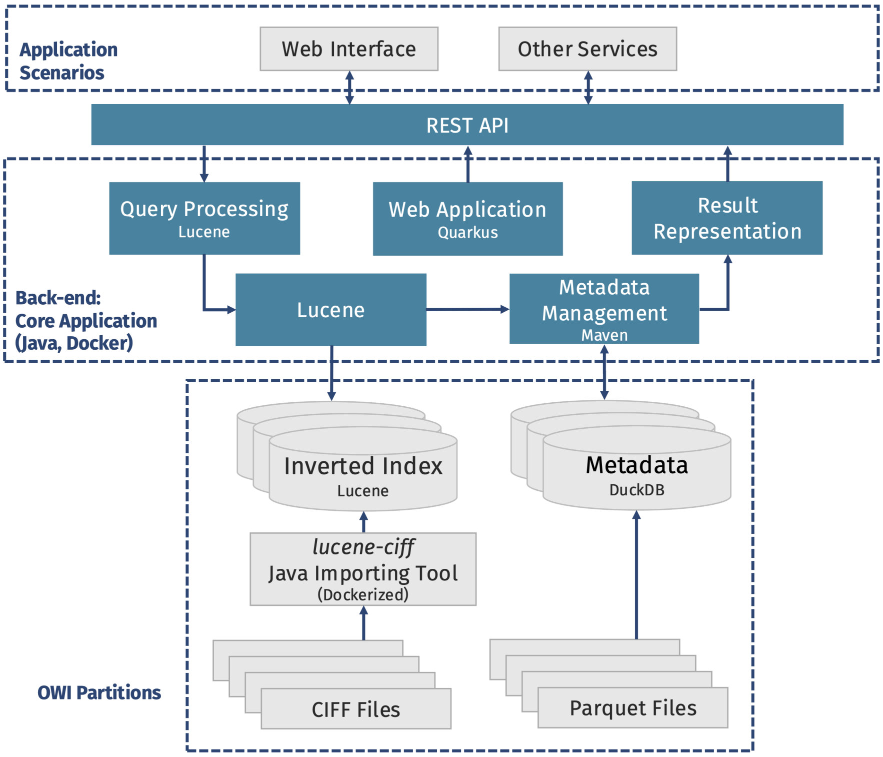


Figure 1: High-level system architecture of the MOSAIC framework. It integrates OWI partitions within its core application while interacting with various application scenarios through a REST API.

After importing OWI partitions, users can submit search queries through the web interface, REST API, or external services. Queries may be rewritten or analyzed using custom analyzers. Additionally, users can integrate LLMs to refine search terms. The Lucene *IndexSearcher* processes the queries using the BM25 ranking function [20]. Additionally, MOSAIC supports re-ranking after retrieval, with a demonstration implementation that adjusts ranking based on the word count of a web page's extracted main content.

MOSAIC refines search results by applying metadata filtering and enrichment. Users specify filtering parameters to refine results based on metadata attributes, while developers implement advanced filtering methods. The system enriches results by adding relevant metadata to provide comprehensive and contextually informative output. It also generates text snippets by extracting key sections from metadata. When necessary, an on-demand mechanism retrieves full extracted document text from Parquet files to improve snippet accuracy.

Additionally, MOSAIC provides a REST API for handling search queries and returning results in either JavaScript Object Notation (JSON) or Extensible Markup Language (XML) format. The `/search` endpoint processes queries across one or multiple OWI partitions and returns results in JSON format, while `/searchxml` follows the OpenSearch protocol[17] and formats responses as XML documents. If no specific OWI partition is defined, MOSAIC searches all available partitions and structures the response accordingly. Additionally, the `/index-info` endpoint provides metadata about the available partitions, including the number of documents and languages that occur in the partition respectively.

3.2. Metadata Modules

To utilize metadata in the search process, MOSAIC provides a structured metadata management system that enables customized filtering, enrichment, and retrieval. The framework follows a modular plugin architecture that allows flexible integration of metadata modules and optional system components. Developers can customize their MOSAIC-based vertical search engine by enabling or disabling specific

[17] https://github.com/dewitt/opensearch

modules without modifying the core system. It relies on Apache Maven[18] for dependency management, build processes, and module lifecycle handling. Metadata modules interact with the core system through defined interfaces and are configured via a dedicated settings file.

Metadata modules in MOSAIC improve search results by adding contextual information such as titles, URLs, language, extracted topics, and geolocation data. Advanced filtering operates through metadata fields in Parquet files, processed via DuckDB. For example, users can further limit search results using geographic coordinates. Each module defines metadata columns and filter parameters, which enable search refinements and enrichment. The framework provides an API for query validation, SQL filter clause generation, and result serialization in JSON and XML. Developers extend MOSAIC by defining custom metadata attributes in Parquet files, making them accessible for filtering, retrieval, and enrichment without requiring changes to the core framework.

4. Evaluation

Assessing the applicability, modularity and integration potential of MOSAIC involved a development study, focus groups, and a domain-specific use case. The development study provided insights into the developer experience and examined how effectively developers can configure and extend the framework. Focus groups with domain experts helped to identify strengths and areas for improvement, particularly regarding modularity and metadata integration. Eventually, a science search application illustrated MOSAIC's use in a specific domain and demonstrated its applicability in vertical search.

4.1. Development Study

The development study took place during a one-day hackathon, organized by the Cognitive and Digital Science (CoDiS) Lab at Graz University of Technology as part of the OpenWebSearch.eu initiative. The hackathon provided a structured, hands-on environment where participants explored MOSAIC. Participants engaged with MOSAIC and delved into its modular architecture, metadata management, and integration with OWI partitions. The study aimed to determine how effectively users could install, configure, and extend MOSAIC, while also identifying strengths and areas for improvement.

4.1.1. Participants

The study involved 13 participants, including 11 computer science students (84.62%) from Graz University of Technology (6 bachelor's students (46.15%) and 5 master's students (38.46%)) and 2 external participants (15.38%) with master's degrees in computer science. The gender distribution included 2 women (15.38%) and 11 men (84.62%). Participants were divided into four teams, with one team participating online. At the end of the hackathon, 11 participants (84.62%) completed a questionnaire designed to capture qualitative and quantitative feedback on MOSAIC's installation, applicability, and potential improvements.

4.1.2. Materials and Methods

The hackathon began with a presentation on web search fundamentals, the OpenWebSearch.eu project, and MOSAIC's architecture. Participants received access to the technical documentation, a developer guide, and example use cases to support development. Collaboration and communication was facilitated through cloud storage, a dedicated Discord server, and a Jitsi meeting room for online participants.

Throughout the hackathon, participants worked locally on their own notebooks. To streamline the development process, pre-prepared OWI partitions were provided. Teams explored predefined tasks, such as creating new OWI partitions, modifying the front-end, integrating MOSAIC as a service, and performing web data analysis.

[18] https://maven.apache.org/

At the end of the hackathon, participants completed a questionnaire designed to collect structured feedback on their experience with MOSAIC. The survey assessed applicability, modularity, and the integration process.

4.1.3. Procedure

Following the introductory session, participants formed teams and brainstormed potential applications of MOSAIC. Each team presented its initial concept in a plenary session and received feedback before proceeding to the development phase, which lasted approximately five hours. The organizers provided technical support and facilitated discussions to refine project implementations. The development study as part of the hackathon concluded with final project presentations, a voting session, and participant questionnaires, which assessed the applicability, modularity, and integration potential of the framework. Rating all aspects of MOSAIC asked in the questionnaire was not mandatory.

4.1.4. Results

The responses of the questionnaire offered information on the applicability of MOSAIC and the developer experience. As indicated in Figure 2, participants responded positively to the technical concept and modularity of MOSAIC, all participants agreeing that it supports the development of vertical search engines. However, installation and integration aspects received more mixed ratings, with 4 participants (36.36%) rating the ease of installation process as "Neutral" or lower. This suggests areas for improvement in applicability and installing MOSAIC.

These results align with qualitative feedback, highlighting the need for enhanced documentation and onboarding resources. The framework's API for search queries and result retrieval was noted as a strength, as it offers transparent and reliable means of accessing the index data. In addition, participants expressed interest in continuing the development with MOSAIC, particularly in the areas of AI-assisted information retrieval, search result ranking improvements, and expanded metadata support.

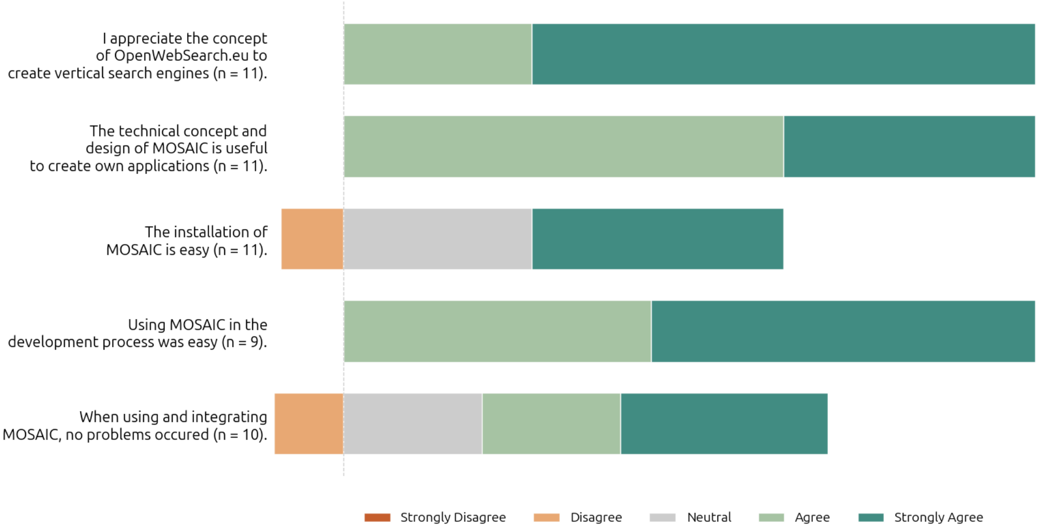


Figure 2: Development study's participant ratings for various aspects of MOSAIC ($n = 11$). Responses were collected using a five-point Likert scale, ranging from *Strongly Disagree* to *Strongly Agree*. The results show positive reception of MOSAIC's concept and applicability, while the installation process and integration experience received more varied feedback. The internal consistency of the questionnaire, measured using Cronbach's Alpha ($\alpha = 0.79$, 95% CI $[0.498, 0.935]$), indicates acceptable reliability [21].

4.2. Focus Groups

While the development study provided insights into MOSAIC's applicability from a developer's perspective, additional feedback from domain experts was gathered through focus group discussions. These sessions aimed to evaluate the technical design, modularity, and applicability of MOSAIC in real-world search scenarios and complement the findings from the development study.

4.2.1. Participants

The focus group discussions involved 10 experts, divided into 5 groups, from the OpenWebSearch.eu consortium. The participants represented institutions such as Radboud University, the University of Passau, the German Aerospace Center (DLR), CERN, and the Webis research group. Among the experts, 7 (70%) held a master's degree, and 3 (30%) had a doctoral degree. Overall, the groups included four PhD students (40%), four researchers (40%), and three professors (30%), and therefore provided a diverse range of expertise. The gender distribution consisted of 3 women (30%) and 7 men (70%).

4.2.2. Materials and Methods

The focus groups were conducted online using BigBlueButton, where participants selected a 30-minute session to join. Discussions were structured into three categories: i) the technical approach, ii) the modular architecture, and iii) the applicability of MOSAIC. The moderators asked predefined questions to guide the discussion and cover aspects such as system applicability, the effectiveness of the modular design, and potential applications. At the end of each session, participants completed an online questionnaire to provide quantitative feedback on the framework. To analyze qualitative feedback from the focus groups, thematic coding was applied to identify recurring patterns and categorize responses into strengths, weaknesses, and areas for improvement.

4.2.3. Procedure

Before the sessions, participants received an email invitation outlining the study's objectives. During the discussions, moderators ensured a structured dialogue and collected qualitative insights while taking notes. The sessions were audio-recorded with participant consent and deleted after analysis. Following the discussion, participants rated various aspects of MOSAIC, such as the system's technical design, modularity, and its suitability for custom search applications.

4.2.4. Results

Experts provided insights into MOSAIC's strengths, weaknesses, and areas for improvement, as summarized in Table 1. The focus groups identified MOSAIC as an effective domain-specific search engine with satisfactory performance and speed. Its modular architecture allows developers to customize components as needed. Participants valued its seamless integration with OWI partitions and its compatibility with the OpenWebSearch.eu project. The framework also supports prototyping and low-code development, which enables a user-friendly installation process and potential for community-driven creation and sharing of modules.

Despite the strengths, certain weaknesses became evident during the discussions. Participants noted that MOSAIC does not yet support dynamic updates, additions, or removals of OWI partitions, which limits adaptability in evolving search applications. Although the framework provides a strong foundation, it remains more of a demonstrator than a fully production-ready system. Concerns were also raised about the increasing complexity as more features are introduced. The ranking process currently operates on a per-partition basis rather than providing a unified ranking across partitions. Additionally, reliance on CIFF was seen as a constraint, as the format is not widely recognized in search engine development.

To address these issues, participants proposed several improvements. Enhancing support for multiple metadata versions and enabling CIFF file combinations could increase adaptability. Simplifying module

Table 1
Summary of strengths, weaknesses, and areas for improvement based on expert feedback from the focus groups. Participants highlighted MOSAIC's modularity and integration capabilities as key strengths, while also identifying limitations in OWI partition management, ranking as well as re-ranking, and CIFF adoption, along with areas for enhanced documentation and advanced search functionalities.

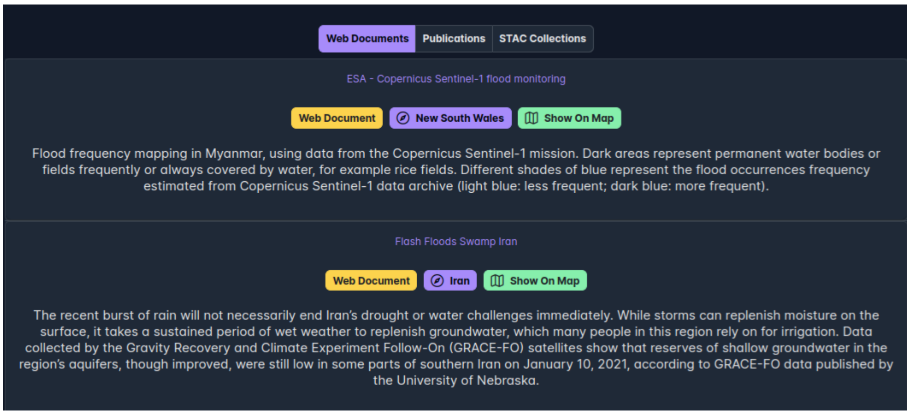
Category	Observations
Strengths	• Effective as a basic search engine, performing as expected with satisfactory performance and speed. • Modular architecture allows flexible plugin functionality and enables developers to add, remove, or customize components. • Easy integration of OWI partitions and compatibility with the OpenWebSearch.eu project. • Well-suited for prototyping and low-code development, which simplifies the creation of custom search engines. • Includes a user-friendly installation process and has the potential for a community-driven exchange mechanism of modules.
Weaknesses	• Lacks support for OWI partition updates, additions, and removals. • More suited as a demonstrator than a production system, with a risk of increasing complexity. • Ranks results separately for each partition rather than using a unified ranking. • Relies on CIFF, which is not widely adopted as an index exchange standard. • The web user interface lacks advanced search result presentation and requires improvements to better match user expectations. • Better suited for specific use cases based on the OWI schema (CIFF and Parquet) rather than broader applications.
Room for Improvement	• Need to combine CIFF files and handle different metadata versions. • Improve accessibility of modules by enabling easy addition via APIs. • Expand advanced search functionalities with additional filters and improved ranking across multiple OWI partitions. • Introduce automated module and OWI partition management. • Provide blueprints and detailed documentation to assist in developing and integrating custom modules.

integration through APIs and expanding search functionalities, including additional filters and a unified ranking mechanism, would improve usability. Furthermore, automating module and OWI partition management could reduce maintenance efforts. Eventually, providing more comprehensive documentation and structured development blueprints would make the framework more accessible to new users.

4.3. Science Search Application

Beyond evaluating MOSAIC's modularity and applicability, a science search application examined its role in scientific information retrieval. While traditional academic search engines provide structured datasets, they often lack real-time context on emerging research topics. To address this, MOSAIC was integrated into a hybrid search system that combined structured scientific metadata with web-based content from OWI partitions [22].

The application focused on earth observation and environmental research by combining structured datasets from DLR with relevant web content from a custom OWI partition of over 500 natural disaster-related web pages. A federated search interface enabled simultaneous querying of both sources, ranking results by semantic relevance and geographic attributes. Metadata modules in MOSAIC facilitated cross-source alignment and enabled filtering and refinement of search results. Figure 3 illustrates the



ESA - Copernicus Sentinel-1 flood monitoring

Flood frequency mapping in Myanmar, using data from the Copernicus Sentinel-1 mission. Dark areas represent permanent water bodies or fields frequently or always covered by water, for example rice fields. Different shades of blue represent the flood occurrences frequency estimated from Copernicus Sentinel-1 data archive (light blue: less frequent; dark blue: more frequent).

Flash Floods Swamp Iran

The recent burst of rain will not necessarily end Iran's drought or water challenges immediately. While storms can replenish moisture on the surface, it takes a sustained period of wet weather to replenish groundwater, which many people in this region rely on for irrigation. Data collected by the Gravity Recovery and Climate Experiment Follow-On (GRACE-FO) satellites show that reserves of shallow groundwater in the region's aquifers, though improved, were still low in some parts of southern Iran on January 10, 2021, according to GRACE-FO data published by the University of Nebraska.

Figure 3: Web pages retrieved from MOSAIC and displayed in the science search application's dashboard. The interface allows users to filter results, view metadata, and explore relevant documents [22].

application's dashboard with keyword-based search, metadata filtering, and a geographical map view to explore environmental events concerning scientific publications.

MOSAIC integrated structured scientific metadata with web content and therefore enabled hybrid retrieval. The system connected datasets from digital libraries with web documents on environmental events. Metadata alignment across different sources remained a challenge, particularly in standardizing metadata between MOSAIC and scientific repositories. Future improvements could explore metadata harmonization and automate extraction methods [22].

4.4. Limitations

While the evaluation provided valuable insights into MOSAIC's modularity and applicability, certain limitations should be considered. The development study included 13 participants, primarily computer science students from a single institution, which limits the generalizability of findings. Although their technical background allowed for a focused assessment of MOSAIC's framework, the study did not capture perspectives from non-technical users or professionals from diverse fields who might engage with MOSAIC differently.

Similarly, the focus groups consisted of ten experts, all affiliated with the OpenWebSearch.eu project. While this ensured informed discussions, it also meant that participants were already familiar with MOSAIC and its objectives. As a result, feedback may have been influenced by prior knowledge and project alignment, rather than reflecting perspectives from independent stakeholders or search engine operators who have not interacted with the framework before.

Another constraint was the availability of OWI partitions during the evaluation. Participants worked with pre-prepared partitions, which restricted their ability to assess dynamic partition updates, the integration of new metadata attributes, or real-time indexing scenarios. Although the study confirmed that MOSAIC can integrate OWI partitions modularly, future evaluations should examine how well the framework handles continuously evolving datasets and broader indexing configurations.

5. Future Directions and Open Challenges

Building on the findings from the development study, focus groups, and science search application, this section outlines directions for future development, explores potential applications leveraging GenAI, and discusses remaining challenges in the framework.

5.1. Technical Improvements

MOSAIC integrates OWI partitions, although several areas could benefit further refinement to enhance its flexibility and performance. One key aspect is improving OWI partition management, particularly in supporting dynamic updates, additions, and removals. While the current system enables OWI partition imports natively, it lacks mechanisms for updating index data incrementally. Introducing automated synchronization methods or incremental indexing could address this limitation without requiring full re-indexing [23]. A strategy to implement improved index data management could be the integration of the OWI partition management tool *owilix*[19].

Another important focus is on refining the modular architecture to improve ease of integration and extensibility. The framework's modularity allows developers to add or remove components, but managing dependencies between modules can result in challenges. MOSAIC could benefit from simplified module configuration, improved API accessibility, and blueprints for metadata modules. These improvements would lower the barrier for adoption and facilitate the creation of own modules.

Furthermore, clear and comprehensive documentation is essential for ensuring MOSAIC's accessibility to a wider audience. Therefore, enhancing the developer guide with detailed examples, step-by-step tutorials, live and pre-recorded trainings, and best practices for module integration would support developers in setting up and extending the framework.

5.2. Future Applications and AI Opportunities

Apart from technical improvements, MOSAIC's current architecture offers a basis for AI-assisted search applications, with potential enhancements in querying, ranking and re-ranking, retrieval, and result representation. One promising direction is RAG, where LLMs retrieve relevant documents from MOSAIC before generating responses. This approach could improve the search and retrieval process by helping generative models ground their responses in transparent and structured index data, such as OWI partitions [19]. Eventually, this could lead to multi-turn and interactive conversations with users to refine and clarify their search intents [24].

Beyond retrieval augmentation, AI could further improve query understanding, ranking, and result summarization. Integrating LLM-based query rewriting could refine user queries and therefore increase retrieval accuracy, as well as satisfying the user's needs [25]. Similarly, AI-assisted re-ranking could leverage semantic similarity models to improve search result relevance [26]. MOSAIC could also serve as a foundation for automated summarization tools to extract key insights from web pages.

Another possible future direction for MOSAIC is mobile-based local search, where OWI web content is accessed entirely offline on a mobile device. This approach allows retrieval without an internet connection and benefits environments with limited connectivity or strict privacy requirements.

Following a low-code approach, a customization tool such as MOSAIC2go[20] could help simplify the creation of domain-specific vertical search applications by providing an interactive interface for configuring MOSAIC-based search engines. This approach would reduce the need for manual setup and therefore allow non-technical users to develop and manage their own search solutions. Additionally, integrating AI-driven features, such as query expansion, RAG, or summarization, into such a customization platform would further extend its capabilities and could bridge the gap between technical and non-technical MOSAIC users.

6. Conclusion

This paper presented MOSAIC, an open-source web search framework designed for modular, vertical search applications that integrate OWI partitions to improve search accessibility. MOSAIC's component-based architecture enables customizable search solutions without relying on proprietary infrastructures. Moreover, the framework's plugin-based modularity allows developers to configure and extend its functionality, which makes it adaptable to various domain-specific search scenarios.

The evaluation through a development study, expert focus groups, and a science search application provided insights into MOSAIC's applicability, modularity, and integration potential. Participants

[19] https://opencode.it4i.eu/openwebsearcheu-public/owi-cli

[20] https://mosaic.ows.eu/mosaic2go/

identified the modular design, metadata management, and ease of OWI partition usage as key strengths. However, challenges remain in OWI partition updates, unified ranking, and metadata alignment, which requires further refinement. The science search application demonstrated MOSAIC's potential to enrich structured scientific search with web-based contextual information. The evaluation faced limitations in sample size, participant familiarity with MOSAIC, and the restricted number of OWI partitions. Broader testing with diverse user groups and datasets would help address these constraints.

In contrast to other modular search systems, MOSAIC integrates directly with OWI partitions, which simplifies working with large-scale open web indexes. While general-purpose search frameworks often require extensive configuration or rely on proprietary index data, MOSAIC offers a transparent and customizable approach to search infrastructure.

Future work will focus on improvements in OWI partition management, modularity enhancements, and ranking refinements. Additionally, MOSAIC's open and extensible design may enable potential applications in AI-assisted retrieval, including query rewriting, RAG features, re-ranking and conversational search. While further improvements are necessary, its modular architecture provides a flexible framework that could support advancements in open search and GenAI for vertical search applications.

Acknowledgements

This work has received funding from the European Union's Horizon Europe research and innovation program under grant agreement No 101070014 (OpenWebSearch.EU, https://doi.org/10.3030/101070014).

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly in order to: Grammar and spelling check. After using these tool, the authors reviewed and edited the content as needed and take(s) full responsibility for the publication's content.

References

[1] W. Hersh, Search still matters: information retrieval in the era of generative ai, Journal of the American Medical Informatics Association 31 (2024) 2159–2161. doi:10.1093/jamia/ocae014.

[2] C. D. Manning, P. Raghavan, H. Schütze, Introduction to Information Retrieval, Cambridge University Press, 2008. doi:10.1017/CBO9780511809071.

[3] W. B. Croft, D. Metzler, T. Strohman, Search engines: Information retrieval in practice, volume 520, Addison-Wesley Reading, 2010.

[4] M. Granitzer, S. Voigt, N. A. Fathima, M. Golasowski, C. Guetl, T. Hecking, G. Hendriksen, D. Hiemstra, J. Martinovič, J. Mitrović, et al., Impact and development of an open web index for open web search, Journal of the Association for Information Science and Technology 75 (2024) 512–520. doi:10.1002/asi.24818.

[5] G. Hendriksen, M. Dinzinger, S. M. Farzana, N. A. Fathima, M. Fröbe, S. Schmidt, S. Zerhoudi, M. Granitzer, M. Hagen, D. Hiemstra, et al., The open web index: Crawling and indexing the web for public use, in: European Conference on Information Retrieval, Springer, 2024, pp. 130–143. doi:10.1007/978-3-031-56069-9_10.

[6] S. Brin, L. Page, The anatomy of a large-scale hypertextual web search engine, Computer networks and ISDN systems 30 (1998) 107–117. doi:10.1016/S0169-7552(98)00110-X.

[7] L. Page, S. Brin, R. Motwani, T. Winograd, et al., The pagerank citation ranking: Bringing order to the web (1999).

[8] C. M. Bowman, P. B. Danzig, D. R. Hardy, U. Manber, M. F. Schwartz, The harvest information discovery and access system, Computer networks and ISDN Systems 28 (1995) 119–125. doi:10.1016/0169-7552(95)00098-5.

[9] M. A. Akca, T. Aydoğan, M. İlkuçar, An analysis on the comparison of the performance and configuration features of big data tools solr and elasticsearch, International Journal of Intelligent Systems and Applications in Engineering 4 (2016) 8–12. doi:10.3390/app112411590.

[10] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, D. Johnson, Terrier information retrieval platform, in: Advances in Information Retrieval: 27th European Conference on IR Research, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005. Proceedings 27, Springer, 2005, pp. 517–519. doi:10.1007/978-3-540-31865-1_37.

[11] S. Papadopoulos, P. Saiz, U. Schwickerath, E. Kleszcz, Architecting the opensearch service at cern, in: EPJ Web of Conferences, volume 295, EDP Sciences, 2024, p. 07006. doi:10.1051/epjconf/202429507006.

[12] A. Bobić, C. Cheong, J. Filippou, F. Cheong, C. Guetl, Infret: Enhancing a tool for explorative learning of information retrieval concepts, in: The Impact of the 4th Industrial Revolution on Engineering Education: Proceedings of the 22nd International Conference on Interactive Collaborative Learning (ICL2019)–Volume 1 22, Springer, 2020, pp. 67–78. doi:10.1007/978-3-030-40274-7_7.

[13] A. Bobić, C. Gütl, C. Cheong, Infret: preliminary findings of a tool for explorative learning of information retrieval concepts, in: Cross Reality and Data Science in Engineering: Proceedings of the 17th International Conference on Remote Engineering and Virtual Instrumentation 17, Springer, 2021, pp. 849–865. doi:10.1007/978-3-030-52575-0_70.

[14] M. Dinzinger, S. Zerhoudi, M. Al-Maamari, M. Istaiti, J. Mitrović, M. Granitzer, Owler: Preliminary results for building a collaborative open web crawler (2024). doi:10.5281/zenodo.10581840.

[15] S. Heineking, I. Zelch, G. Hendriksen, Resilipipe, 2024. URL: https://doi.org/10.5281/zenodo.13784624. doi:10.5281/zenodo.13784624.

[16] D. Vohra, D. Vohra, Apache parquet, Practical Hadoop Ecosystem: A Definitive Guide to Hadoop-Related Frameworks and Tools (2016) 325–335. doi:10.1007/978-1-4842-2199-0.

[17] J. Lin, J. Mackenzie, C. Kamphuis, C. Macdonald, A. Mallia, M. Siedlaczek, A. Trotman, A. de Vries, Supporting interoperability between open-source search engines with the common index file format, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 2149–2152. doi:10.1145/3397271.3401404.

[18] A. Nussbaumer, R. Kaushik, G. Hendriksen, S. Gürtl, C. Gütl, Conceptual design and implementation of a prototype search application using the open web search index, in: 5th International Open Search Symposium, Sl: Open Search Foundation, 2023. doi:10.5281/zenodo.10636166.

[19] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel, et al., Retrieval-augmented generation for knowledge-intensive nlp tasks, Advances in Neural Information Processing Systems 33 (2020) 9459–9474.

[20] S. Robertson, H. Zaragoza, et al., The probabilistic relevance framework: Bm25 and beyond, Foundations and Trends® in Information Retrieval 3 (2009) 333–389. doi:10.1561/1500000019.

[21] L. J. Cronbach, Coefficient alpha and the internal structure of tests, psychometrika 16 (1951) 297–334. doi:10.1007/BF02310555.

[22] A. Nussbaumer, S. Gürtl, J. Honeder, T. Hecking, C. Gütl, Enriching science search with the open search framework mosaic, in: 6th International Open Search Symposium, 2024, pp. 49–52. doi:10.5281/zenodo.13871624.

[23] W. Xia, H. Jiang, D. Feng, F. Douglis, P. Shilane, Y. Hua, M. Fu, Y. Zhang, Y. Zhou, A comprehensive study of the past, present, and future of data deduplication, Proceedings of the IEEE 104 (2016) 1681–1710. doi:10.1109/JPROC.2016.2571298.

[24] J. Gao, C. Xiong, P. Bennett, N. Craswell, Neural approaches to conversational information retrieval, volume 44, Springer Nature, 2023. doi:10.1007/978-3-031-23080-6.

[25] Y. Zhu, H. Yuan, S. Wang, J. Liu, W. Liu, C. Deng, H. Chen, Z. Liu, Z. Dou, J.-R. Wen, Large language models for information retrieval: A survey, arXiv preprint arXiv:2308.07107 (2023).

[26] S. Xu, L. Pang, J. Xu, H. Shen, X. Cheng, List-aware reranking-truncation joint model for search and retrieval-augmented generation, in: Proceedings of the ACM on Web Conference 2024, 2024, pp. 1330–1340. doi:10.1145/3589334.3645336.