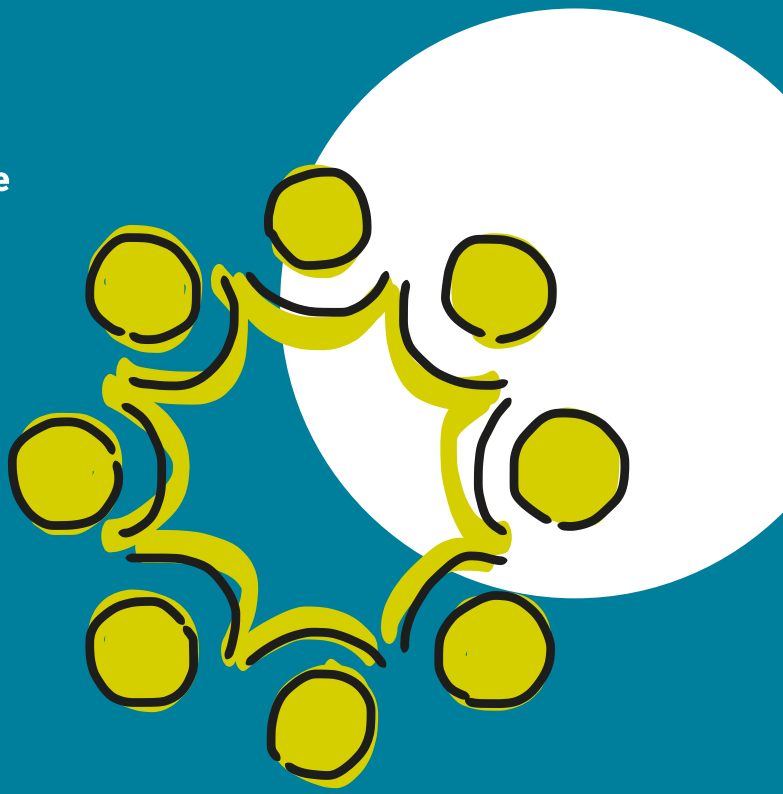
**Editors**
Andreas Wagner
Michael Granitzer
Christian Guetl
Per Öster
Christine Plote
Stefan Voigt

# Proceedings

# 6th International Open Search Symposium

#ossym2024

**9-11 October 2024**
**LRZ - Leibniz Supercomputing Centre**
**Garching, Germany**



ISSN: 2957-4935
ISBN: 978-92-9083-669-8

## Impressum

Editors:

Michael Granitzer, University Passau, Germany
Christian Güetl , Graz University of Technology, Austria
Per Öster, CSC – IT Center for Science, Finland
Christine Plote, Open Search Foundation, Germany
Stefan Voigt, Open Search Foundation, Germany
Andreas Wagner, CERN, Geneva, Switzerland

ISSN: 2957-4935
ISBN: 978-92-9083-669-8
DOI: 10.5281/zenodo.13887343



This report should be cited as:

Proceedings of 6th International Open Search Symposium #ossym2024, Leibniz Supercomputing Centre LRZ, Munich, Germany and Online, 9-11 October 2024, M. Granitzer, C. Gütl, P. Öster, C. Plote, S. Voigt, A. Wagner (eds). https://doi.org/10.5281/zenodo.13887343

## More Information

- Conference Website on CERN Indico
  https://indico.cern.ch/e/OSSYM-2024
- Open Search Community at Zenodo
  https://zenodo.org/communities/opensearch
- Event information at the Open Search Foundation
  https://opensearchfoundation.org/en/events-osf/ossym24

# Foreword

Dear readers,

in the continuation of our #ossym conference series, it is our great pleasure to present the pro-ceedings of the 6th International Open Search Symposium, #ossym24, which takes place from 9 to 11 October 2024 in Garching – Munich, Germany, hosted by the Leibniz Supercomputing Centre (LRZ).

In this year's conference we have 22 accepted papers from 68 authors. The increasing interest in open search and artificial intelligence is reflected in the recent contributions including topics such as "Crawling and Infrastructure", "Preprocessing and ML for Search", "Search Applications and Technologies", "Large Language Models, Retrieval-Augmented Generation and Named Entity Recognition", and "Economics, Ethics and Society".

All in all, the #ossym conference addresses a variety of formats, from scientific presentations, to interactive workshops on horizontal aspects of open search topics, to a panel discussion with industry players and policy makers. It provides also a platform for researchers from two EU projects "Open Web Search" and "NGI Search" to present their results and share knowledge.

Not covered in these proceedings, but nevertheless worth mentioning are the keynote speeches, providing valuable insights into technical, governmental, community-related and ethical aspects:

- Roberto Viola (Director General, Communications Networks, Content and Technology, European Commission) – Opening Keynote
- Prof. Dr. Martin Andree (Researcher at the University of Cologne, and bestselling Author "Big Tech Must Go") – "How we are taking back the net"
- Dr. Richard Socher (CEO of You.com) – You.com
- Nina Leseberg (Head of Communities & Engagement, Wikimedia Deutschland) – "Digital Discourse: how the Wikipedia Community safeguards the quality of the digital encyclopedia"

We want to express our special thanks to all authors for their sound contributions, to the programme committee for their valuable reviews and recommendations, to all keynote and featured speakers for their valuable insights, to all sponsors for their financial support, as well to the local team for their organisational efforts. Without all this great input and helpful support, it would not be possible to successfully run the #ossym conference series.

Our initial motivation was, and still is, the belief that the #ossym conference is an exemplary demonstration of how multifaceted the vibrant Open Web Search community approaches the topic and explores it from a wide range of disciplines and angles. Every #ossym conference brings the Open Web Search Initiative and related disciplines a big step forward year after year.

In this spirit: We are very happy to announce and look forward to the next year's conference – the #ossym25, taking place from 8 - 10 October 2025 in Helsinki hosted by CSC - IT Center for Science!

On behalf of #ossym24

*Andreas Wagner, Michael Granitzer, Christian Gütl, Christine Plote, Stefan Voigt and Per Öster*
Conference Chairs

# Symposium Organisation

## Programme Committee

Prof. Emmanuel Cartier, European Commission JRC
Prof. Dr. Alexander Decker, Technische Hochschule Ingolstadt, Germany
Prof. Dr. Kai Erenli, University of Applied Sciences BFI Vienna, Austria
Msc. Maik Fröbe, Friedrich-Schiller-Universität, Jena, Germany
Priv.-Doz Dr. Christian Geminn, University Kassel, Germany
Prof. Dr. Michael Granitzer, University Passau, Germany
Prof. Dr. Christian Gütl, Graz University of Technology, Austria
Prof. Dr. Andreas Henrich, University Bamberg, Germany
Prof. Djoerd Hiemstra, University of Twente, Radboud University, Netherlands
Dr. Xuke Hu, German Aerospace Centre, Jena, Germany
Dr. Igor Jakovljevic, CERN, Geneva, Switzerland
Prof. Dr. Robert Jäschke, Humboldt University Berlin & L3S, Hannover, Germany
Prof. Dr.-Ing. Nils Jensen, Ostfalia University of Applied Science, Wolfenbüttel, Germany
Prof. Dr. Mohammed Kaicer, Faculty of Sciences Kenitra, Morocco
Dr. Jens Kersten, German Aerospace Centre, Jena, Germany
Dr. Jelena Mitrovic, University of Passau, Germany
Prof. Dr. Engelbert Niehaus, University Koblenz-Landau, Landau, Germany
Dr. Jakub Piskorski, European Commission
Prof. Dr. Melanie Platz, Saarland University, Germany
Prof. Dr. Martin Potthast, Leipzig University, Germany
Prof. Dr. Mirko Presser, Aarhus University, Denmark
Prof. Dr. Georg Rehm, German Research Center for Artificial Intelligence (DFKI), Berlin, Germany Assoc.
Prof. Dr. Gianmaria Silvello, University of Padova, Italy
Dr. Tim Smith, CERN, Geneva, Switzerland
Prof. Dr. Arjen P. de Vries, Radboud University, Netherlands
Dr. Stefan Voigt, Open Search Foundation, Germany
Dr. Andreas Wagner, CERN, Geneva, Switzerland

## Conference Chairs

Prof. Dr. Michael Granitzer, University Passau, Germany
Prof. Dr. Christian Gütl, Graz University of Technology, Austria
Dr. Per Öster, CSC – IT Center for Science, Finland
Christine Plote, Open Search Foundation, Germany
Dr. Stefan Voigt, Open Search Foundation, Germany
Dr. Andreas Wagner, CERN, Geneva, Switzerland

## Local Organization

### LRZ – Leibniz Supercomputing Centre

# Contents

## Research Track Information

CIN - Crawling and Infrastructure
ETS - Ethics & Society
LRN - LLMs, RAG and NER
PML - Preprocessing and ML for Search STE - Search Applications and Technologies YIO - Young innovators for Open Search

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13872517

# ARCHITECTING THE OPENSEARCH SERVICE AT CERN FOR OPENWEBSEARCH.EU*

Noor A. Fathima[†1], M. Dinzinger[2], M. Granitzer[2], A. Wagner[1]

[1] CERN, Geneva, Switzerland
[2] University of Passau, Passau, Germany

*Abstract*

The Open Web Search.eu(OWS) project [1], aims to harness the untapped potential of the Web as a data source and make it accessible by developing a publicly funded, scalable and open European *Web search and analysis infrastructure* [2]. This research initiative is a joint venture that involves 14 research institutions throughout Europe [3], with CERN serving as one of the key technical infrastructure partners. The mission includes a robust, highly available continuous crawling operation, adhering to fundamental principles of openness and legal compliance. To this end, the OWS team has created the Open Web Crawler (OWLer) [4], an open-source software framework that facilitates the extensive collection of Web documents [5].

Amongst other, the OWLer comprises the *URL Frontier Service* which has the data structure containing all URLs discovered during the crawling process, effectively acting as the project's brain. It uses OpenSearch [6] technology as its primary back-end chosen for its robust persistence, distribution, and replication capabilities along with its indexing and querying capabilities, many of its open source plugins, and to leverage an existing OpenSearch-based implementation [7] of the open source URLFrontier framework [8] – which has been funded by NGI Zero [9].

The OpenSearch service is continuously operational at the CERN data center, and this article details along with our experience, it's technical setup and evolution throughout different phases of the project, including the design, implementation, improvements made and the challenges we overcame. This has culminated in us effectively parsing, indexing, and storing ~7TiB of data at the time of writing this article. We also discuss the future roadmap before concluding the paper.

## INTRODUCTION

The computational, data storage, and transfer tasks for the OWS project are executed within a distributed, heterogeneous infrastructure that spans five partner data center sites in Europe, including CERN. These data centers collectively manage a federated data infrastructure that leverages the confluence of HPC and cloud infrastructure [10], ensuring timely data availability for each step of the crawling, preprocessing, and indexing pipeline to collaboratively build a comprehensive web index. [11]. Given the capabilities of the computing and storage resources procured for the OWS project at CERN and their alignment with project requirements, CERN was strategically chosen to establish the following services:

- *URL Frontier Service*: This is the core of the OWLer Web crawling system, comprising the data processing back-end and the Java front-end service applications.
- *Logs Aggregation and Monitoring Service*: Encompasses the data processing back-end and stream processing using Apache Flink [12] as illustrated in Figure 1, chosen for its robust real-time data processing capabilities and scalability.
- *Federated Data Storage and Transfer Management Service*: Manages data storage and transfer across distributed infrastructure [13] using iRODS [14] and MinIO [15] as illustrated in Figure 1.

OpenSearch serves as the data processing back-end for the first two services listed. In this paper, we will concentrate on the OpenSearch service, specifically in context to the URL Frontier Service, as the use case, since the same principles are applicable to the second service.

Data visualization and analysis, observability, and monitoring are implemented for the first two listed services using appropriate tools such as OpenSearch Dashboards [16], Prometheus [17], and Grafana [18].

From the initial phase of the project, at CERN we implemented a streamlined infrastructure set-up to achieve the milestone of developing a pilot version as outlined in the referenced deliverable [19]. This setup involved the manual deployment of vertically scaled and distributed OpenSearch clusters on bare metal machines whose specifications are detailed in a further section. This initial configuration consisted of two production clusters indexing approximately ~5 TiB of URL data totaling approximately 9.8 billion documents and 2.3 TiB of logs and metrics data as of March 2024, respectively, and has one test cluster.

As the main back-end for search, data ingestion, and log and metric aggregation in the OWLer system, this set-up initially fulfilled our requirements. However, as other components of OWLer expanded, it became clear that our infrastructure could not scale efficiently with the limited manpower available for interventions. This realization led to the investigation and evaluation of alternative methods for scalable and more manageable service and server management and maintenance approaches to support the growing needs and future expansion of the OWLer system.

The OpenSearch As-A-Service infrastructure at CERN IT, which provides a managed OpenSearch service for a wide

* This project is funded by the European Commission under the grant agreement GA 101070014 within the Horizon Europe Framework Program.
† noor.afshan.fathima@cern.ch

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13872517



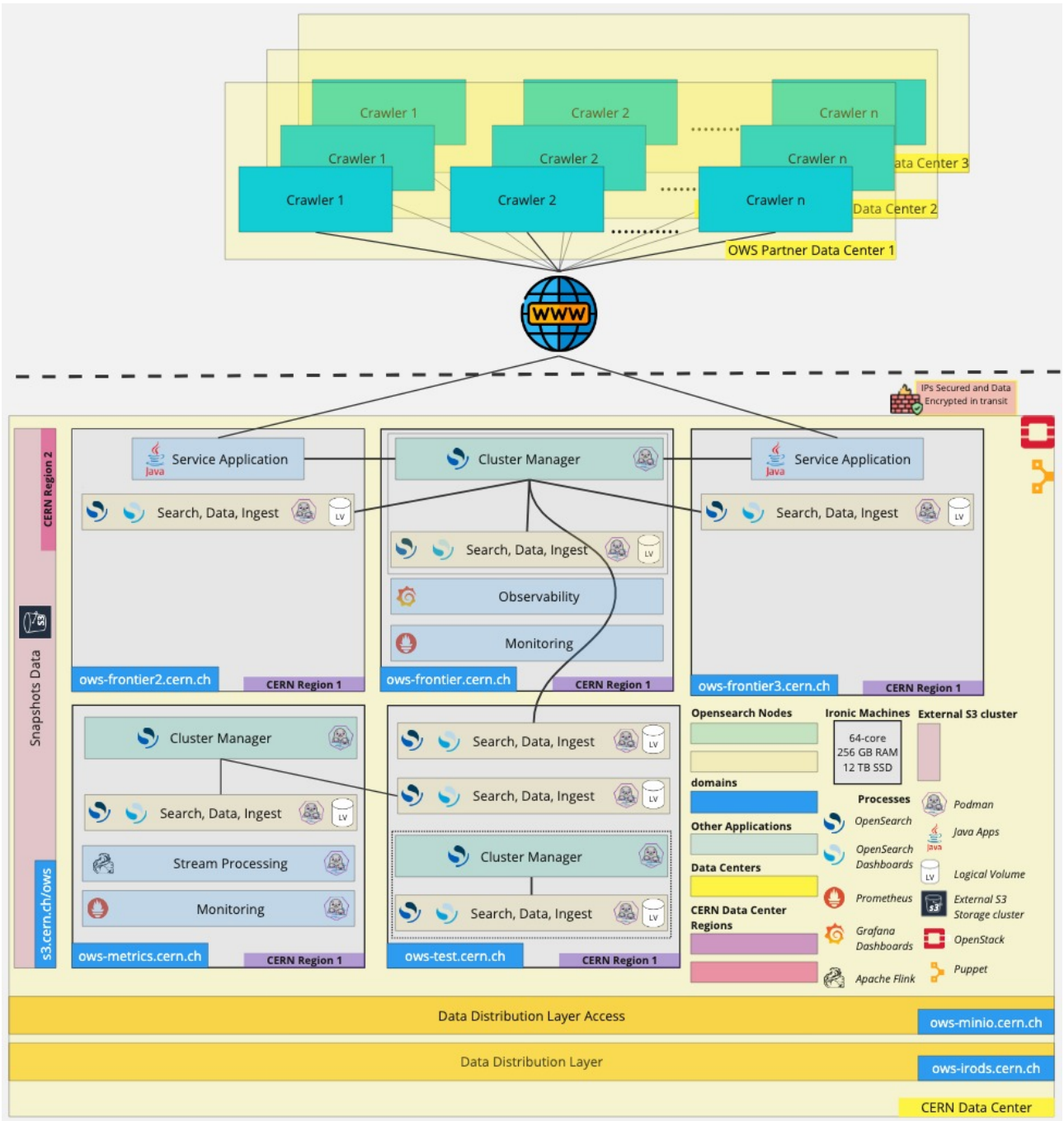



Figure 1: OWS Infrastructure at CERN

range of CERN-based use cases, served as an invaluable reference for our objectives [20]. However, their reliance on Puppet for configuration management presented certain difficulties. In light of OWS's dedication to open source values, we identified that a containerized infrastructure using tools like Podman [21] would offer considerable benefits over a Puppet-based setup. This method delivers a more user-friendly, scalable and portable solution that accommodates various experimental goals within the OWS and fulfills the expected integration needs of third parties [22]. Furthermore, this containerized configuration eases management for the OWS team at CERN by minimizing manual interventions, resulting in minimal downtimes and ensuring reliability.

The initial section covers the architecture of the URL Frontier service and it's interoperability with other services as depicted in Figure 1. The following section elaborates on the design of the OpenSearch service. The subsequent section provides an introduction to the new containerized infrastructure. This is followed by a section that outlines the roadmap, and finally, the remaining section offers a summary.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13872517



## THE URL FRONTIER SERVICE

This section provides a brief overview of the design and functionalities of the URL Frontier Service operational at CERN. It leverages the OpenSearch Service as its backend, which we will discuss in detail in the upcoming section.

By implementing the URL Frontier as a "service", we can handle it as an independent functional unit deployable and manageable across a network. This approach is designed for reusability, scalability, modularity, and interoperability within the broader service-oriented architecture of OWS's federated data infrastructure [19]. Interoperability allows seamless integration and communication with other services, enhancing system efficiency and flexibility. In addition, this design decision ensures effective management, high performance, and reliability, facilitating easier maintenance and updates without disrupting the entire system.

The URL Frontier Service, part of the OWLer Web crawling system, tracks the status of discovered and uncrawled URLs. Crawlers continuously fetch web pages from the public Web and retrieve URLs through remote calls to the URL Frontier service. After crawling, the URLs and discovered outlinks are uploaded back to the URL Frontier service, which updates the crawl status. Each crawling node interacts solely with the URL Frontier service to retrieve and upload web links.

The URL Frontier Service can consist of multiple Java front-end service applications, each connected to one or more crawler services, and an OpenSearch service acting as a back-end.

Figure 1 shows the configuration of the distributed OWS infrastructure of CERN. Two Java front-end service applications operate on distinct machines (ows-frontier2.cern.ch and ows-frontier3.cern.ch as shown in Figure 1), with the OpenSearch cluster distributed across these and two additional machines (ows-frontier.cern.ch and ows-test.cern.ch in Figure 1). The crawlers, hosted in other data centers (OWS Partner Data Center 1-3 in Figure 1), connect to the URL Frontier Service. The two Java Service Applications interact with these peer-to-peer crawling nodes in the partner data centers via the URLFrontier API [23] and connect to the OpenSearch cluster[1] via the OpenSearch Java high-level REST client [24]. It also connects to the OpenSearch cluster in the *Logs Aggregation and Monitoring Service*, which is on another machine (ows-metrics.cern.ch in Figure 1) This setup allows for various operations on the OpenSearch cluster, including indexing, searching, data management, logs, and metrics aggregation.

Figure 1 provides an overview of the above-mentioned services along with the other interface services [25]. OpenSearch Dashboards are utilized for data visualization and analysis due to their powerful and flexible tools that integrate seamlessly with OpenSearch. Smooth data processing and collection require all computing and storage systems to be fully operational and available. To continuously monitor and ensure the health and performance of these systems, we integrated Grafana for monitoring and Prometheus for observability. Grafana provides real-time visualization and dashboards, while Prometheus handles data collection in time series format and alerting. All the mentioned services are currently not accessible from outside the participating services that connect from the partner data center hosts, whose IPs are secured. To ensure encrypted communication between different services, we use SSL certificates from a highly trusted authority.

[1] Cluster: An independent OpenSearch cluster, dedicated to one use case.

## OPENSEARCH SERVICE DESIGN

This section provides an overview of the main subject of this paper. The architecture of the OpenSearch service is presented in detail, including the robust hardware infrastructure on which it operates. The figure also covers both legacy and modern deployment strategies; next we touch briefly on the dedicated test environment setup followed by authentication and authorization processes, and the various security measures implemented. In addition, it highlights the observability and monitoring mechanisms as well as the business continuity and disaster recovery strategies used. Figure 2 offers an illustration of the architecture, which is based on the more detailed design depicted in Figure 1. This figure maintains the representation of the data flow within the distributed setup of the service.

A significant aspect of this architecture is its utilization of CERN's two data center regions, as depicted in Figure 2 to ensure high availability and disaster recovery. By leveraging these geographically separated data centers, the service can maintain operational continuity even in the event of a failure in one region. This is achieved through a dedicated cluster approach, which minimizes external dependencies and enhances the flexibility and adaptability of the service. This approach not only supports the current use cases but is also designed to accommodate future requirements. The dedicated cluster setup ensures that the two services using OpenSearch as their data processing back-end can operate efficiently and reliably.

### *Hardware Specification*

The OpenSearch service at CERN is currently deployed on five physical machines within a single availability zone as depicted in Figure 2. Initially they ran CentOS Stream 8, later we had to migrate to Alma Linux 8 based on the advice of the CERN IT Linux Committee. [26] Each machine features 64 CPU cores, 256 GB of RAM, and 10.5 TB of usable allocated SSD space. These machines are managed by OpenStack [27] Ironic [28] without a virtualization layer. Configuration management is centralized using Puppet [29], with configuration data stored in GitLab at CERN [30].

### *Deployment Strategy*

Our existing two production clusters demand significant resources, hence they are each set up on dedicated machines. The adaptable nature of the configuration allows resource adjustments according to specific requirements. Logical

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13872517



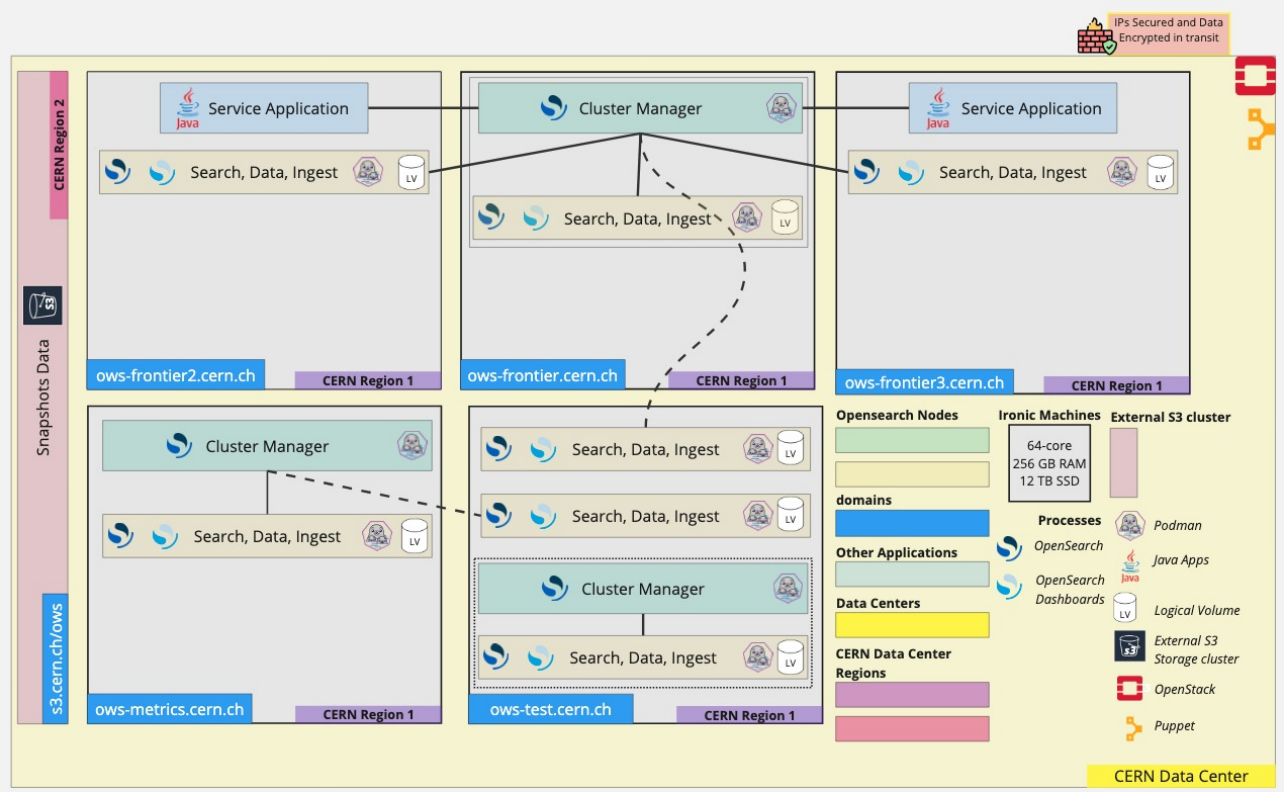



Figure 2: OpenSearch Service at CERN for OWS

Volume Manager (LVM) offers an abstraction layer over the physical storage available, facilitating the creation of Logical Volumes (LVs) that can extend across several physical disks. This allows dynamic volume resizing, backup snapshots, and the integration of new storage devices without requiring downtime. By efficiently managing storage across physical disks, LVs enable easy expansion as clusters near their capacity.

Due to the growing demand for additional data nodes[2], particularly for the URL Frontier Service, we incorporated data nodes (which also manage Search and Ingest roles) on two additional machines and one test machine, as depicted in Figure 2. These nodes are configured to connect to the original cluster manager node, which is different from the data node on the same machine. For greater storage needs, each cluster has the capability to connect to the external S3 storage cluster. In our infrastructure, a data node from each of the clusters handling Search and Ingest roles, similar to the production data nodes, is strategically placed on the test machine.

### *Dedicated test environment*

The test machine ‘ows-test.cern.ch‘ shown in Figure 2 is set up to closely replicate the production environment. This setup includes similar hardware and software configurations. It features a dedicated test cluster and also hosts test data nodes from the other two production clusters. New integrations and third-party services are initially tested on these nodes to ensure they work seamlessly with the production clusters and later on rolled out to production nodes. This setup also allows the validation of software updates, configuration changes, and new features before they are deployed in the production environment. This step is crucial to ensure compatibility and identify any potential problems, e.g. to identify breaking changes. By isolating the test environment, we can perform extensive tests without jeopardizing the stability and performance of the production clusters.

### *Authentication and Authorization*

The authentication is handled by OpenSearch, which maintains an internal database of user roles and encrypted passwords. Authorization is carried out within OpenSearch, incorporating CERN LDAP to manage access based on CERN e-groups.

### *Security*

Let’s Encrypt [31] fulfills our criteria for a highly reliable certificate authority by offering SSL certificates via an automated procedure to secure communications between services. We follow the official OpenSearch documentation for configuring security. Since Let’s Encrypt issues certificates in PEM format, which are incompatible with OpenSearch’s required PKCS12 formats, we use the OpenSSL [32] tool for conversion. By setting up OpenSearch with these SSL certificates, all communications between the nodes, between the clusters, and other services are encrypted, ensuring data

[2] An instance of the OpenSearch process

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13872517



security and integrity. Let's Encrypt certificates have a typical validity of 90 days. Using Certbot [33] we renew these certificates periodically and update the service(s) configuration as needed.

### *Resource Monitoring and Performance Evaluation*

The system utilizes Node Exporter [34] to collect resource indicator data, such as CPU usage, memory usage, and system load, and exports these data to the dedicated port. Prometheus, a powerful monitoring and alerting toolkit, collects these metrics from Node Exporter and stores them as time series data. The time series data allow Prometheus to efficiently store and query metrics over time, providing a historical view of system performance.

Grafana, a visualization tool, is deployed on the same machines to provide a user-friendly interface for real-time monitoring. Grafana connects to Prometheus to fetch the stored metrics and offers customizable dashboards that display various resource indicators. This setup allows us to gain insight into system performance, identify trends, and quickly troubleshoot issues.

In addition, the Prometheus Exporter Plug-In [35] for OpenSearch is used to expose cluster metrics in a format compatible with Prometheus. This plug-in collects key metrics such as cluster status, node status, and index status from the clusters via its REST API. By integrating these metrics into Prometheus and visualizing them in Grafana, we monitor the health and performance of the clusters alongside other system metrics.

### *Business Continuity and Disaster recovery*

We regularly take snapshots to backup the cluster's indexes and state. These snapshots are stored in an S3 cluster (s3.cern.ch/ows in Figure 1) in a different CERN data center region (CERN Region 2 in Figure 2), while our OpenSearch clusters are in another region (CERN Region 1 in Figure 2). This setup improves reliability, uptime, data integrity, and availability. In case of cluster failures, such as red health status, hardware issues, data corruption, or accidental deletions, snapshots have allowed us to restore indexes, minimizing downtime and data loss. We were able to quickly restore the unavailable or corrupted indexes from the latest snapshot to resume normal operations. Each of our clusters uses individual S3 buckets specified in the snapshot repository settings. We have a snapshot schedule for regular backups, semi-automated via the OpenSearch Dashboards Developer Tools, ensuring consistent snapshot creation by executing a query.

### *Migrating to containerized deployments*

To package and deploy the service along with its dependencies in isolated environments for open-source distribution and to manage a horizontally scalable multi-node OpenSearch cluster, we have transitioned to a multi-container infrastructure solution using Podman. Containers are lightweight, portable, and consistent in various environments, making them perfect for deployment and scaling. Using OpenSearch containers in this way allows us to have fine-grained control over the routing of incoming connections to the various other services that we are running. We were able to perform updates in production without service outages. When a third-party user decides to use our open-source distribution of the URL Frontier service, by making use of container limits, they should be able to deploy the services on low-end hardware without significant needs for intervention.

The firewall settings are adapted so that all machines can communicate with each other for each use case. Live data was started to be ingested as a test on one of the newly containerized cluster. The implementation process had near-zero downtime and allowed ample time for testing and verification.

LVs are mounted as persistent storage volumes within the containers. This setup ensures that data persists across container restarts and deployments. Containers are configured to use these mounted volumes for storing data, index templates, retention policies, and OpenSearch Dashboards objects.

## FUTURE WORK

Despite having a limited number of dedicated personnel, there are numerous opportunities to further investigate and improve the service. This section discusses some of the proposed ideas. To support service management and operations, the plan includes further developing and expanding the Puppet repository to automate various operational tasks. The host machines will transition from AlmaLinux 8 to AlmaLinux 9, following the recommendations of the CERN Linux Committee. Secrets management will be handled through Teigi [36]. OpenSearch's snapshot lifecycle management (SLM) feature will be utilized to ensure that snapshots are created consistently and without manual intervention with a fixed schedule. Furthermore, with the implementation of scalable multinode containerized infrastructure, we will also scale up our monitoring and logging infrastructure to a centralized system. This will allow us to aggregate logs from all OpenSearch nodes to a single location for easier management and analysis. Advanced alert mechanisms will be implemented using Prometheus to scrape additional metrics to monitor resource usage and trigger alerts when thresholds are exceeded, thus optimizing resource utilization and performance. Since May 20, 2023, CERN has been recognized as an official OpenSearch partner. The aim is to enhance CERN's visibility within the OpenSearch community and actively participate in meetings and forums to contribute to the community's development and growth.

## CONCLUSION

This paper provided an in-depth exploration of the successful implementation and operation process of the OpenSearch service at CERN for the OWS project. Motivated by maintainability, open source licensing, and improved features availability, the implementation aimed to overcome chal-

https://doi.org/10.5281/zenodo.13872517



lenges faced in the pilot version and leverage containerisations' capabilities. The new service architecture adopts a new improved deployment strategy with multiple nodes and clusters hosted on powerful machines, ensuring resource efficiency. The paper concludes with a roadmap for further automation, community engagement, and exploration of OpenSearch capabilities. In general, it provides a comprehensive and informative account of the implementation journey, making it a valuable resource for research projects considering or undergoing similar transitions.

## ACKNOWLEDGEMENTS

This publication has received funding from the European Union's Horizon [37] Europe Research and Innovation Program under Grant Agreement No. 101070014 (OpenWebSearch.EU, https://doi.org/10.3030/101070014).

## REFERENCES

[1] OpenWebSearch.eu, https://openwebsearch.eu/

[2] Granitzer, Michael and Voigt, Stefan et al. "Impact and Development of an Open Web Index for Open Web Search", in Journal of the Association for Information Science and Technology. August. 2023, Wiley. https://asistdl.onlinelibrary.wiley.com/doi/10.1002/asi.24818

[3] OpenWebSearch.eu, https://openwebsearch.eu/partners/

[4] OWLer, https://openwebsearch.eu/owler/

[5] M. Dinzinger, S. Zerhoudi et al, "A Distributed Open Web Crawler" Open Search Symposium (OSSYM), 2024, to be published

[6] OpenSearch, https://opensearch.org/

[7] OpenSearch implementation of the URL Frontier https://github.com/Presearch0fficial/opensearch-frontier/

[8] NGI funded URL Frontier https://www.ngi.eu/funded_solution/urlfrontier/

[9] NGI Zero https://www.ngi.eu/ngi-projects/ngi-zero/

[10] Golasowski, Martin, and Martinovic, Jan et al. "Toward the Convergence of High-Performance Computing, Cloud, and Big Data Domains", in *HPC, Big Data, and AI Convergence Towards Exascale: Challenge and Vision*, O'Reilly, 2022, doi: 10.1201/9781003176664-1 isbn: 9781032009841, 9781032009919, 9781003176664

[11] G. Hendriksen et al, *The Open Web Index: Crawling and Indexing the Web for Public Use*, Advances in Information Retrieval (ECIR 2024), March 2024. doi: 10.1007/978-3-031-56069-9 https://link.springer.com/chapter/10.1007/978-3-031-56069-9_10

[12] Apache Flink https://flink.apache.org/

[13] Munke, Johannes and Hayek, Mohamad et al, "Data System and Data Management in a Federation of HPC/Cloud Centers", in *HPC, Big Data, and AI Convergence Towards Exascale: Challenge and Vision*, O'Reilly, January 2022, doi:10.1201/9781003176664-4 isbn: 9781032009841, 9781032009919, 9780367766764

[14] iRODS, https://irods.org/

[15] MinIO, https://min.io/

[16] Opensearch Dashboards, https://opensearch.org/docs/latest/

[17] Prometheus, https://prometheus.io/

[18] Grafana, https://grafana.com/

[19] Noor A. Fathima, Wagner, A., Golasowski, M., Truckenbrodt, J., Mankinen, K., Hachinger, S., Granitzer, M. (2024). Launch of the Pilot Infrastructure. Zenodo., https://doi.org/10.5281/zenodo.10838954

[20] Papadopoulos Sokratis, Saiz Pablo et al. "Architecting the OpenSearch service at CERN", EPJ Web of Conf, 2024, volume: 295, pages: 07006 doi: 10.1051/epjconf/202429507006, https://doi.org/10.1051/epjconf/202429507006"

[21] Podman https://podman.io/

[22] 3rd party Data Center partners call https://openwebsearch.eu/community/3rdparty-calls/call3/

[23] original version of the URL Frontier API here. https://nlnet.nl/project/URLFrontier/

[24] OpenSearch Java Client https://opensearch.org/docs/latest/clients/java-rest-high-level/

[25] Noor A. Fathima et al, "Federated data infrastructure for the open web search" Open Search Symposium (OSSYM), 2024, to be published

[26] Migration from CentOS 8 Stream https://linux.web.cern.ch/centos8/

[27] OpenStack, https://www.openstack.org/

[28] Ironic: Bare Metal as a Service, https://ironicbaremetal.org

[29] Puppet https://www.puppet.com/

[30] Gitlab At CERN https://about.gitlab.com/customers/cern/

[31] Let's Encrypt https://letsencrypt.org/

[32] OpenSSL https://www.openssl.org/

[33] Certbot https://certbot.eff.org/

[34] Node Exporter https://github.com/prometheus/node_exporter

[35] Prometheus Exporter Plugin for OpenSearch https://github.com/Aiven-Open/prometheus-exporter-plugin-for-opensearch

[36] U. Schwickerath, P. Saiz, Z. Toteva, Securing and sharing Elasticsearch resources withRead-onlyREST, in EPJ Web of Conferences (EDP Sciences, 2019), Vol. 214, p. 08032.

[37] Horizon Europe, https://research-and-innovation.ec.europa.eu/funding/funding-opportunities/funding-programmes-and-open-calls/horizon-europe_en

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13863799



# ATRA: A POWERFUL, LIGHTWEIGT APPROACH TO CRAWLING

Felix Engl*, University of Bamberg, Bamberg, Germany

*Abstract*

In the digital age, the complexity of traditional web crawlers limits their accessibility to experts and large organizations, leaving a gap in the democratization of web data extraction. In this paper, we present Atra, an innovative solution to this challenge, designed to simplify the web crawling process while maintaining comprehensive scraping capabilities. By enabling comprehensive link extraction from a variety of document formats and archives, Atra facilitates deeper web exploration. It is also easy-to-use, making web crawling accessible to a wider audience without the need of a high investment in programming, including individuals with minimal technical skills, small organizations, and independent researchers.

## MOTIVATION

Web crawlers play a critical role in navigating, indexing, and analyzing web content. Existing web crawlers, such as StormCrawler, Apache Nutch, Heritrix, and Scrapy, provide robust web crawling and data extraction solutions, but often come with a steep learning curve and require significant setup or programming skills. This complexity limits the accessibility of web crawling technologies for a wide range of users, particularly those with non-technical backgrounds or limited programming expertise. This means that not only is the research field of web search and web crawling very difficult to access, but in addition, scientific research is driven only by a small group of independent scientists and large organisations such as Microsoft or Google.

In response to these challenges, we present Atra[1], a novel web crawling solution, implemented in Rust, designed with the primary goal of scraping websites as comprehensively as possible, while ensuring ease of use and accessibility to a wide range of users. The name Atra comes from the Erigone atra, a dwarf spider with a body length of 1.8mm to 2.8mm. Not only do they play a central role in natural pest control in agriculture (aphids), but they are also aerial spiders that can travel long distances by ballooning, also known as kiting.

Atra distinguishes itself from existing web crawlers by its ability to perform link extraction from a wide variety of document formats and archives, coupled with a user-friendly interface, without the need for complex configuration or specialized knowledge. This is crucial for democratizing web crawling, by enabling small organizations, independent researchers, and hobbyists who may lack the resources to engage with more complex web crawling frameworks.

In the next chapter, we will discuss various relate crawler frameworks and briefly discuss their usability in smaller projects. After a description of the currently existing Atra features we compare Atra with the two, currently, most prominent crawlers frameworks StormCrawler and Apache Nutch. In the last chapter we describe the roadmap of Atra for reaching maturity.

## RELATED SOFTWARE

In the field of web crawling, several frameworks have been developed to meet the diverse needs of researchers and practitioners. Among them, StormCrawler[2], Apache Nutch[3], Heritrix[4], and Scrapy[5] have emerged as prominent tools, each with different features, tailored to specific use cases. This section provides a comparative analysis at a superficial level based on personal experience with the frameworks while working on a small crawl for a vertical search project over german schools. The following analysis will focus on describing the scalability, ease of use, flexibility, performance, and usability in smaller projects, followed by a short conclusion.

**StormCrawler** is designed for scalability and real-time processing, leveraging Apache Storm's [1] fault tolerance and stream processing capabilities [2]. Its modularity allows for extensive customization, making it suitable for large-scale, real-time web crawling projects. During a small project to crawl 700 websites for a vertical search system, we learned that the complexity of Apache Storm can present a steep learning curve, and the crawler requires significant setup and configuration efforts. StormCrawler also requires elaborate extensions for use case specific features like budgeting based on the crawler depth on specific domains, and extracting links and harvesting content that is not supported by default (e.g. plain text, XML, Word, PDF, JAR-Files, executable).

**Apache Nutch**, another scalable solution, excels at handling large data sets. It integrates seamlessly with Apache Hadoop [3] for processing and distributed storage, and benefits from a robust community and extensive support due to its long-standing presence in the field [4]. Despite its scalability and strong integration capabilities, the complexity of Apache Nutch, in combination with Hadoop, in configuration and customization may deter novices, and its extensive features may be excessive for small to medium crawling requirements. Another problem is that Apache Nutch is optimized for common crawling scenarios, meaning that specialized crawling scenarios either require a lot of configuration effort or the development of custom modules. This becomes particulary clear when facing domain specific bud-

* felix.engl@uni-bamberg.de

[1] https://github.com/FelixEngl/atra (last visited: 21.03.2024)

[2] `https://stormcrawler.net/` (last visited 28.02.2024)

[3] `https://nutch.apache.org/` (last visited 28.02.2024)

[4] `https://github.com/internetarchive/heritrix3` (last visited 28.02.2024)

[5] `https://scrapy.org/` (last visited 28.02.2024)

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863799



geting or very strict requirements on crawling politeness[6] like crawling with *.onion*-addresses[7].

**Heritrix** is known for its specialization in web archiving [5]. Developed by the Internet Archive, it focuses on capturing a broad array of web resources for preservation purposes. While Heritrix's emphasis on archiving makes it unparalleled for such tasks, its user interface and flexibility may not meet the needs of projects beyond web archiving. While evaluating it for our small project, we noted that Heritrix supports a wide array of link extraction methods for various common web-file formats as well as excellent budgeting for single websites. Similar to Apache Nutch, adding features for depth recording and link graphs, requires either configuration effort[8] or custom modules. Another problem that we encountered while experimenting with Heritrix is, that some of the extraction methods are not up to date, mostly focusing on old formats or brute forcing it by matching needles with haystacks [9].

**Scrapy** offers a balance between usability and functionality, characterized by its straightforward setup and high degree of flexibility for various web scraping and crawling operations [6]. Its efficient resource management and performance make it even suitable for larger crawls, although achieving distributed crawling may require additional tools or cloud-based deployments, potentially limiting its scalability and ease of use compared to frameworks explicitly designed for distributed environments. Compared to the other tools mentioned above we made the experience, that Scrapy requires less configuration and programming effort than the other frameworks. But the effort required when working with non-html-files and custom budgeting is similar.

In conclusion, all tools mentioned above are excellent tools for casual crawling tasks, and depending on the requirements of the project it is necessary to use one of them. But with regards to ease of use in smaller projects, all crawler either require a complex setup or programming for basic functions like export and extraction. For this reasons, we have developed Atra a minimalist, self-contained crawler for crawl tasks, without relying on external services or complex configuration and module systems, in favor of a minimalistic simplicity and robustness.

## FEATURES

In this chapter we will describe the current features of Atra, regarding its focus on comprehensive data harvesting and link extraction, handling legacy web content, and efficient handling of crawled data. Some of the features mentioned below are tailored to meet the needs of specific user groups, ranging from research and academic communities to industry practitioners who require deep, precise web scraping capabilities.

### *Legacy Encoding Support*

Atra incorporates advanced encoding recognition capabilities for websites, when the content-encoding header is missing, leveraging an encoding detection with chardetng [7] and efficient decoding based on the encoding standards [8] of the WHATWG Steering Group to support 35 different encodings. Such comprehensive encoding support—ensuring the correct interpretation and processing over a diverse range of languages and character sets—is especially crucial for web scraping in a vertical search context, where it is in some cases necessary to handle legacy and non-english content with high accuracy.

### *Crawling Strategy and Link Extraction*

To crawl websites as efficiently as possible and reduce the number of accidental revisits, Atra uses a depth-first crawling strategy (see fig. 1).

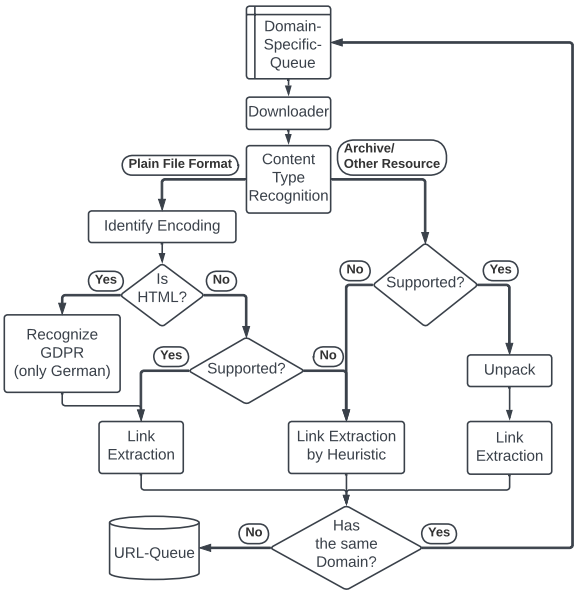



Figure 1: The crawler loop of Atra with the link extraction and GDPR recognition feature.

The link extraction of Atra consists of several steps, starting with a simple content type recognition to distinguish between plain text files and archives/other resources (see fig. 1). If a plain text file is detected, Atra tries to derive its encoding from the response header, possible meta tags (for XML and HTML only), and encoding detection heuristics, and applies the necessary decoding to convert the content to UTF-8 for link extraction. If the plain text file is HTML, Atra also identifies and excludes the General Data Protection Regulation (GDPR)[10] in an additional step, if configured by the

[6] `https://cwiki.apache.org/confluence/display/nutch/OptimizingCrawls` (last visited 28.02.2024)

[7] `https://cwiki.apache.org/confluence/display/NUTCH/SetupNutchAndTor` (last visited 28.02.2024)

[8] see https://github.com/internetarchive/heritrix3/wiki/Common Heritrix Use Cases (last visited 29.02.2024)

[9] see `https://github.com/internetarchive/heritrix3/tree/master/modules/src/main/java/org/archive/modules/extractor` (last visited 29.02.2024)

[10] Currently only for German websites.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863799

user (see section GDPR-Ident). The links in a text file type are then extracted either by a specialized extractor or by a plain text heuristic. Atra's specialized HTML extractor can also resolve links and extract data from elements, such as <area>, <base>, <img>, <iframe>, and <script> tags. Furthermore it can process data URLs and uses heuristics to identify dynamic links triggered by "onclick"-events.

### *File Type Detection*

When Atra detects non-text files, it tries to determine the file type from the content information provided, or infer the file type from the magic number or meta information. The package used to identify the magic numbers supports 437 file types[11].

### *GDPR-Ident*

Atra's *GDPR-Ident* feature represents a significant advancement in vertical/focused web crawling, addressing the critical need to identify and, if necessary, remove GDPR consent banners while crawling a website by leveraging machine learning techniques. Specifically, it uses Term Frequency-Inverse Document Frequency (TF-IDF) vectors derived from HTML subtrees as inputs to a Support Vector Machine (SVM) classifier. The SVM is trained on a curated dataset of 250 annotated documents containing a variety of GDPR consent banner implementations. This approach allows for nuanced detection of GDPR-relevant content across different web pages. In empirical evaluations of the prototype implementation [9], we observed a remarkable accuracy rate of 96% in identifying GDPR banners, see table 1. This high level of accuracy also underscores the effectiveness of combining TF-IDF vectorization with SVM classification in addressing the complexities of regulatory compliance in web crawling.[12]

| | | **True Label** | | |
|---|---|---|---|---|
| | | Positive | Negative | Total |
| **Predicted Label** | Positive | 27 | 0 | 27 |
| | Negative | 2 | 21 | 23 |
| | Total | 29 | 21 | 50 |

Table 1: The confusion matrix obtained by testing the prototype of the GDPR-identification system. [9]

### *Data Handling*

Atra is also designed to archive websites as completely as possible. Hence, Atra does not only collect all header information of a response, but also collects and analyzes all files encountered on a website. For storing the crawled data Atra uses a combination of Notation3 (n3), WARC/1.1 and binary files (for data exceeding 100 Megabyte[13]). Internal data, such as skip-pointers, metadata and the overall crawl state for every URL are stored with RocksDB[14].

### *Modes and Clustering*

Due to the fact, that Atra follows a philosophy of simplicity and self-containment it only supports three different modes: *single*, *multi*, and *cluster*. The following paragraph will explain the different modes and describe how they can be used in research and practice in general.

**Single-Mode** *Single*-mode allows to crawl a single website with Atra using a Command Line Interface (CLI). When starting Atra in *single*-mode, the only information required is the agent name, seed URL, and target crawl depth. The seed is then scraped as completely as possible.

**Multi-Mode** *Multi*-mode runs in multiple threads on a single machine to crawl multiple seeds at once. Basically, Atra behaves in this mode like in *cluster*-mode, but it omits the boilerplate for controlling crawl politeness between multiple machines as well as fail-safe mechanisms required in a network. *Multi*-mode is started via CLI, but instead of simple commands, Atra now requires two configuration files (`atra.ini` and `crawl.yaml`). `atra.ini` contains general application settings like cache sizes, while `crawl.yaml` configures the crawl behavior in a more sophisticated way (e.g. user agent, politeness and domain specific crawl budgets).

**Cluster-Mode** The *cluster*-mode is similar to *multi*-mode, but instead of a single instance on a single machine, there are multiple instances on multiple machines working in a computing cluster as depicted in fig. 2. In *cluster*-mode, Atra needs at least two instances per machine. One of these instances, from here on referred to as "orchestrator", is handling the work balance, connection to the cluster, crawler politeness, and handling subordinates. The other instances on the machine (called "subordinates"), are used for crawling and data collection.

In a cluster each orchestrator is assigned a specific range of domains and other orchestrators in the cluster send URLs, matching this specific domain-range, to the assignee. When the cluster is running, all orchestrators periodically renegotiate their domain assignments to accommodate uneven workloads. Additionally, as a fail-safe mechanism, the orchestrators periodically synchronize the information about the workload and URL of the orchestrator via the Paxos protocol [10] (see fig. 3). If a participant drops out, and is therefore missing during the regular negotiations, the last known state of the dropout is used to crawl the probably missing websites with an other participant.

## COMPARISON

To highlight the differences between Atra and other crawlers, we compare the performance of Atra with Storm-

[11] `https://docs.rs/file-format/0.24.0/file_format/` (last visited 28.02.2024)

[12] Protoype code and research will be published alongside Atra.

[13] Otherwise, the binary files are stored with BASE64 encoding in the WARC-file

[14] https://rocksdb.org/ (last visited 21.03.2024)

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863799



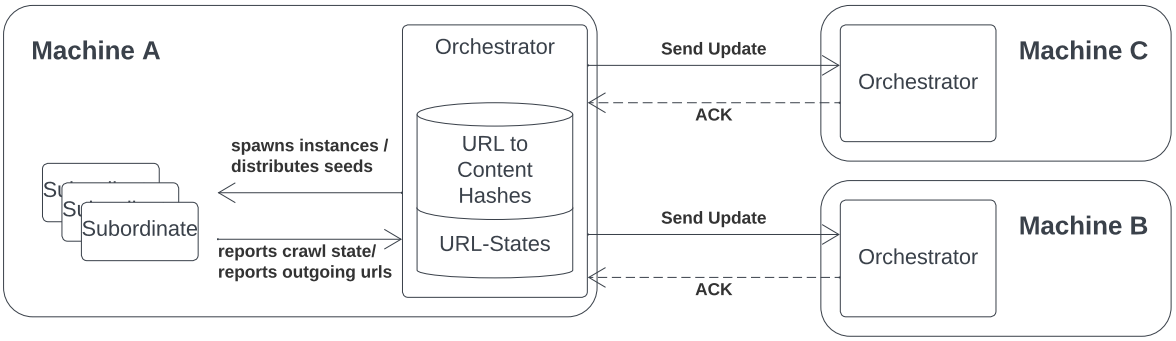



Figure 2: An exemplary view of a machine running an orchestrator with subordinates. Additionally the example depicts the communication between Machine A in a cluster with two other participants.

**Update Package**

**Orchestrator ID:** <MAC> or <IP> or <Unique Name in Cluster>

**Update ID:** <Incremental ID> // All updates from are participant have to be applied in order

**Current time of Orchestrator :** <UTC Timestamp> // Can be out of sync between machines

**Workload Information:**

| Domain | Workload | Timestamp (UTC) | Recrawl In |
|---|---|---|---|
| opensearchfoundation.org | 5min | 2024-03-01 7:05:56.5768348 +00:00:00 | 7 Days |
| ... | ... | | |

**URL Information:**

| URL | STATE | Timestamp (UTC) | LSH |
|---|---|---|---|
| https://opensearchfoundation.org/ | Stored | 2024-03-01 7:00:56.5768348 +00:00:00 | 1536:J2ap3J7EJs...FL0u |
| ... | ... | ... | ... |

Figure 3: A schematic of an update package. The workload information consists of the domains crawled by the orchestrator and information when the crawl is finished and when a recrawl is necessary. The URL information consists of the URL-state (e.g. "Discovered", "Stored") and a locality sensitive hash (abbreviated with LSH).

Crawler, as well as Apache Nutch, under uniform test conditions in table 2. Then we use a comparison table of the most common requirements for vertical crawls to compare the three contestants (see table 3).

## *Performance*

Since StormCrawler and Apache Nutch are designed to crawl in a cluster, this paper does not analyse crawl speed, but single machine efficiency. This is because the pages/sec metric in a cluster does not depend on the efficiency of a program, but on scaling. Furthermore, Atra is intended to be used in smaller companies or by individuals who may not have access to large cluster configurations. Therefore, we decided to compare all three contestants based on their memory footprint during the operation of each crawler in a minimalistic setup.

In order to keep the comparison between the crawlers unbiased with respect to a particular operating system or special configurations, we used the Docker container configurations for each crawler as suggested in the respective crawler manuals[15][16] (see table 2). Assuming that the Docker containers are using the bare minimum required to run the crawlers, we have adopted the Docker container memory metrics for our comparison. This also has the advantage that we can ensure that we are measuring not only the memory usage of the crawler tasks themselves, but also any in-memory buffers used by the OS when writing to a disk.

To mitigate the influence of the caches, we recorded the memory footprint of the cold systems and the warmed up

[15]https://github.com/apache/nutch/ (last visited: 20.03.2024)
[16]https://github.com/DigitalPebble/stormcrawler-docker (last visited: 20.03.2024)

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863799

system after crawling five seeds to an absolute depth of three (resulting in roughly 5000 crawled web-pages). To account for variability and to avoid misrepresentation, especially for Java-based programs where garbage collection can skew the results, we computed the total memory usage by rounding the minima down and the maxima up to the nearest hundred.

The results of our measurements show a stark disparity in resource utilization (see table 2): Atra's memory efficiency is underscored by a footprint of 15 to 60 megabytes. Conversely, Apache Nutch requires 300 to 500 megabytes in its local operation mode within the provided Docker container. StormCrawler's requirements dwarf its counterparts, requiring between 1,500 and 2,000 megabytes due to its multi-service architecture.

| | **Apache StormCrawler** | **Apache Nutch** | **Atra** |
|---|---|---|---|
| MIN (MB) | 1,500 | 300 | 15 |
| MAX (MB) | 2,000 | 500 | 60 |
| Docker Service Count | 5 | 1 | 1 |

Table 2: Comparison of the memory consumption of StormCrawler, Apache Nutch, and Atra during operation, according to the Docker container memory metrics. Measurements were taken five times, from the cold and warmed up system, after crawling five seeds to an absolute depth of three. The lowest (MIN) and highest (MAX) values were then rounded to the nearest hundred. The last row shows the number of Docker containers required to use the crawler on a system.

### *Feature-Comparison*

This section is a comparison of the basic features (see table 3) available in StormCrawler, Apache Nutch and Atra.

**Environment** Regarding the environment, it is noteworthy that Atra has the least environmental requirements, while the other crawlers require some kind of framework or runtime (see table 3).

**Indexing** Looking at the available indexing methods of the three crawlers, only Atra currently supports WARC/1.1 + Community Annotations[17] and Notation3. Due to the fact that Atra does not perform any filtering or other preprocessing steps (except GDPR), it is sufficient for Atra to combine these standards with skip-pointers and RocksDB to provide an efficient method for managing the crawled data, without relying on external services. Atra's roadmap includes expanding its indexing capabilities to encompass Solr and Elasticsearch, aligning with industry standards provided by StormCrawler and Apache Nutch, as well as improving the user-comfort.

**Link Extraction** As mentioned in the previous text, one of Atra's goals is to be a minimalist and self-sufficient crawler with a high link discovery rate. Therefore, Atra supports the most common online formats natively (see table 3). In addition, Atra uses heuristics to extract links from unknown file types as well as raw text (see "Binary" table 3). In future releases of Atra, we will also provide an optional REST API for Tika servers to comply with industry standards. For JavaScript link extraction, Atra currently parallels the capabilities of Apache Nutch.

**Further Features** Unique to Atra is its GDPR recognition/filtering and live web graph building by streaming Notation3 tuples to a file. Similar to StormCrawler and Apache Nutch, Atra supports Single Page Application (SPA) handling. But instead of using the Chrome DevTools protocol, Atra will use "Servo", an embedded browser. However, to comply with industry standards, Atra will also support the Chrome DevTools protocol as an alternative.

## FUTURE WORK

Looking at the road map of Atra, our vision encompasses a comprehensive suite of enhancements aimed at reaching a position as minimalistic, robust and user friendly crawler on par with other state of the art crawlers.

Firstly, we intend to significantly expand the range of file formats that Atra can process for link extraction. Specifically for JavaScript, we plan to improve JavaScript link extraction by enhancing the existing extraction method, as well as executing and analysing JavaScripts (JSs) variables and communication in an embedded, sandboxed environment powered by V8 [11]. These enhancements will allow Atra to dig deeper into web resources, uncovering connections and data across an even wider range of document and media types, making the crawler more useful for research and analysis. We also plan to add support for various protocols, such as FTP, to broaden the available sources for data collection.

A key goal of Atra is to use Servo, an embeddable web browser engine written in Rust, to handle SPAs. Servo has recently seen a resurgence in activity and development, supported by external funding[18], which is expected to lead to a stable embedding API. This upcoming integration aims to enhance Atra's SPA handling capabilities while maintaining its design principles of autonomy, security and efficiency.

In response to the specific needs of law enforcement and cybersecurity professionals, we also plan to integrate support for navigating the dark web through TOR, with a particular focus on onion address crawling. This capability will provide essential tools for secure and anonymous investigations within the confines of the Dark Web.

These envisioned enhancements to Atra are designed to reinforce its core principles while addressing the dynamic challenges of web crawling. Through these strategic developments, we aim to deliver a tool that is not only more powerful

[17] see https://iipc.github.io/warc-specifications/specifications/warc-format/warc-1.1-annotated/ (last visited: 20.04.2024)

[18] "Servo to Advance in 2023" - `https://servo.org/blog/2023/01/16/servo-2023/` (last visited 29.02.2024)

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863799



| | Apache StormCrawler | Apache Nutch | Atra |
|---|---|---|---|
| | **Environment** | | |
| OS | Cross-Plattform | Cross-Pattform | Windows, Linux |
| Written in | Java | Java | Rust |
| Dependencies | Apache Storm | Apache Hadoop | - |
| Docker | + | + | + |
| Scalable | + | + | o |
| | **Indexers** | | |
| WARC | 1.0 | 1.0 | 1.1 + Annotations |
| Solr | + | + | o |
| Elasticsearch | + | + | o |
| SQL | + | - | - |
| AWS CloudSearch | + | + | - |
| OpenSearch | + | - | - |
| Rabbit | - | + | - |
| CSV | - | + | - |
| N3 | - | - | + |
| | **Link Extractors** | | |
| HTML | Native | Native | Native |
| JS | - | Native | Native |
| CSS | - | - | o (Native) |
| PDF | Tika | Tika | o (Native) |
| DOCX/XLSX/PPTX | Tika | Tika | o (Native) |
| TXT | Tika | Tika | Native |
| ZIP | - | Native | o (Native) |
| BIN (Binary) | - | - | Native |
| TIKA-REST | - | - | o (Native) |
| | **Further Features** | | |
| GDPR-Recognition | - | - | + |
| Single Page Applications | Selenium | Selenium | o (Native/Selenium) |
| Dedublication | - | + | o |

Table 3: A feature comparison between Apache StormCrawler, Apache Nutch and Atra. An “o” denotes planned features in future releases of Atra, that are deemed necessary to reach full maturity or to comply with industry standards.

and versatile but also remains user-friendly and accessible to a wide array of users, from researchers to cybersecurity professionals and law enforcement agencies worldwide.

## ACKNOWLEDGMENT

Many thanks to Erik J. Schmidt for his indispensable contribution to Atra with his work on the GDPR-Ident feature.

## REFERENCES

[1] N. Marz, *Apache Storm*, 2024. https://storm.apache.org/

[2] J. Nioche, *StormCrawler*, 2024. https://stormcrawler.net/

[3] Apache Software Foundation, M. Cafarella, and D. Cutting, *Apache Hadoop*, 2024. https://hadoop.apache.org/

[4] Apache Software Foundation, M. Cafarella, and D. Cutting, *Apache Nutch*, 2024. https://nutch.apache.org/

[5] Internet Archive, *Heritrix*, 2024. https://github.com/internetarchive/heritrix3/wiki

[6] Zyte, *Scrapy*, 2024. https://scrapy.org/

[7] H. Sivonen, *Chardetng: A More Compact Character Encoding Detector for the Legacy Web*, English, Blog, 2020. https://hsivonen.fi/chardetng/

[8] WHATWG Steering Group, *Encoding Standard*, English, Book, 2024. https://encoding.spec.whatwg.org/

[9] E. J. Schmidt, “DsgvoAssistant - Removing DSGVO Content from HTML,” German, University of Bamberg, Project report, 2024, Unpublished, p. 21.

[10] L. Lamport, “Time, clocks, and the ordering of events in a distributed system,” en, *Communications of the ACM*, vol. 21, no. 7, pp. 558–565, 1978. 10.1145/359545.359563

[11] Google Inc., *V8 JavaScript Engine*, 2024. https://v8.dev/

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13863478

# OWLER: A DISTRIBUTED OPEN WEB CRAWLER

M. Dinzinger*, S. Zerhoudi,
J. Mitrović, M. Granitzer,
University of Passau, Passau, Germany

*Abstract*

The public availability of web data has become a main driver for innovation in the domains of Artificial Intelligence and Web Search. In this course, we have proposed the Open Web Index (OWI), a publicly funded service for providing enriched and indexed web documents to foster the development of new search applications and data products. This mission requires, among others, a comprehensive, continuous crawling effort, based on the cornerstone principles of openness and legal compliance. In order to overcome the posed technical challenge, we have further presented the Open Web Crawler (OWLer) [4], an open-source software framework driving the large-scale collection of web documents.[1]

OWLer constitutes the backbone of a fully integrated processing pipeline for indexing and sharing large amounts of curated web resources. Due to the ambitious vision of OWI as well as the geographic dispersion of the therefore available compute resources, the crawling system requires to be highly distributed and collaborative. Technically, OWLer bases on open-source projects such as StormCrawler, OpenSearch and the URLFrontier framework. Our work is an aggregation and refinement to existing efforts in open-source web crawling in order to accommodate the combined requirements of scalability, efficiency and transparency. In this paper, we present the conceptual and technical background, discuss design decisions and overview the architecture of the OWLer crawling system.

## INTRODUCTION

The dominance of a few commercial search engines has led to a closed web search ecosystem where publishers must optimize their content for these gatekeepers, potentially sacrificing quality and hindering innovation [5]. To counter this, we have proposed the development of an Open Web Index (OWI) as publicly funded infrastructure, guided by the core principles of open data, legal compliance and collaborative technology. The *Open Web Crawler (OWLer)*, implemented as distributed, incremental crawling system, takes a central position in this joint development effort [6]. The intuition of OWLer resembles the motivation behind the non-profit organization Common Crawl.[2]

Whereas Common Crawl regularly publishes large-scale collections of crawled web documents as well as complementary data products such as clean-text corpora and aggregated web graphs, OWLer is crawling continuously on a federated infrastructure. The collaboration of different European institutions and infrastructure providers is crucial for the success of OWI and the encompassing crawling effort. The raw web data collected by OWLer is integrated with the technical pipeline for collaboratively building a rich web index. In order to collect enough resources given the available structural setup, the system architecture is oriented towards two major objectives:

- Modularity
  The compute resources available for the project are highly heterogeneous and dispersed over multiple datacenters in Europe. In order to maintain such as a system with manageable effort, it requires a compact and modular architecture employing well-defined interfaces for the communication between remote services.

- Scalability
  The ambitious vision of OWI necessites comprehensive crawling covering a significant part of the text-based surface web. No few-node cluster and no single European institution can take up this task on its own. The required performance is rather grounded in the system's ability to scale horizontally, integrating more nodes located in collaborating infrastructure providers.

This paper overviews OWLer by providing general and technical background information on the design of the current system. After taking a look into the related work, Section 3 describes the distributed architecture in more detail by presenting each of its three core software components. We further share preliminary results of its current live deployment (Section 4) and conclude with a resumée of the main challenges in the previous development as well as an outlook on the future development of OWLer.

## RELATED WORK

The domain of web crawling dates back to the origins of the world wide web in early 1990s. Throughout these years, search engine operators and reseachers worked on software tools for the efficient traversal of the web. Due to a wide range of research endeavors, crawling systems have continously improved on its four main quality criteria: coverage and freshness, politeness and robustness [11]. Along the way, numerous technical challenges have been studied and overcome, e.g., near-duplicate detection of web documents [3, 10] and focused crawling [2].

* michael.dinzinger@uni-passau.de

[1] Link to open repositories:
`https://opencode.it4i.eu/openwebsearcheu-public`

[2] `https://commoncrawl.org`

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863478



Several notable software tools have been developed over the years. *Mercator*, described in Najork et al. [13], was one of the first commercial open-source crawlers that targeted high-performance. Along with Mercator, Najork et al. introduced the *URL frontier* (or URL manager). This software component is implemented as multi-level queue-based data structure and schedules URLs depending on some priority criteria (e.g. web page quality), while ensuring politeness towards web servers through request delays. *Heritrix* and the open-source crawler *Apache Nutch* were further early web crawlers that have been extensively used in academia and industry [8, 12]. The *IRLbot*, published in 2008 [9], was a pioneering effort in scaling open-source web crawling to handle billions of web pages on a single-machine setup. Similarly, *UbiCrawler* and its successor *BUbiNG* were developed by Boldi et al. [1] to achieve maximal throughput on a single powerful machine. The crawling tool is able to process several thousand pages per second, achieving an optimized utilization of the hardware while respecting politeness constraints.

## SYSTEM COMPONENTS

The OWLer crawling system is designed to handle the challenges of a highly heterogeneous and distributed infrastructure with machines of different sizes and locations. The modular architecture consists of three loosely-coupled but collaborating tiers, as shown in Figure 1. Each tier is a self-contained software project and fulfills distinct, complementary tasks. The implementation of crawlers and the distributed database system can be interchanged due to well-defined interfaces in the URL Frontier layer.

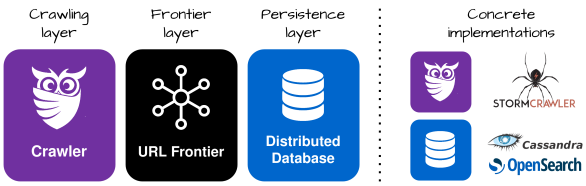



Figure 1: Three layers in the system architecture

### *Crawling*

The first layer consists of software tools for continuously fetching web pages without managing the set of discovered links (crawl space), which is the task of the other two tiers. A crawler instance retrieves URLs to be fetched through a remote call to a URL Frontier instance and adds them to its internal task queue (see Figure 2). The web pages are downloaded, parsed, and supplemented with meta information corresponding to the page content. Finally, the URL and discovered outlinks are uploaded back to the URL Frontier instance, which updates the status of the crawl. The crawlers are lightweight and can run on commodity-sized machines in any computing center with sufficient external network bandwidth.

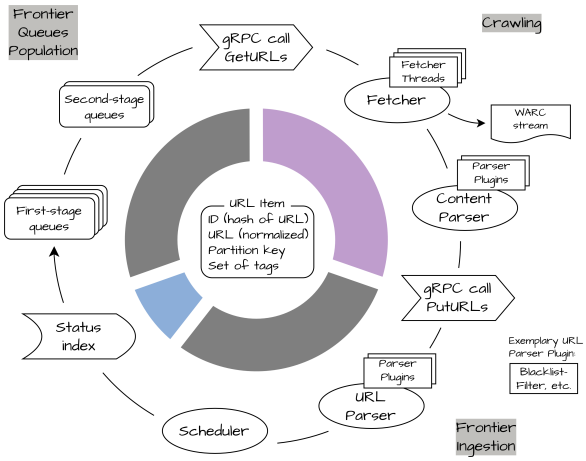



Figure 2: Technical crawling pipeline

### *URL Frontier*

A crawling node communicates only with the URL Frontier instance, from which it retrieves and to which it uploads web links (*URL items*). The communication between the crawler and frontier nodes is implemented as remote calls over a gRPC connection,[3] resulting in a loose coupling between them. They rely on a well-defined Protocol Buffers [4] API in the URL Frontier project.[5] The URL items exchanged consist of the normalized plain-text URL, a unique identifier (based on the URL hash), the partition key, and a set of tags. The tags are extracted by the Content Parser or the URL Parser, which internally call Parser Plugins. Tags can also be provided by users through a social tagging system, potentially contributing to a higher quality crawl by integrating user-curated meta information during data collection.

The frontier tier comprises one or more instances (also called services), each connected to one or more crawler instances and one storage backend for persisting URL items. Each URL Frontier service is assigned to a section of the *crawl space*, defined as the set of discovered web links that expands over time as the crawl continues.[6] A single service is responsible for its own partition of the crawl space, providing its clients with the next URLs to be fetched while ensuring a sufficient time interval between subsequent fetches of the same resource.

As shown in Figure 2, URL items uploaded by the crawlers are ingested and persisted. The first part of the frontier ingestion pipeline filters and parses the URL, including the execution of Parser Plugins. One exemplary Parser Plugin with a significant positive impact on the system's robustness

[3] `https://grpc.io`
[4] `https://protobuf.dev`
[5] Find the original version of the API here: `http://urlfrontier.net`
[6] The border between the crawl space and the undiscovered web is also called *frontier*, which is the source of the name URL Frontier.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863478

is the `BlacklistFilter`, which checks web links against a number of public spam databases. The ingestion pipeline also includes the Scheduler component, which determines the next planned fetch date of the web resource based on the quality and change frequency of the page content.

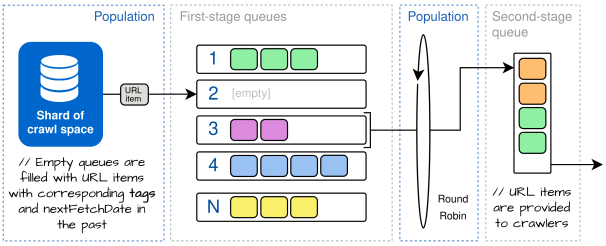



Figure 3: Population of frontier data structure

The URL Frontier service constantly queries the storage backend to populate its internal frontier data structure (see Figure 3), while simultaneously ingesting incoming URLs. A service can define an "interest" using a signature of tags, which limits the set of web resources it fetches from the backend. This allows for focused crawling within the overall general crawling system. For example, some crawlers may only be interested in sitemaps of family-friendly websites, so the corresponding URL Frontier service only retrieves web links tagged as `Sitemap` and `FamilyFriendly`.

To ensure strict politeness and high throughput, the frontier's internal data structure uses two stages of queues. The first-stage queue is determined by the *queue ID*, which is based on the hash of the Paid-Level Domain (PLD). Web resources with the same PLD (shown as the same color in Figure 3) are always kept in the same queue. Empty first-stage queues are refilled to ensure a constant supply of links to be fetched. In addition to matching the queue ID and the scope of interest (represented by required tags), URL items retrieved from the crawl space must have a `nextFetchDate` in the past. This simple yet effective approach enforces a time interval between two fetches of the same link.

The second-stage queue (or buffer queue) collects and sends URL items to crawlers requesting new fetch tasks. It is populated by iterating over the first-stage queues in a Round Robin manner. A delay between each traversal round prevents the buffer queue from being populated too frequently with URL items having the same queue ID (and possibly the same domain). OWLer defaults to a minimum five-minute delay between subsequent rounds of buffer queue population, and for each first-stage queue, only the first ten items are popped and added to the buffer queue. The buffer queue has a maximum capacity smaller than the number of first-stage queues, ensuring that a crawler working in a streaming manner will not visit pages with the same Paid-Level Domain more than 20 times in five minutes. Each crawler also applies a default crawl delay of 1 second (or as specified in the robots.txt file) to avoid overwhelming a web server at any given moment.

### *Persistence*

The storage backend choice is crucial for the system's overall performance, as it is the main factor impacting the latency of GetURLs and PutURLs requests. OWLer initially relied on the open-source Search & Analytics Platform OpenSearch,[7] which had certain drawbacks. An interface was added to allow substituting the concrete backend implementation, further modularizing the URL Frontier software and minimizing dependency on a single technology.

The persistence layer stores the crawl space on long-term memory and constantly provides the previously persisted URL items to be fetched next. The database system must accommodate a high number of read operations for populating first-stage queues and read-insert operations for updating the crawl space (tens of thousands per second). Any storage backend must be optimized for two commands: data querying (DQ) and data manipulation (DM), resulting in a tight data model. This data modeling step is implemented slightly differently in every database system, but it is key to achieve low request latency and high crawling throughput. Beyond data modeling, query routing and data locality significantly impact performance in a distributed setup. The partition key, previously mentioned as part of the URL item, has proven useful for effectively spreading the crawl space over tables and machines, enabling efficient data querying. However, these aspects are not discussed in more detail as they are core concepts of any distributed system and can be successfully managed by the concrete DBMS.

## OWLER IN ACTION

This section shares insights and results obtained with the current, preliminary crawling system setup. The past nine months have been a phase of iterative development, during which OWLer was set up and integrated with the downstream Open Web Indexing pipeline. During this time, the deployment was not fully permanent but interrupted by several breaks used to eliminate deficiencies in politeness, performance, and robustness discovered along the way. Nevertheless, within these nine months, OWLer discovered 10.2B web links that are persisted on the current OpenSearch-based storage backend. 1.17B of them have been visited at least once by one of the StormCrawler-based agents. The visited URLs are distributed over 37.3M hosts, dispersed over different topical and geographical domains of the web. As some web pages have been recrawled, OWLer has processed a total of 3.50B URLs, with an 88% successful fetch rate. This leads to a total of around 3.08B web documents provided to the indexing pipeline.

In addition to crawling, OWLer has ingested parts of publicly available dumps of Common Crawl to fill our crawl space with a broad range of seed URLs and provide a continuous output of crawled web documents despite the discontinuous deployment of OWLer in the early phases. The result is approximately 150 TiB of mostly HTML web documents,

[7] `https://opensearch.org`

https://doi.org/10.5281/zenodo.13863478



compressed and archived in WARC file format. All data was made publicly available as compressed Parquet and CIFF [8] files encoding the Open Web Index as inverted files with complementary metadata information.

During a four-week period of consistently high performance, a setup of six crawlers, two URL Frontier services, and one OpenSearch node achieved around 36M to 42M visits per day (equivalent to over 1 TB of WARC files per day). This means between 6M and 7M visits per crawling node and around 75 URLs per second per node. Ongoing experiments indicate the potential for further significant performance increases. Software and infrastructure engineering efforts in enhancing the processing pipeline, improving the backend data model, and extending the OpenSearch cluster suggest an increase in throughput by a factor of two to three in a similar but more robust setup. More advanced crawling tools manage up to 1000 URLs per second, and according to our tests, a URL Frontier instance with one OpenSearch node can consistently provide enough URLs to supply it. As a next step in order to meet OWI's aspiration, the system needs to scale horizontally by employing a multi-node database cluster.

## CONCLUSION

This report introduces the Open Web Crawler (OWLer), a crucial component of the Open Web Index (OWI) development project. The OWI aims to promote open access to web data and encourage innovation in web search technologies. By providing a publicly funded, legally compliant, and open-source alternative to web indices of commercial search engines, the OWI challenges their current dominance and strengthens the community-driven collection and processing of web data.

OWLer's modular and scalable architecture is designed to handle the diversity and geographic distribution of European compute resources effectively. This design enables the system to manage the large-scale, continuous crawling required to capture a comprehensive snapshot of the web. By integrating open-source technologies such as StormCrawler, OpenSearch, and URLFrontier, OWLer demonstrates a commitment to utilizing and contributing to the open-source community. The tier architecture consisting of crawler, frontier and persistence layer modularizes the software project. The URL Frontier services take the central position within this architecture and define interfaces towards crawlers and the distributed storage system.

Throughout the project, significant technical challenges, primarily related to achieving efficient distribution and robustness in data handling, have been addressed. These efforts have already resulted in the collection of billions of URLs, demonstrating the system's ability to handle web-scale data. Future development of OWLer will focus on improving performance and expanding crawling capabilities. By continuously refining the system, the project aims to make even more substantial contributions to the open web ecosystem, facilitating the creation of innovative applications and services that leverage the vast amounts of data processed by OWLer.

[8] CIFF denotes the Common Index File Format; for more details, see [7]

## ACKNOWLEDGEMENT

Bundesministerium für Bildung und Forschung

This work is part of OpenWebSearch.eu, funded by the EU under GA 101070014, and part of CAROLL, funded by the German Federal Ministry of Education and Research (BMBF) under 01|S20049.

## REFERENCES

[1] P. Boldi, A. Marino, M. Santini, S. Vigna, *BUbiNG: Massive Crawling for the Masses*, ACM Trans. Web 12 (2), May 2018.

[2] S. Chakrabarti, M. van den Berg, B. Dom, *Focused crawling: a new approach to topic-specific Web resource discovery*, Computer Networks, Volume 31, Issues 11–16, 1999, pp. 1389–1640.

[3] M. Charikar, *Similarity Estimation Techniques from Rounding Algorithms*, Proc. of the 34th Annual ACM Symposium on Theory of Computing, 2002, pp. 380–388.

[4] M. Dinzinger et al, *OWLer: Preliminary results for building a Collaborative Open Web Crawler*, Proc. of the 5th International Open Search Symposium (OSSYM), Oct. 2023.

[5] M. Granitzer et al, *Impact and development of an Open Web Index for open web search*, Journal of the Association for Information Science and Technology, Aug. 2023.

[6] G. Hendriksen et al, *The Open Web Index: Crawling and Indexing the Web for Public Use*, Advances in Information Retrieval (ECIR 2024), Mar. 2024.

[7] D. Hiemstra et al, *Challenges of Index Exchange for Search Engine Interoperability*, Proc. of the 5th International Open Search Symposium (OSSYM), Oct. 2023.

[8] R. Khare, D. Cutting, K. Sitaker, A. Rifkin, *Nutch: A Flexible and Scalable Open-Source Web Search Engine*, 2005.

[9] H. Lee, D. Leonard, X. Wang, D. Loguinov, *IRLbot: Scaling to 6 Billion Pages and Beyond*, Proc. of the 17th International Conference on World Wide Web, 2008, pp. 427–436.

[10] G. S. Manku, A. Jain, A. Das Sarma, *Detecting Near-Duplicates for Web Crawling*, Proc. of the 16th International Conference on World Wide Web, 2007, pp. 141–150.

[11] C. Manning, P. Raghavan, H. Schütze, *Web crawling and indexes* (Chapter 20), In: *Introduction to Information Retrieval*, Cambridge University Press, 2008.

[12] G. Mohr, M. Kimpton, M. Stack, I. Ranitovic, *Introduction to Heritrix, an archival quality web crawler*, Proc. of the 4th International Web Archiving Workshop (IWAW'04), Jul. 2004.

[13] M. Najork, A. Heydon, *High-Performance Web Crawling*, Handbook of Massive Data Sets. Massive Computing (4), Springer, 2002.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13872163

# FEDERATED DATA INFRASTRUCTURE FOR THE OPEN WEB SEARCH*

Noor A. Fathima[†1], C. Ariyo[2], M. Dinzinger[6], M. Golasowski[5], M. Hayek[4], G. Hendriksen[7], M. Karlsson[2], K. Mankinen[2], S. Moiras[1], J. Truckenbrodt[3], L. Vojacek[5], S. Hachinger[4], J. Martinovič[5], M. Granitzer[6], A. Wagner[1]

[1] CERN, Geneva, Switzerland
[2] CSC - IT Centre for Science, Espoo, Finland
[3] DLR, Berlin, Germany
[4] LRZ, Munich, Germany
[5] IT4I, Ostrava, Czech Republic
[7] Radboud University, Nijmegen, The Netherlands
[6] University of Passau, Passau, Germany

*Abstract*

The World Wide Web, as an indispensable technological construct and infrastructure, catalyses innovation and underpins the foundation of our digital economy. It provides a comprehensive technological scaffold for a myriad of services, advancements, and enterprises, and is a pivotal data reservoir for researchers, creators, commercial entities, and the wider society. The initiative Open Web Search.eu [1], funded by the Horizon Europe programme [2], is designed to harness the untapped potential of the Web as a data repository and to construct a scalable and open *European open Web search and analysis infrastructure*. This infrastructure, in the absence of any comparable alternative, is predominantly under the dominion of a select cadre of gatekeepers who currently constitute an oligopoly, thereby monopolising search functionality and, by extension, the accessibility to information residing on the Web [3]. The pilot version is the *open web search federated data infrastructure*, the technological complexities of which we elucidate in this document, given that it represents the midpoint in the timeline of this initiative. It embodies the idea of a confluence of cloud, high-performance computing, big data, providing a foundation for the development of new search applications and the training of AI models. This manuscript elucidates the cutting-edge computational, storage, and interconnection technologies that serve as the backbone of this infrastructure and critically examines the deployment strategies employed in the integration of these technologies. In pursuit of the objective for this infrastructure to accommodate petabyte-scale operations upon the culmination of the project's timeline, we concurrently present a concise overview of the technological roadmap devised to actualise our performance benchmarks.

## INTRODUCTION

The confluence of cloud computing, High Performance Computing (HPC), and big data is a crucial component of the OpenWebSearch.eu's (OWS) strategy to develop a flexible and inclusive *European open web search data and analysis infrastructure*, which is inspired by Lewandowski's idea of an "open web index" [4] and the corresponding core principles [5]. This effort is carried out through a well-thought-out research and innovation strategy aimed at meeting the needs of diverse scientific communities and various public and private stakeholders interested in driving significant digital advancements using web data. These web data, considered big data due to their large volume, [6] are crawled and stored in a distributed manner, yet accessible within a unified view, and processed through independent serial/parallel computing tasks on the data segments. We aim to crawl 30–50 percent of publicly indexable web content within 3 years timeline of the project. Crawling operations are performed on cloud computing platforms that offer convenient access to computing resources typical of data centres. The subsequent preprocessing and indexing of these crawled data are categorised as big data processing tasks, ideally suited for execution on HPC systems. Currently, OWS is collaborating with 4 HPC partners (who also serve as cloud partners) and 1 cloud partner. Our collaborative efforts are primarily focused on establishing a scalable pipeline for crawling, preprocessing, and indexing on modern heterogeneous architectures. Additionally, we aim to provide easy access to the index and other datasets through various interfaces, which form the basis for developing search applications and training artificial intelligence models. The infrastructure is designed in a modular manner to enhance the efficiency of the pipeline and abstract users from its technical intricacies. By integrating HPC technologies with cloud computing and big data using user-friendly interfaces, we are working toward a practical solution that demonstrates the convergence of these technologies in a real-world scenario.

## INFRASTRUCTURE ARCHITECTURE ELEMENTS

Figure 1 shows the structured layout of the OWS Federated Data Infrastructure (OWS-FDI), designed to oversee data aggregation, processing, indexing and distribution across

* This project is funded by the European Commission under the grant agreement GA 101070014 within the Horizon Europe Framework Programme
† noor.afshan.fathima@cern.ch

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13872163



various distributed computing and storage facilities. The architecture is divided into five clear and organised layers:

- Layer 1, enclosed in green boxes, are collectively known as *Interfaces*.
- Layer 2, represented by a purple box, is dedicated to the authentication and authorisation infrastructure, referred as *AAI layer*.
- Layer 3 encompasses the *cracking, pre-processing and index generation and the index post-processing layer*, contained within the ochre boxes.
- Layer 4, enclosed in a yellow box, comprises the *Computer Infrastructure layer*.
- *The OWS Data Distribution Layer*, visualised in a cyan box at the bottom, manages the distribution of data throughout the network.

Additionally, there is the

- *Single Sign-On (SSO)* functionality, which interfaces with most components to simplify user authentication and system entry.
- *Logging and Monitoring Service*, with the self-explanatory name.
- The two back-end elements, namely the *Crawling Queue* and *Compute workflow orchestration engine*, are explained with reference to the corresponding layers.

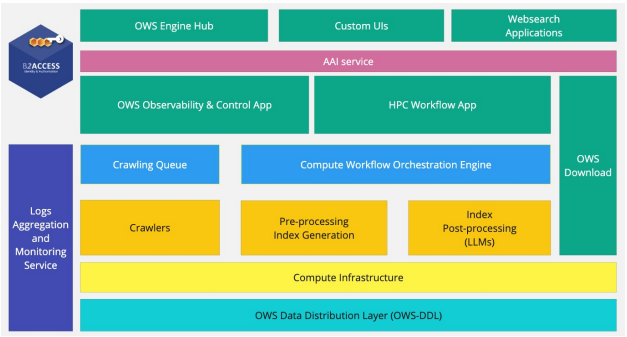



Figure 1: Layered Federated Data Infrastructure

A few months after the project kicked off, web crawling operations on the nascent OWS-FDI were started. As of March 2024, Table 1 provides a detailed overview of essential metrics following these operations. This table presents updated numbers, approximately one month subsequent to the exposition of a comparable table in the referenced deliverable [7].

Table 1: Crawling, Preprocessing and Indexing statistics as of March 2024

| | |
|---|---|
| No. URLs discovered | ca. 9.8 billion |
| No. of web pages crawled | 1.23 billion |
| No. of Hosts Accessed | 29 million |
| Crawling queue data volume | 4.7 TiB |
| WARC data volume | 96 TiB |
| Preprocessed data volume | 10 TiB |
| Index data volume | 2.4 TiB |
| Logs and Metrics data volume | 2.3 TiB |

In this section, we will outline the service/components within each layer defined; In the subsequent section, the current status of implementation and deployment, as well as the prospective future plans for each service are described.

### *Single Sign On (SSO) feature via B2ACCESS*

In the OWS-FDI framework, we adopted EUDAT B2ACCESS [8] for single-sign-on (SSO) capabilities, as it adheres to security standards and supports multiple authentication options such as eduGAIN [9] and ORCiD [10]. This system, integral to EUDAT [11] Collaborative Data Infrastructure (EUDAT CDI [12]), operates through an OpenID Connect (OIDC [13]) server and acts as both an OpenID Provider (OP) and a connector for various Identity Providers (IdPs). The B2ACCESS proxy model translates credentials and manages trust and authorisation policies, incorporating TLS and mutual client authentication with X.509 certificates. Positioned as the main gateway to log into the federated layered data infrastructure (Figure 1), B2ACCESS is envisioned to extend its functionality to all public interfaces detailed in interfaces-section and developer-specific interfaces, such as log into individual iRODS zones [14], ensuring security and regulatory compliance throughout OWS-DDL.

### *Layer 1: Interfaces*

**OWS Engine Hub** The Open Web Index (OWI) and the OpenWebSearch Engine Hub (OWSE-HUB) constitute the two main products of the OWS project. OWSE-HUB, conceptualised similar to Docker Hub [15] but for search engines, is a web-based GUI offering various search engine stacks. This setup is designed to efficiently expedite the creation of new search categories. It is configured to interface with the OWS-DDL, as depicted in Figure 1, helping to retrieve the index specifications. More details are available in the following deliverable [16].

**Custom UIs** This interface layer targets specialised applications, like web analytics, that use web data for various scenarios beyond simple search functions. The key to any custom user interface application is authentication, which is essential for accessing data. Data access is offered in two primary forms:

- Download: Users can download data sets or files organised by the HPC Workflow App for local use. This method adapts to the client's storage technology, but limits data selection to partitions or files, requiring clients to perform further data filtering.
- Offload: Through OWS Observability Control App, authenticated users can set up and schedule data transfers to different storage technologies. This option allows for more extensive filtering and is more bandwidth-efficient, suitable for clients with limited storage.

The project aims to facilitate the access to data for third-party developers to create applications using these resources, enhancing the data according to their needs.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13872163

**Web search applications** The OWS-FDI was established to streamline the development of specialised search applications using the OWI, differing from broad search engines like Google [17] or Bing [18], which focus on general topics. This framework supports the creation of vertical search engines that offer targeted search and retrieval options. Partners working on this aim to provide guidance, technical documentation, and a prototype application [19] created to access the OWI and send queries to it through a REST API based on a modular and configurable system depending on purpose and needs; this has facilitated the development of two specific search applications [20]. This initiative is part of the broader OWS goal of fostering a network of niche search engines and related applications, with the OWS-FDI hosting and providing access to these resources.

**OWS Observability and Control App** This is a software tool designed for system monitoring and management, providing comprehensive analysis and operational oversight. Designed as a single-page web application [21], it uses various cloud services, with a functional summary displayed in Figure 2. It requires B2ACCESS account authentication and incorporates essential features as outlined below.

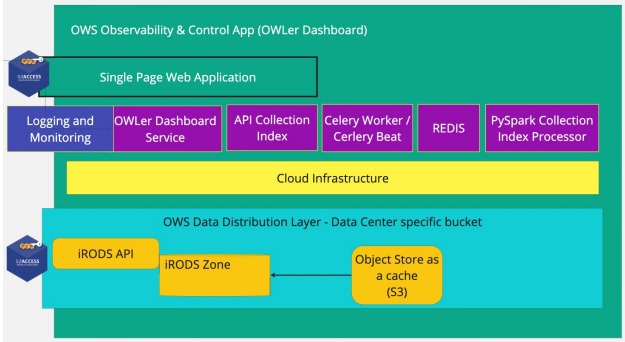



Figure 2: Observability and Control App

- *Crawling Metrics Dashboard*: Termed as the OWLer Dashboard service, it tracks and analyses metrics like crawled pages, pre-processed documents, and indexed data over various timeframes. It provides aggregated statistics for websites and hosts, in addition to live tracking of crawling activities. More information is available in the following deliverable [34].
- *Collection Index API*: Facilitates access to statistics for primary and specific subindices. Users can define collection indices and transfer data to storage systems such as OpenSearch [22], Elasticsearch [23], and S3 [24] using off-load filters.
- *Worker Components*: Executes user-defined tasks using nearby cloud infrastructure to minimise network costs. Data centres host worker groups that synchronise with the API through a REDIS [25] message queue, using a PySpark [26]-based library, the Collection Index Processor.

Interfaces to other Components

- Most functions necessitate authentication through B2ACCESS.
- An API offers programmatic access to the various functions and data elements.
- The application can access the crawling queue through the metrics server and the OWS-DDL to collect statistics and perform different operations (e.g. data offloading).

**Compute workflow orchestration engine** The backend of HPC Workflow App, as shown in Figure 1, optimises the HPC workflows for OWS by managing the pre-processing and indexing of the crawled data. Initially reliant on traditional batch scheduling for resource allocation, it now incorporates the compute workflow orchestration engine of the LEXIS [27] framework, addressing the diversity, scale and geographical spread of OWS-FDI [28]. This engine also interfaces with the batch scheduler and AAI for security, enhancing overall efficiency. It manages data inconsistencies via OWS-DDL and handles software provisioning, VM creation, and task distribution across the compute infrastructure (Layer 4), coordinating with HEAppE [29] middleware for secure access to HPC/cloud infrastructures.

The first paper is ready to be submitted, if you have time today , could you please skim through it? otherwise it's ok, I will go ahead and submit it.

**HPC Workflow App** The GUI facilitates interaction with the Compute workflow orchestration engine, enhancing workflow management by organising them as directed acyclic graphs (DAGs). It supports creating, viewing, and modifying datasets, managing metadata, and performing file operations within dataset directories. Integrated with the OWS-DDL, it improves data discoverability and replication across multiple supercomputing centres, accommodating specific requirements like GPU usage. This component is part of the LEXIS framework.

**OWS Download** This component, as illustrated in Figure 1, simplifies access to key datasets after preprocessing and indexing. Users can download these through HPC Workflow app. The OWS-DDL Dataset Listing [30] feature provides a summary of the datasets available in OWS-DDL, including metadata, all stored in the platform.

### *Layer 2: Authentication and Authorisation Infrastructure Layer (AAI)*

The OWS-FDI features a network of services that supports various user roles and domains, as depicted in Figure 1. These services, which include user interfaces and back-end support, such as OWS-DDL, can function independently and collaboratively. Integration with AAI, facilitated by the LEXIS framework and Keycloak [31], a single sign-on (SSO) identity and access management (IAM) solution, allows seamless access to all platform services. Keycloak integration with B2Access streamlines access, enhances security, and improves user experience. It supports OIDC,

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13872163



REST interfaces and JWT tokens for service interaction, and uses role-based access control (RBAC) or attribute-based access control (ABAC). The decentralised AAI approach enhances system resilience, with HEAppE middleware acting as a crucial integration layer with local HPC centre's AAI solutions.

### *Layer 3: Crawling, Pre-processing, Index generation*

**Crawling queue** This component monitors the status and sequencing of crawled and pending URLs, access statistics, and cache digest. The Frontier queue, depicted in Figure 3, stores URLs discovered or visited during crawls. As summarised in Table 1, the crawling queue contains 4.7 TiB of stored URLs and contains a total of 9.8 billion discovered URLs.

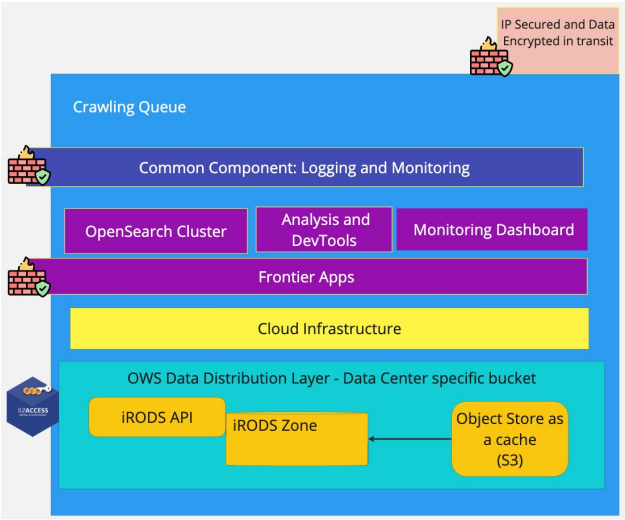



Figure 3: Crawling Queue Service

The crawling queue consists of the following components, as depicted in Figure 3:

- *Frontier applications*, connected to the opensearch cluster, centralises and coordinates crawling priorities among data centres, using the NGI-funded URL Frontier [33], enhanced for dynamic partitioning. These applications interact with Logging and Monitoring service. For further information, consulting the following deliverable [34] is recommended.
- *The OpenSearch Cluster* serves as the back-end for the frontier apps and monitors operational metrics. OpenSearch integrates with Grafana [35] for real-time cluster status and provides dashboards for data visualisation and cluster management.

Access to these services is secured through IP addresses that are specifically whitelisted and restricted to service participants in partner computing centres.

**Crawlers** Figure 4 shows the components of the Crawler service. Automated crawlers collect data from the Internet, generating WARC [36] files (collections of HTTP data streams from web crawls), stored in the OWS-DDL's object store for shared access. [37] As summarised in Table 1, the aggregate amount of WARC data is 96TiB. This sophisticated system is based on the StormCrawler [38] platform within an Apache Storm [39] cluster. It includes three primary pipelines:

- WARC2WARC for integrating external WARC files;
- Exploratory crawling for regular web crawls; and
- Sitemap crawling, which uses the Sitemap protocol.

The crawlers operate in various data centres, communicating with the crawling queue via frontier apps and Logging and Monitoring service, as illustrated in Figure 4.

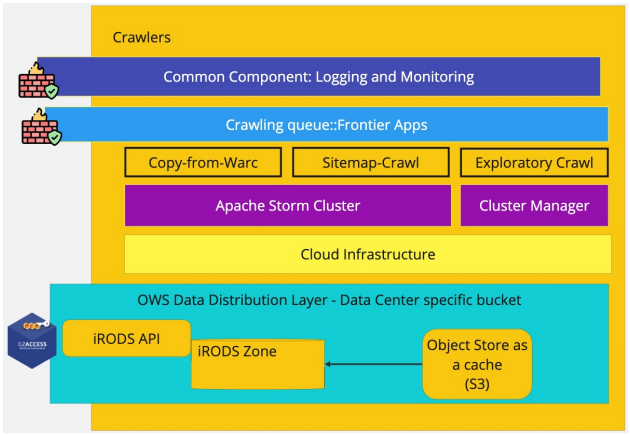



Figure 4: Crawlers

For additional details, it is recommended to refer to the deliverable cited [34].

**Pre-processing and Index generation** The pre-processing is divided into two main tasks, as depicted in Figure 5 and described below.

- *Pre-processing*: This phase focusses on enrichment and content analysis. It starts with retrieving WARC files from the local object store in OWS-DDL, populated by crawlers. The cleaned HTML and metadata are extracted from these files and stored separately within the same OWS-DDL framework. The metadata from this stage are saved in Apache Parquet [40] format.
- *Evaluation of the pre-processing plugin*: This task involves evaluating plugins for the content analysis library. These plugins, developed by project members or external contributors, enhance the enrichment process. Using Apache Spark [41] batch jobs, leveraging Resiliparse [42] to parse and clean HTML content, and implementing various metadata enrichments.

The index generation process, a continuation of preprocessing, transforms the cleaned and enriched data into a practical index, often in the form of an inverted file. This transformation involves segmenting the indexes into 'shards' based on specific metadata categories such as topic and language, identified during preprocessing. These shards are then formatted and distributed in the Common Index File Format (CIFF). Similarly to the pre-processing phase, index-

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13872163

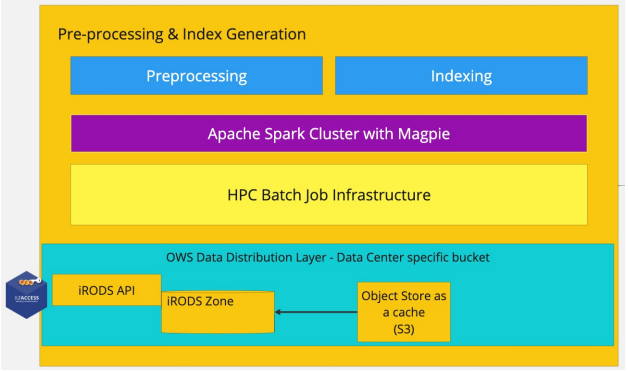



Figure 5: Preprocessing and Indexing

ing is executed through an Apache Spark batch operation, enabling the creation of semantically cohesive shards of the complete web index. This method leverages the metadata obtained during preprocessing to facilitate the development of specialised search engines tailored for distinct purposes like different languages, regions, or specific topics such as news and sports. For additional details, refer to the deliverable cited here [43].

As summarised in Table 1, the aggregate amount of preprocessed data is 10TiB and the size of the index is 2.4TiB.

**Index post-processing (LLMs)** Once the index is created, a post-processing phase is implemented, as shown in Figure 1. This stage may involve training extensive language models (LLMs) to analyse or categorise the content. These post-processing tools, developed from initial indexing results and potentially using LLMs trained on selected web content subsets, are not yet available. However, as the data become openly accessible to both the project consortium and the broader community, new applications and methodologies are expected to emerge, improving the utility of the indexed data.

## *Layer 4: Compute infrastructure and storage*

Infrastructure providers offer vital computing and storage resources for the project, including traditional High-Performance Computing (HPC) and Infrastructure-as-a-Service cloud (IaaS-cloud) solutions. The project leverages a combination of large-scale cloud computing resources and HPCs distributed in various locations, integrated with some of the top 20 supercomputers worldwide [44]. These computing resources form the backbone of all OWS-FDI operations. As depicted in Figure 1, the HPC/Cloud infrastructure and storage layer enable the interplay between HPC and cloud systems, supplying the necessary computational strength and storage space. All facilities comply with ISO 27001 [45] standards, ensuring that IT security management is in accordance with this framework.

**The confluence of Cloud and HPC in OWS-FDI** Some partners provide Infrastructure-as-a-Service (IaaS) platforms featuring on-premise cloud solutions for on-demand provisioning of high-performance virtual machines and bare metal machines. These support multiple CPU cores, RAM, and run services in different zones. The machines are equipped with HDD/SSD and scalable S3 cloud storage clusters, optimised for tasks like crawler applications via Kubernetes [46], and built on OpenStack [47] or VMWare [48]-based IaaS clouds. This setup combines cloud computing advantages with physical infrastructure control, supporting the deployment and management of scalable applications. Storage clusters, secured with TLS for API communications, are accessible through machines or containers. This cloud ecosystem complements the HPC infrastructure, merging cloud and supercomputing capabilities to efficiently manage large volumes of crawled data. Security is managed by internal teams, with access to the infrastructure restricted to specific IP addresses, which limits external access from other services or centres.

## *Logging and Monitoring Service*

This module collects logs and metrics from all other modules and interfaces with OWS Observability and Control App, as shown in Figure 1, making public log data and metrics accessible. As summarised in Table 1, the aggregate amount of log data is 2.3 TiB. This approach improves transparency, increases community participation, and helps collaborative problem solving. Public logs are especially valuable for open source projects or public services, as they establish credibility by displaying real-time system performance and serve as an educational resource for a deeper understanding of the system. They also comply with regulatory mandates for operational transparency. Care is taken to preserve these public logs to ensure security and privacy standards.

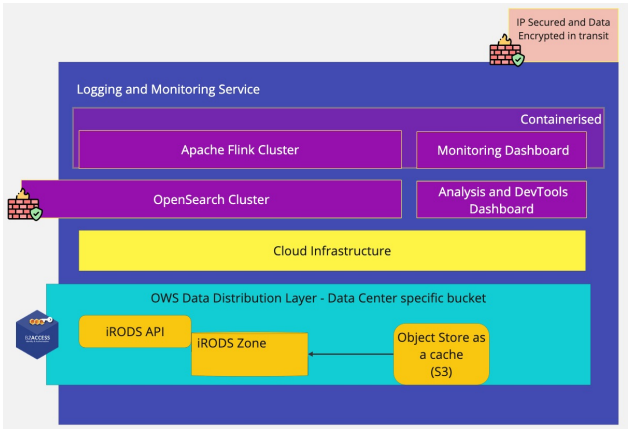



Figure 6: Logging and Monitoring Service

In the upper layers of Figure 6, the central component is an Apache Flink [49] Cluster, a monitored data streaming platform that works in conjunction with the OpenSearch Cluster. These layers interact with OWS-DDL, granting access to logs and metrics from different components. OpenSearch dashboards [50], serving as analysis and development interfaces, provide visual representations of OpenSearch data

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13872163



and aid in managing and expanding the OpenSearch cluster, benefiting both development and operational activities.

### *OWS Data Distribution Layer (OWS-DDL)*

The OWS-FDI uses a distributed architecture for data computation and storage. Figure 7 shows the distributed storage. This layer employs geodistributed storage and mirroring to ensure data redundancy and safety, although mirroring may not always be required. Using the Integrated Rule-Orientated Data System (iRODS) and EUDAT-B2SAFE, each site corresponds to an iRODS zone, illustrated in Figure 7. Data storage and management are spread across 5 participating data centres; for the sake of illustration, we have used 3 of them named A, B, and C, each equipped with local storage compatible with S3 and a corresponding iRODS zone. These zones are interconnected, facilitating data transfer and sharing. iRODS handles metadata for datasets in an iCAT [51] metadata catalogue, allowing a unified view of data across all sites. The access privileges are managed using the iRODS mechanisms, supported by AAI for SSO. This layer oversees various data sets and facilitates secure, location-independent data retrieval, and integration with other OWS-FDI components through REST APIs for data exchange. Its objective is to provide seamless cross-site access to datasets, making them available as part of a unified file system in different back-end data systems [52].

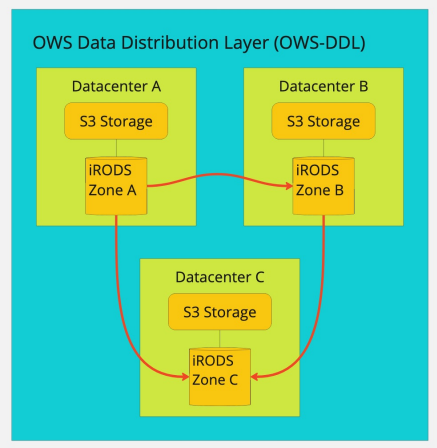



Figure 7: OWS Data Distribution Layer

## IMPLEMENTATION STATUS AND ROADMAP

This segment provides a comprehensive update on the current implementation and deployment status of each specific component, together with the strategic plan established for the remaining duration of the project. Unlike the previous section, where components were categorised by layers, this section will individually discuss each component to provide a more detailed overview. Overall, all components are set up and ready for testing, which includes:

- enhancing scalability,
- extending to more data centers, and
- adding extra functions/features.

### *Single Sign On (SSO) feature via B2ACCESS*

B2ACCESS currently serves as the login gateway for OWS Observability and Control App and operates as one of the Identity Providers (IdP) integrated into the Keycloak interface of AAI. Its integration with the iRODS zone at a partner data centre site is underway, and other sites are expected to follow suit over the next year. Furthermore, efforts to incorporate B2ACCESS for authentication between different components within interface layers, as described in previous sections and illustrated in Figure 1, will begin in the coming year.

**OWS Engine Hub** The OWSE-HUB initiative has begun with steps such as incorporating the indexer and the CIFF standard into TIRA [53]. This integration has provided insight on the use of pipeline indexes on a central platform, guiding future plans. Detailed plans for OWSE-HUB will be developed, focussing on defining the search engine's formulation and presentation. The 'hub' will be configured to distribute these specifications, as shown in Figure 1. The aim is to launch the OWSE-HUB's initial version by the end of the second year.

**Web search applications** The prototype application along with a pair of use case applications are currently housed on-site at the developer partner locations. Planning is in place for their transition to the infrastructure provided by the cloud service partners. Close work is ongoing with developer partners themselves, who have experience in search applications and user experience, to design deployment strategies that make the best use of resources. Our goal is to deploy these applications in a way that is efficient, secure, and publicly accessible, as illustrated in Figure 1.

**OWS Observability and Control App** This part includes the OWLER Dashboard and the Collection Index API, illustrated in Figure 2, both of which are currently in the initial development stages. The development partners aim to improve user-friendliness and collaborate with external parties to use the available data.

**HPC Workflow App and Compute workflow orchestration engine** The sequential processes of Preprocessing and Indexing were previously managed by a batch scheduler on different computing nodes, accessed mainly through SSH and a CLI on the front-end node of an HPC cluster. Now, as shown in Figure 2, the system is becoming automated with the HPC Workflow App as the front-end interface and the compute workflow orchestration engine as the back-end.

This transition leverages the advanced execution environment offered by HPC partners within the LEXIS framework, providing access to supercomputers and cloud resources. This integration improves the management and transfer of

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13872163

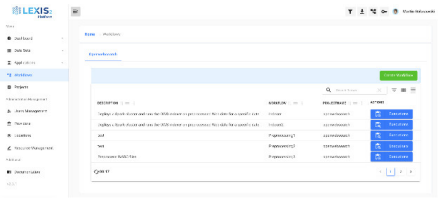


Figure 8: The Lexis Platform

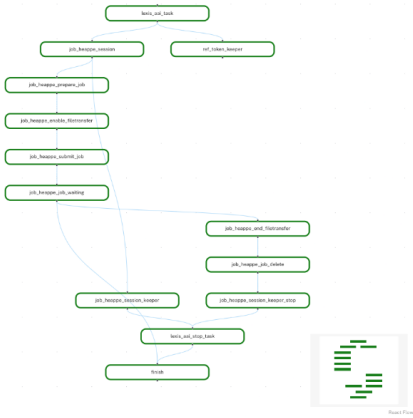


Figure 9: Workflow execution organized and displayed as Directed Acyclic Graphs (DAGs)

large data sets, particularly in relation to Figure 7. The HPC Workflow App initiates preprocessing and indexing tasks, triggering the respective scripts. The variety of workflows operating on the LEXIS platform is depicted in Figure 8. These processes, structured as Directed Acyclic Graphs (DAGs), are shown in Figure 9. For both workflows, an Apache Spark cluster is established in the HPC environment, and a Spark job is submitted to index the preprocessed data for a specific day (as defined in LEXIS).

The pipeline is operable through the HPC Workflow app and can interface with S3 storage. The incorporation of the LEXIS Data Staging [54] and Data Set API is being implemented. This integration is a focus, aiming to allow developers to divide tasks into executable blocks within the LEXIS framework. Plans also include monitoring distributed data and interim results through the HPC Workflow app, with B2ACCESS supporting secure authentication.

**Authentication and Authorisation Infrastructure (AAI) Layer** The LEXIS framework integrates HPC Workflow App, Compute Workflow Orchestration Engine, and OWS Download to improve user access via the AAI layer. This is depicted in the top three layers of Figure 1. Keycloak, the current AAI platform, handles the authentication process on the login page. B2Access, integrated as an IdP in Keycloak, enables users of the OWS consortium to authenticate with their institutional credentials, eliminating the need for multiple logins. Future enhancements aim to extend this access to more OWS-FDI components and OWS-DDL, further easing platform integration.

**Crawling Queue** The queue configuration, as shown in Figure 3, links two Frontier App instances to a separate instance that hosts a Back-End OpenSearch cluster. Logical volumes manage storage across physical hard disks. Data availability is ensured by external S3 clusters, and data in transit are protected by TLS encryption. OpenSearch oversees authentication, maintaining a database of user roles and hashed passwords. Basic-auth credentials are managed by partner sites. The current deployment is hosted on a partner cloud site, accessible via SSH.

Future plans include enhancing scalability for Frontier Apps by scaling up data nodes and incorporating a warm/hot storage mechanism. [55] Memory usage is optimised by adhering to the maximum standard limit per process. In the next stages, we plan to connect to OWS-DDL to share public logs and metrics, implement horizontal scaling for additional crawlers, and integrate with the open webmaster console [56].

**Crawlers** The crawlers implemented, shown in Figure 4, are integrated into the cloud infrastructure of three sites and are operational. The WARC files are stored in S3 buckets at each site for easy access by other components. Currently, crawlers are handling about 1.5TB of content daily, with plans to increase this to between 2 and 10 TB/day in the second year.

**Pre-processing and Indexing** Operational pipelines use the same infrastructure as crawlers, using HPC infrastructure. They align with the details in the previous section (Figure 5). Pre-processing and indexing were executed as Spark batch processes using the Magpie script collection. This method has been phased out as detailed in the HPC Workflow App and Compute Workflow Orchestration Engine sections. An instance of a TIRA platform is operational in a data centre.

Current efforts for Y2 focus on expanding the preprocessing/enrichment and indexing processes for crawled content. Functionality will be developed to estimate resources needed for processing a day's worth of WARC files, ensuring scalability with the volume of content crawled.

**The Log Aggregation and Metrics Service** The cloud-based service, as shown in Figure 6, collects log files from the crawler component (Figure 4) and monitors various components. All log data and metrics are stored in OpenSearch indexes.

OpenSearch is used to manage large data sets and aggregate logs. It uses a Dockerized Flink cluster, monitored by Dockerized Grafana as shown in Figure 6. It processes crawler logs in real time, extracts, combines, and writes them back into the OpenSearch cluster.

Cron jobs populate the blacklist index and periodic snapshots of OpenSearch clusters are stored in an external S3 cluster. In the second year, the plan is to scale the OpenSearch cluster, aggregate logs from other components through OWS-DDL, and identify and address any structural bottlenecks.

https://doi.org/10.5281/zenodo.13872163



### *OWS Data Distribution Layer (OWS-DDL)*

All partner data centres have implemented iRODS, enabling OWS-DDL. They can transfer data through iRODS zones, offering various storage options, including cloud CEPH [57] storage clusters, NFS-accessible storage, and S3 storage clusters. Currently, all HPC/data centres in OWS use S3 object storage for caching, with one using it for disaster recovery.

## CONCLUSION

The ambitious goal of processing 1 PB of raw web data in OWS, followed by its preprocessing and indexing for search applications, requires a sophisticated, user-friendly, and flexible infrastructure. Leveraging both cloud and HPC resources, a robust system has been developed to achieve this goal within the planned timeframe. OWS partners have effectively created and deployed a fully operational pilot Federated Data Infrastructure that supports the entire workflow from web crawling to preprocessing and indexing across various infrastructure partners. By March 20, 2024, the project had indexed approximately 1.23 billion web pages in 185 languages, utilising about 77 TiB of storage. In the project's final phase, the aim is to refine and expand the pilot OWS-FDI, extending it to additional consortium partners' sites to enhance resource utilisation. Concurrently, the performance and throughput of the existing layers depicted in Figure 1 will be assessed for scalability and potential improvements.

## ACKNOWLEDGEMENTS

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

## REFERENCES

[1] OpenWebSearch.eu, https://openwebsearch.eu/

[2] Horizon Europe, https://research-and-innovation.ec.europa.eu/funding/funding-opportunities/funding-programmes-and-open-calls/horizon-europe_en

[3] Smyrnaios, N. (2018). Internet oligopoly: The corporate takeover of our digital world. Emerald Group Publishing.

[4] Dirk Lewandowski. *The Web Is Missing an Essential Part of Infrastructure: An Open Web Index*, Commun. ACM 62(4), 24 (2019)

[5] Granitzer, Michael and Voigt, Stefan et al. "Impact and Development of an Open Web Index for Open Web Search", in Journal of the Association for Information Science and Technology. August. 2023, Wiley. https://asistdl.onlinelibrary.wiley.com/doi/10.1002/asi.24818

[6] Golasowski, Martin, and Martinovic, Jan et al. "Toward the Convergence of High-Performance Computing, Cloud, and Big Data Domains", in *HPC, Big Data, and AI Convergence Towards Exascale: Challenge and Vision*, O'Reilly, 2022, doi: 10.1201/9781003176664-1 isbn: 9781032009841, 9781032009919, 9781003176664

[7] Fathima, N. A., Wagner, A., Golasowski, M., Truckenbrodt, J., Mankinen, K., Hachinger, S., Granitzer, M. (2024). Launch of the Pilot Infrastructure. Zenodo., https://doi.org/10.5281/zenodo.10838954

[8] B2Access, https://eudat.eu/service-catalogue/b2access

[9] eduGAIN, https://edugain.org/

[10] ORCiD, https://orcid.org/

[11] EUDAT, https://www.eudat.eu/

[12] EUDAT CDI, https://eudat.eu/eudat-cdi

[13] OpenID Connect, https://openid.net/developers/how-connect-works/

[14] iRODS, https://irods.org/

[15] Docker Hub, https://hub.docker.com

[16] Hendriksen, G., Hiemstra, D., de Vries, A. P., Schmidt, S., Zelch, I., Potthast, M., Granitzer, M., Dinzinger, M., Zerhoudi, S., Fathima, N. A. (2023). The OpenWebSearch Hub and the Open Web Index. Zenodo., https://doi.org/10.5281/zenodo.10369512

[17] Google, https://www.google.com

[18] Bing, https://www.bing.com

[19] Mosaic search application, https://opencode.it4i.eu/openwebsearcheu-public/mosaic

[20] Open Science Search Application at CERN, https://open-science-search.docs.cern.ch/

[21] OWLer, https://openwebsearch.eu/owler/

[22] OpenSearch, https://opensearch.org/

[23] Elasticsearch, https://www.elastic.co/elasticsearch

[24] S3 Object storage, https://aws.amazon.com/s3/

[25] REDIS, https://redis.io/solutions/messaging/

[26] PySpark, https://spark.apache.org/docs/latest/api/python/index.html

[27] LEXIS, https://lexis-project.eu/web/high-performance-computing/

[28] Golasowski, Martin, and Martinovic, Jan et al. "The LEXIS Platform for Distributed Workflow Execution and Data Management" in *HPC, Big Data, and AI Convergence Towards Exascale: Challenge and Vision*, O'Reilly, 2022, doi:10.1201/9781003176664-2 isbn: 9781032009841, 9781032009919, 9781003176664

[29] HEAppE Middleware, https://opencode.it4i.eu/lexis-platform

[30] Lexis Dataset Listing, https://docs.lexis.tech/_pages/architecture/users_view.html#lexis-dataset-listing

[31] Keycloak, https://www.keycloak.org/

[32] OWLer Documentation, https://openwebsearch.eu/owler/

[33] URL Frontier https://nlnet.nl/project/URLFrontier/

https://doi.org/10.5281/zenodo.13872163

[34] Zerhoudi, S., Dinzinger, M., Schmidt, S., Hendriksen, G., Potthast, M., Fathima, N. A., Granitzer, M. (2023). The OpenWebSearch Crawler and the Crawling Frontier, https://doi.org/10.5281/zenodo.10355322

[35] Grafana, https://grafana.com/

[36] WARC File Format, https://en.wikipedia.org/wiki/WARC_(file_format)

[37] G. Hendriksen et al, *The Open Web Index: Crawling and Indexing the Web for Public Use*, Advances in Information Retrieval (ECIR 2024), March 2024.

[38] Storm Crawler, https://stormcrawler.apache.org/

[39] Apache Storm, https://storm.apache.org/

[40] Apache Parquet File Format, https://parquet.apache.org/

[41] Apache Spark, https://spark.apache.org/

[42] Resiliparse, https://resiliparse.chatnoir.eu/en/stable/

[43] Schmidt, S., Zelch, I., Bevendorff, J., Fröbe, M., Potthast, M., Granitzer, M. (2024). The OpenWebSearch WARC parsing content analysis library, https://zenodo.org/records/10838722

[44] Top supercomputers in the world, https://www.top500.org/lists/top500/

[45] ISO 27001, https://www.iso.org/standard/27001

[46] Kubernetes, https://kubernetes.io/

[47] OpenStack, https://www.openstack.org/

[48] VMWare, https://www.vmware.com/

[49] Apache Flink, https://flink.apache.org/

[50] Opensearch Dashboards, https://opensearch.org/docs/latest/

[51] ICAT Server, https://icatproject.org/user-documentation/icat-server/

[52] Munke, Johannes and Hayek, Mohamad et al, "Data System and Data Management in a Federation of HPC/Cloud Centers", in *HPC, Big Data, and AI Convergence Towards Exascale: Challenge and Vision*, O'Reilly, January 2022, doi:10.1201/9781003176664-4 isbn: 9781032009841, 9781032009919, 9780367766764

[53] TIRA Platform, https://www.tira.io/

[54] LEXIS Staging, https://docs.lexis.tech/_pages/data_system/staging_api.html

[55] Noor A. Fathima, M. Dinzinger et al, "Architecting the OpenSearch Service at CERN for OpenWebSearch.EU" Open Search Symposium (OSSYM), 2024, to be published

[56] Open Console, https://open-console.eu/index.html

[57] Ceph Storage System, https://ceph.io/en/

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13871889



# A DATASET OF GDPR COMPLIANT NER FOR PRIVACY POLICIES

Harshil Darji*, Stefan Becher, Jelena Mitrovic,
Armin Gerl, Michael Granitzer
University of Passau, Passau, Germany

*Abstract*

Privacy policies play a vital role in informing users about the data practices of online platforms. They are intended to help them make informed decisions regarding the processing of their personal information. Still, privacy policies are often long and complicated, making it difficult for users to understand how their data is being handled. Natural Language Processing (NLP) techniques, such as Named Entity Recognition (NER), can be employed to automatically extract meaningful information from privacy policies to ease the making of informed decisions. In this work, we present a dataset of privacy policies improved with NER annotations. The dataset consists of privacy policies from 44 online platforms. These policies were annotated to comply with the GDPR guidelines. The privacy policies are manually annotated with NER tags, highlighting relevant entities of GDPR privacy policies such as data controllers, data sources, authority, etc. We also provide the annotation guidelines used by the annotators. This annotated dataset is a valuable resource for training and evaluating NER models in the context of privacy policies.

## INTRODUCTION

In our digital age, data privacy has become a crucial issue due to the widespread use of online platforms [1]. To safeguard individual rights and ensure transparent data handling, regulatory frameworks such as the General Data Protection Regulation (GDPR)[1] have been implemented. An important GDPR compliance requirement is that organizations must provide concise, transparent, intelligible, and easily accessible privacy policies, using clear and plain language to inform users about the processing of their personal information. However, in reality, privacy policies are often extensive, complicated, and hard to understand [2], making it challenging for users to comprehend the data processing procedures. Thus, a gap between the regulatory requirements and the real-life implementation of privacy policies exists due to the necessity of presenting various and extensive information on the processing of personal data, which is clearly defined, while the communication and transparency requirements are only conceptually defined and, therefore, harder to implement. Therefore, there is a need for resources that can help bridge this gap and facilitate the understanding of privacy policies for non-expert readers.

The understanding of privacy policies is crucial to protecting personal information. Natural Language Processing techniques, especially Named Entity Recognition (NER), are instrumental in identifying entities within the text, such as *Data subjects* and *Personal Data entities* [3]. However, NER has limitations in revealing complex document relationships and structures, which are essential for a thorough comprehension of privacy policies.

The GDPR policies on the web require in-depth statistical analysis. This evaluation helps users identify trustworthy policies and express their preferences. One way to improve this assessment may be to incorporate Relationship Extraction (RE). It has the potential to provide an in-depth analysis of the links between recognized entities, which can fill in any gaps left by NER. This approach can offer users a complete perspective of privacy policies.

This paper aims to address this gap in research by introducing a GDPR-compliant privacy policy dataset that has been annotated with NER tags. The dataset comprises European privacy policies from various online platforms, annotated with NER tags to identify and highlight important entities within the policies, such as Data Controller, Data Processor, Data Source, etc.

The remainder of this paper is structured as follows: Section Related work studies the related research that displays the introduction and use of similar datasets. Section Dataset introduces the dataset in question and also provides some statistics related to this dataset.

## RELATED WORK

The study presented in [4] aims to provide insights into the techniques used for extracting information from textual documents and their applications by conducting a systematic mapping study on the automated analysis of privacy policies. The study analyzed 39 papers out of 1097 publications, identifying the potential for extracting individual pieces of information from privacy policies. The research addresses the growing demand for automated privacy policy analysis across various stakeholders as well as the importance of understanding privacy concerns and complying with relevant data protection laws.

The research [5] proposes PrivacyGLUE, the first benchmark for measuring general language understanding in the privacy language domain, especially focusing on privacy policies. According to this study, privacy policies need a separate benchmark due to their distinct language. PrivacyGLUE comprises seven tasks related to privacy policies and evaluates the performances of five transformer language models.

* Harshil.Darji@uni-passau.de

[1] https://gdpr-info.eu/

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13871889



In the domain of data privacy, numerous datasets related to privacy policies have surfaced, requiring further studies. [6] created the OPP-115 data set, which is a collection of 115 manually annotated privacy policies, in 2016. Due to its creation date, the privacy policies the data set is based on are not compliant with the GDPR. There have been attempts to map the OPP-115 categories to GDPR articles to modernize the OPP-115 data set [7]. While this can create GDPR-compliant labels, it does not affect the outdated privacy policies as the basis of the data set. [8] annotated 350 mobile app privacy policies with privacy practices, which form the APP-350 data set in 2019. It is used to check certain compliance issues, e.g., whether a privacy policy is present, but this is limited to the privacy policies of apps. [9] created a data set by collecting over one million privacy policies, which span more than two decades, based on more than 130000 websites. They discovered interesting changes in the policies over the years, like more self-regulation and especially the impact of the GDPR. While the publicly available corpus is a good basis for investigating long-term trends, it is missing annotation for NER. All of these data sets serve a certain purpose. But there is currently no up-to-date, i.e., GDPR-compliant data set with NER annotations for categorizing data handling practices in detail.

In the context of general legal text accessibility, [10, 11] introduced annotated German legal text corpora, addressing a scarcity similar to the one reported in GDPR-compliant privacy policies. [10] introduced two German legal text corpora, addressing the lack of annotated legal resources. The first corpus is a compilation of decisions from 131 German courts, while the second is an annotated subset tailored for machine learning applications in understanding Urteilsstil. Complementing this, [11] introduced a dataset of 2944 meticulously annotated German legal references, with 21 properties each, improving legal text analysis. Their work highlights the need for annotated datasets to enhance machine readability and user comprehension. This aligns with our efforts to improve the accessibility of privacy policies through named entity recognition (NER) annotations. It highlights a shared objective across different legal fields.

The potential for improving the accessibility and understanding of privacy policies through technology is also displayed in [12, 13]. The former focuses on a structured way to categorize and analyze web pages, including privacy policies. By effectively classifying web pages, this research aids in automatically identifying privacy policies across the internet. Such capabilities are crucial for ensuring compliance with data protection laws like the General Data Protection Regulation (GDPR), as they facilitate the automated extraction of relevant information from privacy policies, aiding both users and regulatory bodies in evaluating compliance. The latter focuses on developing the OWler web crawler, a significant step in improving web crawling efficiency by focusing on topic-based content discovery, including privacy policies. This approach simplifies the process of gathering privacy policies for further analysis.

## DATASET

The enactment of the GDPR in 2018 introduced stricter requirements for data privacy within the European Union. It gives users more control over their personal data by the introduction of Data Subject Rights [14, Art. 12 - Art. 23] and forced many service providers to rethink their handling of personal user data. The changes in the data handling practices directly led to a rework of existing privacy policies, in order to comply with the legal requirements of the GDPR for transparency [14, Art. 5]. This shift in the legal landscape created a research gap for a GDPR-compliant, NER-annotated data set of privacy policies because existing data sets, which were created before the enactment of the GDPR, are not applicable to the European Union anymore. We have shown, that up-to-date data sets are either missing NER tags or have another focus but web privacy policies. Therefore, we created a GDPR-compliant NER data set of web privacy policies to fix this gap.

Our data set consists of 44 European privacy policies, which have been manually annotated by legal experts. To create GDPR-compliant annotations, we have chosen the Data Privacy Vocabulary (DPV) [15], which represents the latest efforts to build a standardized ontology for privacy terms, as a basis. The DPV consists of several hierarchies, which focus on the handling of personal data as required by the GDPR, e.g., purposes, processing, or recipients. For the creation of our label set, we have chosen the most relevant entries of the DPV. Therefore, we compared several privacy policy languages, like SPECIAL [16], LPL [17], or JACPoL [18], and privacy preference languages, like YaPPL [19] or ConTra [20], in order to find a common basis of required elements. Privacy policy languages create machine-readable privacy policies, which can be further customized by the user. Privacy preference languages allow the user to define rules regarding these customization options. When a user has presented a privacy policy, represented by a privacy policy language, the preferences add support by automatically picking customization options or giving hints about mismatches. As this concept only works, if the privacy policy is machine-readable, we envision automatically translating plain-text privacy policies into such representations to enable preference matching.

Therefore, we added the following elements (based on their DPV notation), which were most commonly used in the languages we analyzed, to the label set: Data Controller (**DC**), Data Processor (**DP**), Data Protection Officer (**DPO**), Recipient (**R**), Third Party (**TP**), Authority (**A**), Data Subject (**DS**), Data Source (**DSO**), Required Purpose (**RP**), Not-Required Purpose (**NRP**), Processing (**P**), Personal Data (**PD**), Non-Personal Data (**NPD**). In addition, we analyzed the DPV for the most relevant legal terms with regard to the GDPR. Existing data sets often lack legal annotations, so with our intention to create a GDPR-compliant data set, this was an important step to take. Based on their DPV notation, the most important legal terms, regarding GDPR are Organisational Measure (**OM**), Technical Measure (**TM**),

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13871889



Legal Basis (**LB**), Consent (**CONS**), Contract (**CONT**), Legitimate Interest (**LI**), Automated Decision Making (**ADM**), Retention (**RET**), Scale EU (**SEU**), Scale Non-EU (**SNEU**), Right (**RI**), Lodge Complaint (**LC**). On top of these terms, we decided to individually add the most important Data Subject Rights as labels, because the GDPR requires them to be listed in the privacy policies. This further allows for an automated compliance check. Therefore, the final labels are Art. 15 Right to access by the data subject (**DSR15**), Art. 16 Right to rectification (**DSR16**), Art. 17 Right to erasure (**DSR17**), Art. 18 Right to restriction of processing (**DSR18**), Art. 19 Notification obligations (**DSR19**), Art. 20 Right to data portability (**DSR20**), Art. 21 Right to object (**DSR21**), Art. 22 Automated individual decision-making, including profiling (**DSR22**). This results in a total of 33 categories, which form our label set. The data set consists of 33 labels with the following distribution (see Figure 1). This figure demonstrates the overall token distribution with *I-* and *B-* annotations.

### *Annotation guidelines*

1. **Data Controller**: The individual or organization that decides (or controls) the purpose(s) of processing personal data. (E.g., *This document states the OpenStreetMap privacy policy for services formally operated and provided by the **OpenStreetMap Foundation (OSMF)**.*)

2. **Data Processor**: A *processor* means a natural or legal person, public authority, agency, or other body that processes personal data on behalf of the controller. (E.g., *We may share your data with **analytics providers**, which helps us understand how customers are using our services.*)

3. **Data Protection Officer**: An entity within or authorized by an organization to monitor internal compliance, inform and advise on data protection obligations, and act as a contact point for data subjects and the supervisory authority. (E.g., *A copy of these can be requested from the **Data Protection Officer**.*)

4. **Recipient**: A recipient of personal data can be used to indicate any entity that receives personal data. This can be a Third Party, Processor (GDPR), or Controller. (E.g., *The data collected on the systems will be accessible by the system administrators and the appropriate OSMF **working groups**.*)

5. **Third Party**: A *third party* means a natural or legal person, public authority, agency, or body other than the data subject, controller, processor, and people who, under the direct authority of the controller or processor, are authorized to process personal data. (E.g., *Cycle and Transport Map layers available via the openstreetmap.org website operated by **Gravitystorm Limited, New Malden, United Kingdom**.*)

6. **Authority**: An authority with the power to create or enforce laws or determine their compliance. (E.g., *We may disclose your data in response to official requests (e.g., court orders, subpoenas, search warrants, national security requests, etc.) ("requests") that we receive from **government authorities** or **parties to legal proceedings**.*)

7. **Data Subject**: The term *data subject* is specific to the GDPR but is functionally equivalent to the term *individual* and the ISO/IEC term *PII Principle*. (E.g., *This document is mainly intended for **OpenStreetMap contributors**.*)

8. **Data Source**: *Source* is the direct point of data collection; *origin* would indicate the original/other points where the data originates from. (E.g., *User to user messages are visible to the **sender** and recipient.*)

9. **Required Purpose**: The purpose of processing personal data required for service provision. (E.g., *We also use cookies and similar technologies **to recognize and improve your use of our websites**.*)

10. **Not-Required Purpose**: The purpose of processing personal data is not required for service provision.

11. **Processing**: The processing performed on personal data. (E.g., *When you visit this website or other websites, your **browser transmits data to our server**.*)

12. **Non-Personal Data**: The term Non-Personal Data is provided to distinguish between Personal Data and other data, indicating which data is regulated by privacy laws. (E.g., *We collect **information about your browser or application and your interaction with our website**, including (a) IP address, (b) **browser and device type**, (c) **operating system**, (d) **referring web page**, (e) **the date and time of page visits**, and (f) **the pages accessed on our websites**.*)

13. **Personal Data**: This definition of personal data encompasses the concepts used in GDPR Art.4-1 for *personal data* and ISO/IEC 27001 for *personally identifiable information (PII)*. (E.g., *The **full personal name** and **residential address** of members of the organisation.*)

14. **Organisational Measure**: Organisational measures may consist of internal policies, organizational methods or standards, and controls and audits that controllers and processors can apply to ensure the security of personal data. (E.g., *In this case, a so-called **opt-out** cookie is stored in your browser.*)

15. **Technical Measure**: Technical measures can be defined as the measures and controls afforded to systems and any technological aspect of an organization, such as devices, networks, and hardware. (E.g., *In order to protect the security of your data during transmission, we use appropriate **encryption methods in line***

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13871889

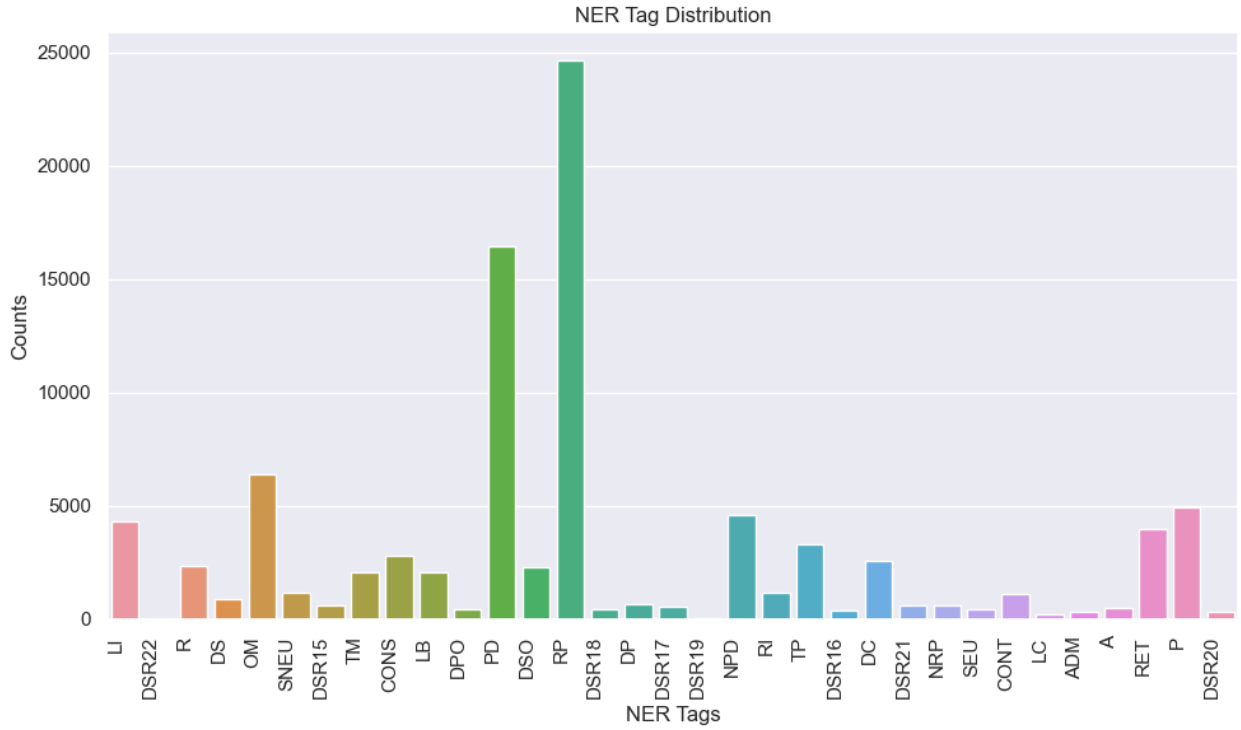



Figure 1: The number of occurrences of each NER tag in the annotated data set.

***with the latest technology (e.g., SSL/TLS)*** *and secure technical systems.*)

16. **Legal Basis**: Legal basis (*plural: legal bases*) are defined by legislations and regulations, whose applicability is usually restricted to specific jurisdictions. (E.g., *The processing of this data is necessary for compliance with a* ***legal obligation (see GDPR article 6.1c)****.*)

17. **Consent**: Consent of the Data Subject for specified processing. (E.g., ***You can stop this behaviour*** *by explicitly turning Gravatar support off in your account settings.*)

18. **Contract**: Creation, completion, fulfillment, or performance of a contract involving specified processing. (E.g., *To our operations and working group personnel that have signed* ***confidentiality agreements****.*)

19. **Legitimate Interest**: Legitimate interests of a Party as justification for specified processing. (E.g., *We value your privacy and strive to achieve a balance between the* ***legitimate interests of the OpenStreetMap project*** *and your interests and rights.*)

20. **Automated Decision Making**: Processing that involves automated decision making. (E.g., *If you have consented to data processing or if a contract for data processing exists and* ***data processing is carried out using automated processes****.*)

21. **Retention**: Duration, temporal limitation, or condition on storage of personal data. (E.g., *Payment details for both classes of members is retained for accounting purposes* ***as long as required by law****.*)

22. **Scale EU**: Geographic coverage of processing within the European Union. (E.g., *This Section 14.2 applies only to natural persons* ***residing in the European Economic Area*** *and the United Kingdom.*)

23. **Scale Non-EU**: Geographic coverage of processing outside the European Union. (E.g., *Map tiles are provided by* ***a global network*** *of cache servers.*)

24. **Right**: The right(s) applicable, provided, or expected. (E.g., *We value your privacy and strive to achieve a balance between the legitimate interests of the OpenStreetMap project and* ***your interests and rights****.*)

25. **Lodge Complaint**: A data subject can complain to a supervisory authority if the data subject considers that the processing of personal data infringes GDPR. (E.g., *You also have the right to complain to the Bavarian state commissioner for data protection.*)

26. **Data Subject Rights** (26-33):

    - Art. 15 Right of access by the data subject
    - Art. 16 Right to rectification
    - Art. 17 Right to erasure ('right to be forgotten')
    - Art. 18 Right to restriction of processing

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13871889



- Art. 19 Notification obligation regarding rectification or erasure of personal data or restriction of processing
- Art. 20 Right to data portability
- Art. 21 Right to object
- Art. 22 Automated individual decision-making, including profiling

These *Data Subject Rights* are outlined in Chapter 3 of the GDPR[2]. (E.g., *If incorrect personal data are processed, you have the* ***right to correct them (Art. 16 GDPR)***.)

Table 1 shows the entity frequency table for individual tokens. This table simply states the overall frequency of tokens available in the dataset.

| Label | Frequency | Percentage |
|---|---|---|
| PD | 4200 | 23.24% |
| P | 2909 | 16.09% |
| RP | 1745 | 9.65% |
| DC | 1559 | 8.62% |
| NPD | 955 | 5.28% |
| TP | 942 | 5.21% |
| CONS | 686 | 3.79% |
| TM | 648 | 3.58% |
| R | 585 | 3.24% |
| DS | 510 | 2.82% |
| LB | 419 | 2.32% |
| DSO | 408 | 2.26% |
| OM | 386 | 2.14% |
| LI | 306 | 1.69% |
| RET | 291 | 1.61% |
| SNEU | 246 | 1.36% |
| RI | 221 | 1.22% |
| DP | 143 | 0.79% |
| CONT | 129 | 0.71% |
| A | 124 | 0.69% |
| ADM | 109 | 0.60% |
| SEU | 100 | 0.55% |
| DSR17 | 84 | 0.46% |
| DSR15 | 67 | 0.37% |
| DPO | 58 | 0.32% |
| DSR16 | 57 | 0.32% |
| DSR21 | 50 | 0.28% |
| NRP | 38 | 0.21% |
| DSR18 | 37 | 0.20% |
| LC | 29 | 0.16% |
| DSR20 | 29 | 0.16% |
| DSR19 | 4 | 0.02% |
| DSR22 | 2 | 0.01% |
| **Overall** | **18076** | **100.00%** |

Table 1: Entity frequency table with percentages (*rounded to two decimal places*) and overall total.

[2] https://gdpr.eu/tag/chapter-3/

The privacy policies have been reviewed by two legal experts and annotated. While annotating privacy policies, the annotators ensured proper formatting, such as line and word breaks. For inter-annotator agreement, the F1-measure between the two annotators, based on a set of 20 documents, is **0.6563** while Cohen's Kappa score is **0.6412**. Although the F1-score of **0.6563** indicates moderate agreement between annotators, it does not account for chance agreement. Cohen's Kappa, however, factors this in by underscoring the potential existence of systematic bias or inconsistencies in annotation.

The lower score is primarily the result of discrepancies in the use of Word's comment feature rather than disagreements in labeling. The decision to utilize Word's comment feature for annotating sentences or words was influenced by the annotators' familiarity with this method. When annotators highlight text for annotation, slight inconsistencies in selecting text (*including an extra space before or after a word*) can lead to discrepancies in the annotated data. These minor differences, while seemingly trivial, can affect automated processing. This affects the inter-annotator agreement scores, as it may appear that annotators disagree on the annotation of the same text when, in fact, they are aligned in their understanding but differ in their selection.

After the final annotation task, we performed a basic error analysis using Precision, Recall, and F1 scores. The results showed a precision of **0.70**, a recall of **0.62**, and an F1 score of **0.65**. To encourage further academic and practical explorations in privacy policy analysis and NER applications, our dataset is publicly accessible at the following link[3]. The dataset follows the CoNLL-2002 [21] format.

## CONCLUSION

In this study, we present a dataset enriched with Named Entity Recognition (NER) annotations that comply with GDPR. It is designed to enhance the readability and accessibility of privacy policies from 44 online platforms. The Cohen's Kappa of **0.64** reflects the reliability and consistency of the annotation process but may be influenced by sentence segmentation variations. This dataset is a fundamental resource for the ongoing discussion on online privacy. Online data privacy presents dynamic challenges that require scrutiny, enhancements, and expansions.

This dataset lays the groundwork for future research in making privacy policies more accessible. By identifying key entities, subsequent research can focus on summarizing these policies, generating user-friendly interpretations, or creating visualization tools that simplify understanding privacy policies. Integrating Relationship Extraction (RE) could expand the dataset by capturing intricate relationships between entities and providing a more holistic understanding of privacy policies. We envision this corpus as a stepping stone towards these goals.

[3] https://huggingface.co/datasets/PaDaS-Lab/gdpr-compliant-ner

https://doi.org/10.5281/zenodo.13871889

## ACKNOWLEDGMENTS

SPONSORED BY THE

Federal Ministry
of Education
and Research

The project on which this report is based was funded by the German Federal Ministry of Education and Research (BMBF) under the funding code 01|S20049, by the project DEEP WRITE (Grant No. 16DHBKI059) and the OpenWebSearch.eu project, funded by the EU under the GA 101070014.

## REFERENCES

[1] M. Smith, C. Szongott, B. Henne, and G. von Voigt, "Big data privacy issues in public social media," in *2012 6th IEEE International Conference on Digital Ecosystems and Technologies (DEST)*, 2012, pp. 1–6. 10.1109/DEST.2012.6227909

[2] D. Ibdah, N. Lachtar, S. M. Raparthi, and A. Bacha, ""why should i read the privacy policy, i just need the service": A study on attitudes and perceptions toward privacy policies," *IEEE Access*, vol. 9, pp. 166 465–166 487, 2021. 10.1109/ACCESS.2021.3130086

[3] G. B. Herwanto, G. Quirchmayr, and A. M. Tjoa, "A named entity recognition based approach for privacy requirements engineering," in *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, 2021, pp. 406–411. 10.1109/REW53955.2021.00072

[4] J. M. Del Alamo, D. S. Guaman, B. García, and A. Diez, "A systematic mapping study on automated analysis of privacy policies," *Computing*, vol. 104, no. 9, pp. 2053–2076, 2022.

[5] A. Shankar, A. Waldis, C. Bless, M. Andueza Rodriguez, and L. Mazzola, "Privacyglue: A benchmark dataset for general language understanding in privacy policies," *Applied Sciences*, vol. 13, no. 6, p. 3701, 2023.

[6] S. Wilson *et al.*, "The creation and analysis of a website privacy policy corpus," 2016, pp. 1330–1340. 10.18653/v1/P16-1126

[7] E. Poplavska, T. Norton, S. Wilson, and N. Sadeh, "From prescription to description: Mapping the gdpr to a privacy policy corpus annotation scheme," in 2020. 10.3233/FAIA200874

[8] S. Zimmeck *et al.*, "Maps: Scaling privacy compliance analysis to a million apps," *Proceedings on Privacy Enhancing Technologies*, vol. 2019, pp. 66–86, 2019. 10.2478/popets-2019-0037

[9] R. Amos, G. Acar, E. Lucherini, M. Kshirsagar, A. Narayanan, and J. R. Mayer, "Privacy policies over time: Curation and analysis of a million-document dataset," *CoRR*, vol. abs/2008.09159, 2020. https://arxiv.org/abs/2008.09159

[10] S. Urchs., J. Mitrović., and M. Granitzer., "Design and implementation of german legal decision corpora," in *Proceedings of the 13th International Conference on Agents and Artificial Intelligence - Volume 2: ICAART*, INSTICC, 2021, pp. 515–521. 10.5220/0010187305150521

[11] H. Darji, J. Mitrović, and M. Granitzer, "A dataset of german legal reference annotations," in *Proceedings of the Nineteenth International Conference on Artificial Intelligence and Law*, 2023, pp. 392–396. 10.1145/3594536.3595173

[12] M. Al-Maamari, M. Istaiti, S. Zerhoudi, M. Dinzinger, M. Granitzer, and J. Mitrovic, "A comprehensive dataset for webpage classification,"

[13] M. Dinzinger, S. Zerhoudi, M. Al-Maamari, M. Istaiti, J. Mitrović, and M. Granitzer, "Owler: Preliminary results for building a collaborative open web crawler,"

[14] E. Commission, *Regulation (eu) 2016/679 of the european parliament and of the council of 27 april 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing directive 95/46/ec (general data protection regulation)*, accessed: 2022-03-22, European Commission, 25, 2018. https://eur-lex.europa.eu/eli/reg/2016/679/oj

[15] H. J. Pandit *et al.*, "Creating a vocabulary for data privacy," in *On the Move to Meaningful Internet Systems: OTM 2019 Conferences*, 2019, pp. 714–730.

[16] S. H2020, *Scalable policy-aware linked data architecture for privacy, transparency and compliance*, accessed: 2022-03-22, SPECIAL H2020. https://www.specialprivacy.eu/

[17] A. Gerl, N. Bennani, H. Kosch, and L. Brunie, "Lpl, towards a gdpr-compliant privacy language: Formal definition and usage," in *Transactions on Large-Scale Data-and Knowledge-Centered Systems XXXVII*, 2018, pp. 41–80.

[18] H. Jiang and A. Bouabdallah, "Jacpol: A simple but expressive json-based access control policy language," in *Information Security Theory and Practice*, 2018, pp. 56–72.

[19] M.-R. Ulbricht and F. Pallas, "Yappl - a lightweight privacy preference language for legally sufficient and automated consent provision in iot scenarios," in *DPM/CBT@ESORICS*, 2018, pp. 329–344. 10.1007/978-3-030-00305-0_23

[20] S. Becher and A. Gerl, "Contra preference language: Privacy preference unification via privacy interfaces," *Sensors*, vol. 22, no. 14, 2022. 10.3390/s22145428

[21] E. F. Tjong Kim Sang, "Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition," in *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002. https://aclanthology.org/W02-2024

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13863647



# UTILISING TRANSFORMER MODELS FOR CONTROLLABLE SCIENTIFIC ABSTRACTIVE SUMMARIZATION

S. Weidinger, S. Frank[1], ISDS, Graz University of Technology, Graz, Austria
A. Wagner, CERN, Geneva, Switzerland
C. Gütl, ISDS, Graz University of Technology, Graz, Austria
[1]also at CERN, Geneva, Switzerland

*Abstract*

With the rapid pace of new publications, researchers face significant challenges in finding and extracting relevant information. However, the summarization of scientific texts remains challenging due to the complexity of domain-specific knowledge and length of the text, as well as the need for traceability of information back to the source. In this paper, we utilize SLED, an efficient long-document transformer approach, with the aim of facilitating more efficient information retrieval. SLED has demonstrated promising results according to the short document SCROLLS benchmark, as well as exceeding both extractive and abstractive baselines, offering a trade-off between performance and computational costs. Integration of semantic search methods can effectively identify original sentences, further enhancing the reliability and trustability of a system.

## INTRODUCTION

Scientific research is essential for people who want to stay informed about scientific developments, but exploring and evaluating the wide selection of articles requires considerable time and effort. With the rapid pace of new publications, researchers face significant challenges in finding and extracting relevant information. Even considering filtering functions such as sorting by age or citations, traditional keyword-based search engines require users to explore countless results. Considering this, automatic summarization seems like a natural approach to tackle this challenge.

Automatic summarization aims to condense essential information from one or more documents into a concise and coherent summary. Although automatic summarization methods have received significant attention in the context of general text summarization, scientific articles present additional challenges that need to be considered. Document length, domain specificity, presence of complex or technical terminology, interconnected concepts, and often consistent structure present different challenges and opportunities and require additional consideration [1]. The need for attribution of information to its source further complicates the issue, as this traceability is essential for research. Aside from this, the rapid increase in computational costs of training or fine-tuning transformer-based language models makes considering economic solutions essential.

Research in long-text summarization is complicated by the shortage of datasets. Although some options exist, the focus on scientific articles, in particular, and the requirement of full-text data limit the selection. With this in mind, we created two datasets for controllable scientific abstractive summarization and used them to train and test a new summarization model built on computer-science-centered articles. Our approach seeks to enhance the accessibility and usability of scientific literature, facilitating faster and more efficient information retrieval for researchers, scholars, and practitioners in various scientific disciplines.

In this paper, we present these datasets and explain the considerations that were made for their creation. Furthermore, we propose a method for low-resource, traceable scientific summarization along with experimental results showing its effectiveness and utility. We also discuss the implications of the findings and outline future research directions.

## RELATED WORK

Although not as prevalent as general automatic summarization, long document summarization and the summarization of scientific articles have seen an increase in active research in recent years. In this section, we give an overview of the existing literature that covers one or both of the specific research fields.

Even outside of summarizing scientific search results, specifically, the summarization of long texts comes with challenges. Early approaches of automatic summarization methods usually focus on news articles that are significantly shorter than most scientific articles. Most commonly used models have a maximum token length of 1024. However, scientific articles have been found to have an average token length of around 10.7k [7]. This means that many of these methods are unusable for the summarization of full-text scientific articles due to the difference in length. This has created the task of the more specific *Long Document Summarization*, which typically requires more effort as knowledge requirements increase.

Previously studied methods for long-document summarization have used a variety of approaches to extract essential information while retaining coherence and key insights. LexRank, which was one of the first methods to find widespread use, calculated sentence similarity using graph-based methods to select the most important sentences [4]. More recently, advances in machine learning have enabled the development of transformer-based summarization models such as BERT [5], BART [6], and LongT5 [7]. All of these have had a significant impact on the field and gave rise to a plethora of fine-tuned models of their own. Although these methods have shown promising results in summarizing general and shorter text documents, such as news articles,

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863647

handling long documents remains a challenge due to the low token limit of most models and the rapid increase in computing costs.

Several studies have specifically addressed the task of summarizing scientific articles [9–12]. Even with the advances brought by transformer-based language models, the summarization of scientific text remains challenging due to the complexity of terminology, domain-specific knowledge, and the general length of the text, as well as the need for traceability for information. Specialized approaches are necessary. For example, Beltagy et al. (2019) introduced SciBERT, a pre-trained language model fine-tuned on scientific text, which demonstrated improved performance in various scientific NLP tasks, including summarization [8]. FacetSum, which was first presented in 2021 by Meng et al., placed a special focus on faceted summarization of scientific documents and was a fine-tuned version of BART, showed the best results for input consisting of introduction and conclusion, with full text input reaching lower scores [13]. In their recent work, Creo et al. evaluated the impact of prompting techniques on scientific summarization results and found that the use of decoder prompting led to improved performance, particularly for smaller summarization models [14].

However, even with these advances, handling long documents, capturing nuanced scientific concepts, enabling traceability of information, and creating coherent and informative summaries remain a challenge. Moreover, additional evaluation metrics and benchmarks are needed to assess the quality and effectiveness of scientific article summarization systems, as well as metrics to evaluate the factuality and allow traceability of information to provide the reliability of the summary that is essential for scientific contexts.

## DEVELOPMENT

During the research stage, the focus was on using performant language models and explainable results. For this, it was necessary to look at the problem from two angles. For one thing, it was necessary to select a dataset according to a set of requirements. For another, a model or method had to be selected to be used for further processing/fine-tuning. In both cases, we made several considerations, which will be elaborated in the following.

### *Dataset Creation*

After evaluating the existing datasets, it was found that they did not meet the requirements set in the context of this research. Either they do not meet the criteria of long-document collections, lack quality and diversity in their target summaries, or are sourced from a non-technical domain. Due to this, it was decided to create specialized datasets that included not only abstracts and/or conclusions, but the entire article text, as well as a focus on articles from the field of computer science.

Of the journals using *OpenReview*, a selection was made considering the focus on computer science domains, with four being determined for each dataset. The selected journals are listed in Table 1, in part due to their higher number of past conferences, which implied more papers.

Although our new datasets *OpenReview Contribution* (1.7k) and *OpenReview Summary* (11k) are smaller than commonly used datasets such as ArXiv (215k) or PubMed (133k), they are comparable to more specialized datasets such as SciTLDR (3.2k) and FacetSum (5.8k) [13]. Uniquely, the OpenReview datasets provide multiple target summaries for each input document, not only for those contained in the test and validation splits such as SciTLDR. In particular, the dataset is divided into seven different summary lengths (described in Figure 2), facilitating length-controllable summary training. The data set was divided into training, validation, and testing sets using an 80-10-10 ratio. Furthermore, it should be noted that all summaries were crafted by academic experts, since access to the OpenReview website is restricted to academic members.

### *Model Selection*

Although many approaches utilize very large language models, this usually comes with a high computing cost. In an effort to improve model accessibility, we placed emphasis on computing efficiency as well as what performance one can expect from these performant models when compared to the more common, larger models. In this study, we employ the efficient long-document transformer approach known as SLED [16]. SLED models utilize pre-trained, short-range encoder-decoder architectures, processing long-text input by dividing it into multiple overlapping chunks. Each chunk is encoded independently and then fused in the decoder phase (fusion-in-decoder). SLED has demonstrated promising results according to the long-document SCROLLS [18] benchmark, offering a trade-off between performance and computational cost. SLED using pre-trained BART models in the standard range as backbones compete with long-range transformers such as LongT5 [7] or UL2 [26], which have a larger parameter count. Moreover, the computational complexity is due to its fusion-in-decoder approach that is much smaller than that of standard transformer models with the same number of parameters, allowing higher input sizes with comparable computational costs. Additionally, it demonstrates better results than other efficient long-document transformers, such as LED [17], which are based on sparse attention.

The extractive summary is generated using a semantic search approach. This involves comparing the sentences from the abstractive summary with those of the original document. Initially, keywords are extracted from the abstractive summary using KeyBERT [23], leveraging SciDeBERTa [24] for nuanced scientific word contextualization. Candidate sentences that contain these keywords are then identified. Then, sentence embeddings are generated with SentBERT [25]. Finally, sentences are compared using cosine similarity and the most similar sentences are selected for inclusion in the extractive summary. This approach is called Sim. Search throughout this work and is depicted in Figure 1 in the post-processing step.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863647



| OpenReview Contribution | OpenReview Summary |
|---|---|
| Neural Information Processing Systems | |
| Uncertainty in Artificial Intelligence | International Conference on Learning Representations |
| Automated Machine Learning | Medical Imaging with Deep Learning |
| Transactions on Machine Learning Research | Conference on Robot Learning |

Table 1: Journals that were selected for inclusion in each dataset

## STUDY

Based on the limitations determined during the review of related work, the focus was placed on efficient long-document summarization of scientific research articles that emphasized the traceability of summary sentences back to their source. The following research questions were defined:

- How do traditional small LLMs and efficient long-document models compare to elaborate models such as GPT?
- How can the traceability of information be emphasized for automatically created summaries?

### *Experimental Setup*

Long-document summarization requires specialized language models, both because not all methods are able to cope with long-range dependencies and also because transformer-based models become infeasible due to resource limitations quickly. With this in mind, specific considerations had to be taken when the process was divided into components, as shown in Figure 1.

First, training the model required a specialized dataset that was created from scientific articles obtained from OpenReview[1]. The text was extracted from the PDFs using the GROBID-based Python library *SciPDF parser*[2].

To test the SLED model and the semantic similarity search approach, namely Sim. Search, several abstractive and extractive standard methods are selected. The BART model, with an input size of around 1,000 tokens, serves as an abstractive and the standard extractive method. TextRank is used as an extractive baseline. The simple heuristic of using the abstracts of the articles provides an additional comparison measure. For text quality analysis, the methods are further compared with a large state-of-the-art GPT (*gpt-3.5-turbo-16k*[3]) model, capable of processing 16,000 tokens. In contrast, with the hardware setup in this work, the SLED model can handle up to 12,000 tokens. For evaluation, the GPT model was prompted to generate similar summaries in style and length using the provided API[4] by OpenAI.

A central question in this study revolves around the comparison of text quality of smaller and more advanced large language models. The relatively small BART-base model has around 139 million parameters. Additionally, BART-base

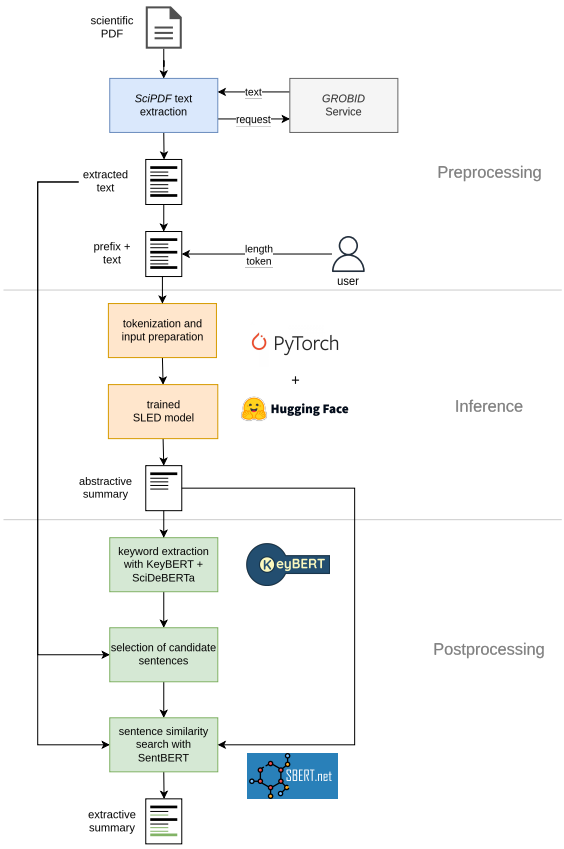



Figure 1: System architecture. From "Analyzing Long-Document Transformer Models For Scientific Abstractive Summarization" by S. Weidinger, 2024, *Master's Thesis*, p. 59

serves as the foundational model for SLED. Consequently, SLED encompasses the same number of parameters. In contrast, an initial version of the work by Singh et al. suggests that GPT-3.5-turbo has an impressive parameter count of approximately 20 billion [22].

To measure the performance of the methods, two metrics are applied: ROUGE [19] and BERTScore [20]. ROUGE is a set of lexical-based metrics that evaluate the overlap of n-grams, including ROUGE-1 and ROUGE-2, and the longest common subsequence (ROUGE-L). BERTScore calculates the similarity between reference and candidate summaries, using the BERT model *microsoft/deberta-large-mnli*[5] for contextualization, as it shows high human correlation according to the Github repository[6].

[1] https://openreview.net/
[2] https://github.com/titipata/scipdf parser/tree/master
[3] https://platform.openai.com/docs/models/gpt-3-5
[4] https://openai.com/blog/openai-api
[5] https://huggingface.co/microsoft/deberta-large-mnli
[6] https://github.com/Tiiiger/bert_score

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863647

To evaluate the quality of text, we employ UniEval [21], a multidimensional deep learning-based evaluator. UniEval measures four key dimensions: coherence, factual consistency, fluency, and relevance. These dimensions collectively contribute to an overall quality score derived from their average values.

### *Findings and Discussion*

Evaluation of the efficient transformer model SLED was carried out using ROUGE and BERTScore metrics on the OpenReview Contribution test set, comparing it to both extractive and abstractive baselines. Throughout the experiments, expert-crafted summaries served as the benchmark. The evaluation results, shown in Table 2, demonstrate the superior performance of SLED over all baselines in both metrics, particularly surpassing extractive methods by a significant margin. BART achieved results comparable to those of SLED on the ROUGE metric. However, when considering the similarity-based BERTScore metric, SLED exhibited a more substantial lead over BART, implying that SLED generates summaries that closely resemble human-crafted benchmark summaries and is better at copying their style and wording. Furthermore, the similarity search approach Sim. Search showed strong performance, outperforming TextRank and the heuristic method, although it fell short compared to SLED and BART. The results indicate that the similarity search approach effectively extracts meaningful information from the input text, generating good extractive summaries.

In particular, the abstracts of the articles proved to be a more suitable summary source compared to the outputs generated by TextRank, which received the lowest scores among all the evaluated methods. Thus, relying on the abstracts is generally more reliable than using TextRank summaries. Furthermore, the experiments confirmed that length-controllable summarization approaches improve the outcomes, yielding more valuable summaries. When using a length token as guide signal, significant improvements were observed in both ROUGE and BERTScore metrics.

Additionally, a detailed analysis was conducted to assess the performance of each method in various summary lengths. As shown in Figure 2, the SLED model consistently outperforms other methods regardless of the summary length, with the heuristic method yielding comparable results, particularly in very long summaries. Consequently, for scientific articles, automatic summarization techniques prove more advantageous for shorter recaps, given that the abstracts of articles are typically lengthy and already encompass essential information. Additionally, it should be noted that the similarity search approach demonstrates effectiveness for shorter summaries, although its relevance diminishes with longer recaps. Nevertheless, Sim. Search consistently produces results that are either better or comparable to those of the heuristic method and abstractive models, proving that semantic search is an effective approach to finding sentence origins.

For evaluation purposes of summary text quality, a subset of the OpenReview Contribution test set was selected that consists of 91 randomly chosen summaries along with their corresponding articles. These summaries are classified under the label "long" containing between 90 and 125 words each. The evaluation assesses the quality of these summary texts in terms of coherence, factual consistency, fluency, and relevance. The summaries composed by experts serve as a benchmark. The results are shown in Table 3. Although the abstracts of the articles are of superior quality on average, they often lack fluency and relevance. Notably, among automatic summarization methods, GPT followed by SLED demonstrate the most impressive results, particularly excelling in fluency and relevance. Both GPT and SLED consistently produce more pertinent summaries with high-quality sentences. From Table 3, it is evident that the summary texts generated by the extractive methods exhibit deficiencies in coherence and relevance. However, the similarity search method shows higher factual consistency compared to SLED and BART. This suggests that simply extracting phrases from input texts can enhance the system's factuality and increase trustworthiness. Furthermore, it should be noted that language models with fewer parameters can rival larger, more complex models such as GPT in terms of text quality. As illustrated in Table 3, SLED scores only 3.36 points less in the overall score, demonstrating comparable performance despite its smaller size. Surprisingly, the standard range model BART demonstrates nearly identical overall performance compared to the long-document model SLED. However, a deeper investigation revealed that BART tended to hallucinate in around ten percent of the summaries by creating human-like comments, primarly to fulfill the length requirement. These comments and annotations were incorporated by experts in some samples to provide additional information to the authors of the articles and were learned during the training by the models. On the contrary, SLED generated hallucinatory annotations in only about one percent of cases. Therefore, SLED, in contrast to BART, included more relevant content of the input and consequently produced more informative summaries overall. As a result, long-document models are needed to capture the essential information, often incorporated in the results and conclusion, hence, in the end of the articles.

## CONCLUSION AND FUTURE WORK

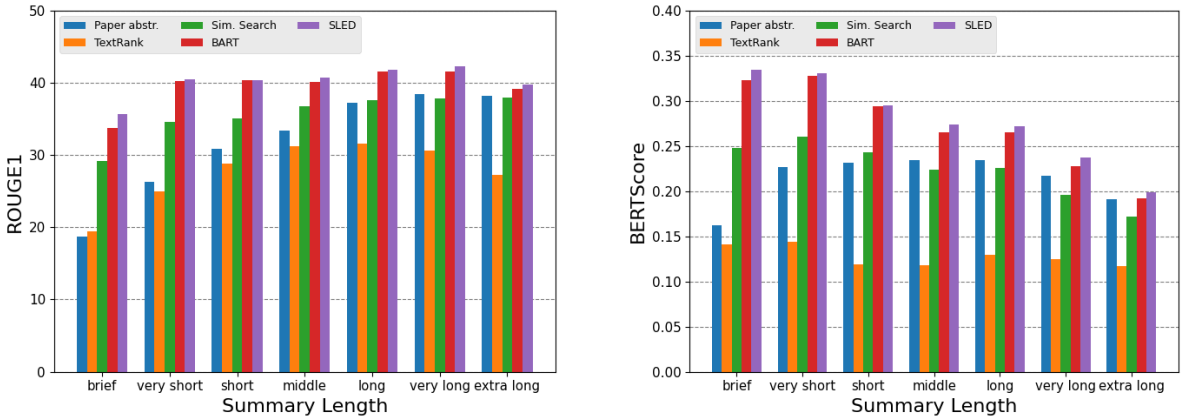
This study found that large language models (LLMs) with relative low parameter count and computational costs can produce competitive results when compared to large sophisticate models such as GPT. In particular, the efficient long-document transformer SLED, employing the fusion-in-decoder technique, exhibited impressive performance that exceeded both extractive and abstractive baselines. SLED showcases its proficiency in capturing long-distance relationships and generating highly relevant summaries, outperforming standard range models such as BART. Furthermore, the integration of semantic search methods can effectively

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863647



| Method | Input source | Length signal | ROUGE1 | ROUGE2 | ROUGELsum | BERTScore |
|---|---|---|---|---|---|---|
| heuristic | paper abstr. | no | 32.73 | 9.39 | 20.23 | 0.220 |
| TextRank | full paper | no | 29.09 | 6.21 | 19.26 | 0.114 |
| TextRank | full paper | yes | 30.95 | 6.52 | 20.41 | 0.128 |
| Sim. Search | summ.+paper | yes | 35.77 | 9.60 | 23.44 | 0.229 |
| $BART_{base}$ | 1K tokens | yes | 36.81 | 10.45 | 33.06 | 0.276 |
| $SLED_{base}$ | 12K tokens | no | 32.68 | 9.90 | 29.40 | 0.268 |
| $SLED_{base}$ | 12K tokens | yes | **36.95** | **10.81** | **33.12** | **0.282** |

Table 2: Performance evaluation was conducted on the OpenReview Contribution test set. SLED surpasses both extractive baselines and BART in terms of ROUGE and BERTScore metrics, underscoring SLED's ability to effectively capture long-term dependencies. The similarity search approach Sim. Search shows strong performance, outperforming both TextRank and heuristic methods. The inclusion of a length signal significantly improved overall performance.




(a) Comparison on ROUGE1.

(b) Comparison on BERTScore.

Figure 2: Performance comparison across summary lengths, conducted using the OpenReview Contribution dataset. The dataset was automatically divided into seven bins, with slight adjustments for better label balance. The lengths were classified according to bin sizes as follows: "brief" - [0, 40), "very short" - [40, 55), "short" - [55, 70), "middle" - [70, 90), "long" - [90, 125), "very long" - [125, 200), and "extra long" - [200, ∞). SLED consistently outperformed other methods on average, regardless of summary size.

| Method | Type | #Params | Coherence | Consistency | Fluency | Relevance | Average |
|---|---|---|---|---|---|---|---|
| paper abstr. | extr. | - | **94.19** | **94.35** | 88.80 | 85.42 | **90.69** |
| TextRank | | - | 40.36 | 68.28 | 76.71 | 35.82 | 55.29 |
| Sim. Search | | - | 61.55 | 82.91 | 87.55 | 55.21 | 71.80 |
| $GPT_{zero\text{-}shot}$ | abstr. | ~20B | 92.37 | 84.47 | **91.63** | **91.52** | 90.00 |
| $BART_{base}$ | | 139M | 90.22 | 82.84 | 86.11 | 86.81 | 86.49 |
| $SLED_{base}$ | | 139M | 89.08 | 80.99 | 88.93 | 87.54 | 86.64 |

Table 3: Text quality was compared based on the dimensions coherence, consistency, fluency and relevance. Summaries composed of experts serve as a benchmark. While the abstracts of the articles generally exhibited superior quality, they were often deficient in fluency and relevance. In particular, among automatic summarization methods, GPT followed by SLED showed the most impressive results, demonstrating strong performance, particularly in fluency and relevance.

identify original sentences, thereby enhancing the reliability and trustability of the system.

However, the fusion-in-decoder approach is constrained by its contextual window, which may cause it to fail to establish accurate long-distance connections. Future research should be done to evaluate models that incorporate alternative efficient methods for long-document processing. Additionally, the method of searching for similar sentences based solely on entire sentences in the input text fails to capture sentence fusion and shortening, noticeably limiting performance. Moreover, traceability could help to improve the factual consistency in abstractive summaries by incorporating facts from the original sentences. Addressing these points could further improve the overall performance of the system.

https://doi.org/10.5281/zenodo.13863647



## REFERENCES


[1] Koh, H. Y., Ju, J., Liu, M., & Pan, S. (2022). An empirical survey on long document summarization: datasets, models, and metrics. *ACM Computing Surveys*, 55(8), 1–35. https://doi.org/10.1145/3545176

[2] Sen, P., Namata, G., Bilgic, M., Getoor, L., Galligher, B., & Eliassi-Rad, T. (2008). Collective Classification in Network Data. AI Magazine, 29(3), 93. https://doi.org/10.1609/aimag.v29i3.2157

[3] Clement, C.B., Bierbaum, M., O'Keeffe, K.P., & Alemi, A.A. (2019). On the Use of ArXiv as a Dataset. ArXiv, abs/1905.00075.

[4] Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. Journal of artificial intelligence research, 22, 457-479.

[5] Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. https://doi.org/10.18653/v1/N19-1423

[6] Lewis, M., Liu, Y., Goyal, N., Ghazvininejad, M., Mohamed, A., Levy, O., Stoyanov, V., & Zettlemoyer, L. (2020). BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. https://doi.org/10.18653/v1/2020.acl-main.703

[7] Guo, M., Ainslie, J., Uthus, D., Ontañón, S., Ni, J., Sung, Y., & Yang, Y. (2022). LongT5: efficient Text-To-Text transformer for long sequences. *Findings of the Association for Computational Linguistics: NAACL 2022*. https://doi.org/10.18653/v1/2022.findings-naacl.55

[8] Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A Pretrained Language Model for Scientific Text. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. https://doi.org/10.18653/v1/d19-1371

[9] Altmami, N. I., & Menai, M. E. B. (2022). Automatic summarization of scientific articles: A survey. *Journal of King Saud University-Computer and Information Sciences*, 34(4), 1011-1028. https://doi.org/10.1016/j.jksuci.2020.04.020

[10] Callegari, E., Vajdecka, P., & Xhura, D. (2023). Generating Academic Abstracts: Controlled Text Generation Using Metrics from a Target Text Allows for Transparency in the Absence of Specialized Knowledge. *Proceedings of the Workshop on GENerative, Explainable and Reasonable Artificial Learning co-located with the 15th Biannual Conference of the Italian SIGCHI Chapter (CHITALY 2023)*

[11] Sukmandhani, A. A., Arifin, Y., Zarlis, M., & Budiharto, W. (2023). Recent Trends for Text Summarization in Scientific Documents. *2023 IEEE 9th International Conference on Computing, Engineering and Design (ICCED)*, pp. 1-6. https://doi.org/10.1109/ICCED60214.2023.10425025

[12] Aswani, S., Choudhary, K., Shetty, S., & Nur, N. (2024). Automatic text summarization of scientific articles using transformers—A brief review. Journal of Autonomous Intelligence, 7(5).

[13] Meng, R., Thaker, K., Zhang, L., Dong, Y., Yuan, X., Wang, T., & He, D. (2021). Bringing Structure into Summaries: a Faceted Summarization Dataset for Long Scientific Documents. *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pp. 1080-1089. https://doi.org/10.18653/v1/2021.acl-short.137

[14] Creo, A., Lama, M., & Vidal, J. C. (2023). Prompting LLMs with content plans to enhance the summarization of scientific articles. *arXiv preprint*. https://doi.org/10.48550/arXiv.2312.08282

[15] Weidinger, S. (2024). Analyzing Long-Document Transformer Models For Scientific Abstractive Summarization. *Master's Thesis*. Graz University of Technology

[16] Ivgi M., Shaham U., & Berant J. (2023). Efficient Long-Text Understanding with Short-Text Models. *Transactions of the Association for Computational Linguistics 2023 (Volume 11)*. pp. 284–299. https://doi.org/10.1162/tacl_a_00547

[17] Beltagy, I., Peters, M. E., & Cohan, A. (2020). Longformer: The Long-Document Transformer. *Computing Research Repository (CoRR), arXiv preprint*. https://doi.org/10.48550/arXiv.2312.08282

[18] Shaham, U., Segal, E., Ivgi, M., Efrat, A., Yoran, O., Haviv, A., … Levy, O. (2022). SCROLLS: Standardized CompaRison Over Long Language Sequences. *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 12007–12021. https://doi.org/10.18653/v1/2022.emnlp-main.823

[19] Lin, C.-Y. (2004, July). ROUGE: A Package for Automatic Evaluation of Summaries. *Text Summarization Branches Out*, 74–81. Retrieved from https://aclanthology.org/W04-1013

[20] Zhang, T., Kishore, V., Wu, F., Weinberger, K. Q., & Artzi, Y. (2020). BERTScore: Evaluating Text Generation with BERT. *International Conference on Learning Representations*. Retrieved from https://openreview.net/forum?id=SkeHuCVFDr

[21] Zhong, M., Liu, Y., Yin, D., Mao, Y., Jiao, Y., Liu, P., … Han, J. (2022, December). Towards a Unified Multi-Dimensional Evaluator for Text Generation. In Y. Goldberg, Z. Kozareva, & Y. Zhang (Eds.), *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 2023–2038. https://doi.org/10.18653/v1/2022.emnlp-main.131

[22] Singh, M., Cambronero, J., Gulwani, S., Le, V., Negreanu, C., & Verbruggen, G. (2023). CodeFusion: A Pre-trained Diffusion Model for Code Generation. *arXiv preprint*. Retrieved from http://arxiv.org/abs/2310.17680

[23] Grootendorst, M. (2020). KeyBERT: Minimal keyword extraction with BERT (Version v0.3.0). Version v0.3.0. https://doi.org/10.5281/zenodo.4461265

[24] Jeong, Y., & Kim, E. (2022). SciDeBERTa: Learning DeBERTa for Science Technology Documents and Fine-Tuning Information Extraction Tasks. *IEEE Access (Volume 10)*, pp. 60805–60813. https://doi.org/10.1109/ACCESS.2022.3180830

https://doi.org/10.5281/zenodo.13863647



[25] Reimers, N., & Gurevych, I. (2019). Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pp. 3982–3992. https://doi.org/10.18653/v1/D19-1410

[26] Tay, Y., Dehghani, M., Tran, V. Q., Garcia, X., Wei, J., Wang, X., … Metzler, D. (2023). UL2: Unifying Language Learning Paradigms. *arXiv [Cs.CL]*. Retrieved from http://arxiv.org/abs/2205.05131

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13871185

# CREATING EXPLAINABLE SUMMARIES FOR LONG SCIENTIFIC DOCUMENTS USING LARGE LANGUAGE MODELS

S. Frank[1], S. Schäffer, ISDS, Graz University of Technology, Graz, Austria
A. Wagner, CERN, Geneva, Switzerland
T. Gütl, A. Nussbaumer, ISDS, Graz University of Technology, Graz, Austria
A. Steinmaurer, IT:U – Institute of Digital Sciences Austria, Austria
[1]also at CERN, Geneva, Switzerland

*Abstract*

This paper proposes a system intended for the summarization of long scientific documents, with particular emphasis being placed on accuracy, coherence, and transparency to create trustable summaries through the combination of a large language model with an explanation mechanism. The primary goal of this system is to help users efficiently acquire the most significant information from scientific papers for applications and use cases related to science search. The resulting system was evaluated with two user studies to measure the quality of the resulting summaries and the efficiency of the system's explanation functionality, as well as its impact on trustability of the systems. Although the results were generally promising, there was a high deviation in the ratings for some metrics, indicating that further research is needed to provide reliable and consistent performance.

## INTRODUCTION

With the increase in content available on the Web showing no sign of slowing, it has become an unavoidable source of information for its users. Although this vast amount of information opens up new opportunities, it can also present challenges. Search engines are becoming increasingly indispensable on a Web that encompasses more content than anyone could ever sift through, with people recognizing the potential for trouble in this situation. Facilitating access to information involves securing its availability and delivery. With most search engines based on very few indices until now, the question of retrieval and potential for censorship and bias is being addressed with the creation of the Open Web Index as part of the Open Web Search project.

Although this addresses information retrieval, the presentation of the results in a way that does not cause information overload remains difficult. Keeping up with and reviewing content has become challenging, particularly in the scientific field, where the number of online publications keeps increasing and is quickly outpacing what simple search functions can make manageable. This task is further complicated by the fact that scientific articles are usually several pages long and require intensive analysis by the reader. The summarization of scientific articles is one way to make this process more efficient. However, the creation of manual summaries is just as time-consuming and can, furthermore, be subjective. This increases the need for an automated system that can generate precise, cohesive, and informative summaries.

This paper proposes a system intended for the summarization of long scientific documents, with particular emphasis being placed on accuracy, coherence, and transparency to create trustable summaries through the combination of a large language model with an explanation mechanism. The primary goal of this system is to help users efficiently acquire the most significant information from scientific papers for applications and use cases related to science search.

The resulting system was evaluated with two user studies which evaluated the quality of the summaries and the efficiency of the system's explanation functionality and intended to answer the following research questions:

**RQ 1:** How do authors evaluate the quality of the summaries generated by the AI system according to selected metrics?

**RQ 2:** Can this approach improve the transparency of the summary and does this influence user trust?

This paper will first give an overview of the background and related work, followed by a description of the system pipeline. Subsequently, two user studies will be discussed, as well as limitations of the proposed solution. In the last section, we consider possible future work and conclude the paper.

## BACKGROUND AND RELATED WORK

The amount of data accessible on the Internet has expanded rapidly in recent years, leading to a need for methods to condense it into valuable information for users. This need for a strategy to manage this overload, as well as the introduction of the attention-based transformer model, as presented by Vaswani et al. [1], has had a significant influence on advances in the field of automatic summarization. Although initial research focused primarily on the production of brief summaries of news articles and the creation of corresponding datasets [2–4], the potential for further domains has been recognized, leading to further rapid developments in the field [5, 6].

Although the results have been promising and subsequent systems have found strong mainstream acceptance [7], problems remain before automatic summarization and large language models can be considered reliable and may be used in professional contexts. As deep learning techniques continue to advance in power and complexity, it becomes increasingly difficult to understand the reasoning behind the

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13871185



results. However, this understanding is essential to assess the trustworthiness of the model. Hallucination, that is, the creation of text whose information is not implied by the source, can lead to misrepresentation of facts and mislead the user, making the results of these models inherently untrustworthy. Furthermore, it is generally unclear to the user what data the language model was trained on. This prevents the evaluation of potential biases in the data or the resulting model [8]. Explainable Artificial Intelligence (XAI) focuses on addressing these issues by highlighting the need to clarify the decision-making processes used by complex AI models and improving the comprehensibility and transparency of black-box models [9]. This can be done in several ways.

Feature-importance-based methods use character-level features [10], n-grams [11], or latent features [12] to calculate the relevance of the feature to the result. Alternatively, a simple, more transparent model can be trained on results and then used to explain the original model's behavior. An example that makes use of this approach is *Local Interpretable Model-Agnostic Explanations* (LIME), which does so locally for single predictions [13].

Other systems use different approaches: QUINT utilizes provenance-based explainability to explain reasoning; the decision process is described to the user. It does so by visualizing the sequence of actions between the user's natural language input and the system's response [14]. ESCA, on the other hand, uses a more direct approach that also enables the user to intervene to direct the process if necessary. It produces explainable results by communicating centrality, sentence interactions, and attribute scores such as novelty and relevance to the user, showing impressive results compared to state-of-the-art models [15]. Another system, explAIner [18], uses TensorBoard and extends it by embedding explainers and enabling the execution of explainability methods at run-time. Visualizing the decision process is done using graphs.

Another approach to facilitate explainability of a result is to link back to the information source or sources for the different parts of the summary. Norkute et al. proposed a system for the use with legal document summarization [19]. They compared the attention vector approach with the source attribution approach and evaluated the effect they had on the trustworthiness of the results, as well as the effect on editor efficiency.

Attention highlighting was found to considerably speed up the review process, while source highlighting did not have a significant effect on this task. There was a similar conclusion for trustworthiness - while attention highlighting more closely mirrored how editors would go about summarizing documents and thus made users more confident that the whole document was considered by the system and increased trust because of it, source highlighting did not have this effect.

However, although promising, the second approach only included two editors who reviewed the system and needs to be evaluated in more settings and with a larger test group to be considered reliable.

## SYSTEM DESIGN

Similarly to the systems described above, facilitating the explainability of the result was an essential aspect for the creation of this summarization system. To create explainable summaries, there were multiple steps in the process that needed to be addressed. Since the input was PDF files, processing them to create usable data for training and testing was an essential first step. Text extraction, cleaning of this extracted text, and sentence segmentation are considered to be part of this.

The next stage focused on the summarization process, which consisted of further pre-processing tasks via chunking and tokenizing, as well as the actual summarization step. Due to the need for a high maximum token count to summarize scientific articles, the choice of language model was limited. Subsequently, the explainability of the summary is addressed. Figure 1 shows a simplified representation of the system architecture.

Focusing on the approaches proposed by Norkute et al. [19] due to the available evaluation results, the source highlighting approach was chosen in the first step. This allowed for a comparison of the effect of the approach on legal summarization uses versus scientific summarization applications.

For the different steps of implementation, we made use of a variety of existing tools to simplify the process. The user interface itself uses Gradio to facilitate the upload of PDF files. Following this, the entire document text is extracted using the PDFMiner library. References and bibliographic entries are excluded, and any headings are marked to simplify later steps in the process.

As previously mentioned, large language models have a given maximum sequence length that the input cannot exceed - usually, any text surpassing this limit will be truncated. The exact number of tokens depends on the chosen model; it was with this in mind that the selection was made for this case. Scientific articles are usually many pages long, which limited the number of possibilities. The final selection was the Llama-2-7b-Chat model, which, although it supports input lengths of up to 4096 tokens, could not handle full-length scientific articles. Due to this, the text was first segmented to fit the token limit, taking care to set chunk overlap to 50 characters to facilitate the retention of context and continuity. Each of these text chunks is then fed to the language-model pipeline supplied by HuggingFace, resulting in a summary that is aggregated into a full-text summary.

The Explainable AI module, whose aim is to facilitate explainability of the created summary, functions by finding the most similar sentence from the source text for each summary sentence and presenting it to the user. Once again, the first step is tokenization, both for the original text and the generated summary. Special care was taken to recognize commonly used abbreviations such as "etc." as non-terminating entities. Following this, semantic analysis takes place using the SBERT based language models provided by SentenceTransformers as hosted on HuggingFace.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13871185

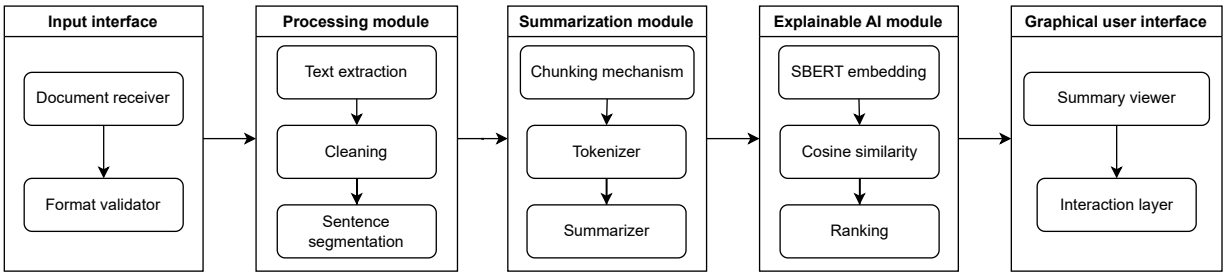



Figure 1: A simplified representation of the system architecture and its core components. From "Summarizing Long Scientific Documents: Leveraging Llama2-7B-Chat with Explainable AI" by S. Schäffer, p. 53. Adapted with permission [16].

Within these experiments, the all-MiniLM-L6-v2 model was found to have the best balance of performance and speed. Using mean pooling for standardizing the vector representations of the sentences incorporates the meaning of the entire sentence. Finally, the similarity between the sentences is calculated by using cosine similarity. The highest similarity score is returned to the user.

## STUDY DESIGN, FINDINGS AND DISCUSSION

Two studies were carried out to evaluate the described explainability system, both using user feedback used to measure the quality of the results. In accordance with RQ 1, the first study asked the authors of the summarized papers to evaluate summary quality in a number of metrics. The second study evaluates the impact on the explainability method and how it impacts trustability in users, as well as overall summary quality metrics, as posed in RQ 2. For this evaluation, a variety of users were asked to rate the applicable metrics on a Likert scale of 1 to 5.

### *Author Study*

For this study, we used the described system to create 20 summaries of articles published in the Journal of University Computer Science (J.UCS). The associated 68 authors were invited to evaluate the generated summaries using Likert scales. Of these, 11 authors answered the questionnaire.

As seen in Table 1, accuracy was generally rated well, with a majority of the participants rating this aspect 4 out of 5, with the last choosing 5. Both the highlighting of key contributions and the general coherence of the summaries displayed high variation in their ratings, indicating some level of inconsistent performance in the summarization process and inconsistent quality of the resulting summary. Nevertheless, general satisfaction with the summaries was rated medium to high.

In addition to these criteria, users were asked to rate the length of the generated summary. Of the 11 participants who answered the questionnaire, six found the summary length appropriate (3), three rated it "a bit too long" (4) and one person each considered it "a bit too short" (2) and "too long" (5), respectively.

Finally, participants were invited to provide feedback via free text. The responses given in this section indicated that secondary information, such as acknowledgments, should be excluded from the generated summaries, as well as that some summaries displayed a high degree of redundancy.

### *Explainability Study*

The second study focused on the functionality of the system's explainability module. Participants were asked to rate the explainability features to evaluate how they affected subjective user trust in the result. Unlike the first study, in this case anyone was welcome to participate. The evaluation was completed by 34 people, 33 (97.1%) of them between 18 and 34, and 1 (2.9%) between 45 and 54 years old. They came from various educational backgrounds, with 6 participants (17.6%) being high school graduates, 20 (58.8%) having a bachelor's degree, 6 (17.6%) with a master's degree, and two (5.9%) with a completed doctorate, covering a variety of demographics. Seven participants (20.6%) indicated that they considered themselves experienced (4) with artificial intelligence tools. The majority (15 participants; 44.1%) stated moderate experience (3), 11 (32.4%) considered themselves inexperienced, and one participant (2.9%) alleged that they had no experience. This range of experience levels means that the system was reviewed from a variety of points of view.

Users were provided with a brief description of the system, including images, as well as a YouTube video explaining its use. In the next step, they were asked to review the summaries with their respective source fragments according to a variety of criteria specified in Table 2.

The evaluation scores were generally high and all metrics received scores of 3 and 4 out of 5 from the majority of participants. The trust in the accuracy of the summary was scored the lowest, with a mean of 3.41, but a median of 3. This indicates that the transparency may not be sufficient with the used visualisation technique, or the quality of summaries insufficient, overall.

As in the previous study, participants received a free text question that allowed them to provide additional feedback. Multiple participants noted the increased time efficiency of automatic summarization, as well as the clarity and structure of the results and their direct comparison to portions of the source text, which they felt increased the level of trust towards the system.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13871185



| Metric | Very low | Low | Moderate | High | Very high | Mean | Stdev |
|---|---|---|---|---|---|---|---|
| Accuracy | 1 | 1 | 0 | 6 | 3 | 3.82 | 1.25 |
| Key contributions | 1 | 1 | 2 | 3 | 4 | 3.73 | 1.35 |
| Coherence | 1 | 1 | 4 | 4 | 1 | 3.27 | 1.10 |
| Overall satisfaction | 1 | 0 | 3 | 5 | 2 | 3.64 | 1.12 |

Table 1: Author evaluation of quantitative evaluation criteria on a scale from 1 (Very low) to 5 (Very high)

| Metric | Very low | Low | Moderate | High | Very high | Mean | Stdev |
|---|---|---|---|---|---|---|---|
| Clarity | 0 | 1 | 5 | 21 | 7 | 4.00 | 0.69 |
| Trust in accuracy | 1 | 3 | 15 | 11 | 4 | 3.41 | 0.91 |
| Explainability | 0 | 2 | 7 | 13 | 12 | 4.03 | 0.89 |
| Coherence | 0 | 2 | 4 | 17 | 11 | 4.09 | 0.82 |
| Effect on trust | 0 | 3 | 4 | 20 | 7 | 3.91 | 0.82 |
| Interaction support | 0 | 1 | 7 | 18 | 8 | 3.97 | 0.75 |

Table 2: User evaluation of metrics regarding summary quality such as clarity and coherence, as well as metrics evaluating the effectiveness of the explainability module using a scale from 1 (Very low) to 5 (Very high)

In contrast, the participants mentioned that the sentences did not read as a cohesive text, stating that they were lacking cohesiveness. Furthermore, the writing was considered monotonous, with repetitive sentence structures and wording. Multiple participants questioned the reliability of the system, one stating that the summarization may omit important facets of the source text and another suggesting that the lack of critical view may impact assumptions of a reader.

### *Discussion*

Both studies involved a limited number of participants (11 and 34, respectively) which only allows for tentative conclusions. However, some interesting points can be made. Looking at the metric *Coherence*, which was rated by both the authors (see Table 1) and the general users (Table 2), it can be seen that the evaluation mean varies quite significantly, with the authors appearing to have higher expectations for the summaries than the users. This makes sense as the authors of the papers have a better overview of the full content of the papers they wrote. However, what also needs to be considered is what (if any) role subjectivity may play in this situation.

In general, the authors rated the quality of the summaries moderate to high, with a significant standard deviation in all evaluated metrics. This answers RQ 1 and implies that greater consistency in summary quality is necessary for this system to improve acceptance among authors.

The explainability module was very well received by the users in the user study, gaining high results on average. The question posed by RQ 2, how this approach influences user trust, was answered with a "high" by 58.82% of participants. The trust in accuracy was not rated as highly, which indicates that although the increased transparency of the summary did have an influence, it did not have a significant enough impact on some participants. This metric also had the highest standard deviation, which shows this disagreement among the participants.

## CONCLUSION AND FUTURE WORK

This paper describes a system that intends to summarize scientific articles in a transparent and reliable way. To this end, it looks at the problem from two perspectives: that of the author and of the general user. The research questions were posed accordingly: Is the summary quality high enough to be acceptable for the authors? Do the users feel more secure in the summary due to the transparency increasing methods and how does this affect user trust?

The system that was created, as well as the two studies that were conducted in the course of this research, aimed to investigate these topics by assessing the performance of automatic summarization twofold. On the one hand, the created summaries were evaluated, while on the other hand, the explainability module was examined. Although the first results showed promising findings, the variation in responses for both studies indicates that more work is needed before such a system can be considered reliable, particularly in the two areas of summary quality and transparency-increasing measures.

Future work may focus on evaluating different visualization methods for the similarity score to facilitate explainability and trustability to an even higher degree. Furthermore, different language models can be used in the explainable AI module, as well as in the summarization module. Finally, it may be useful to provide not only the most similar sentence to the user but multiple sentences, as well as a confidence score, depending on the target audience of the system.

## ACKNOWLEDGEMENTS

This research and publication has been partially supported by the project OpenWebSearch.EU funding from the European Union's Horizon research and innovation programme under grant agreement No 101070014.

https://doi.org/10.5281/zenodo.13871185



## REFERENCES


[1] Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, 30.

[2] Narayan, S., Cohen, S. B., & Lapata, M. (2018). Don't Give Me the Details, Just the Summary! Topic-Aware Convolutional Neural Networks for Extreme Summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 1797-1807. https://doi.org/10.18653/v1/D18-1206

[3] Fabbri, A. R., Li, I., She, T., Li, S., & Radev, D. (2019). Multi-News: A Large-Scale Multi-Document Summarization Dataset and Abstractive Hierarchical Model. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1074-1084. https://doi.org/10.18653/v1/P19-1102

[4] Ahuja, O., Xu, J., Gupta, A., Horecka, K., & Durrett, G. (2022). ASPECTNEWS: Aspect-Oriented Summarization of News Documents. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 6494-6506. https://doi.org/10.18653/v1/2022.acl-long.449

[5] Beltagy, I., Lo, K., & Cohan, A. (2019). SciBERT: A Pretrained Language Model for Scientific Text. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 3615-3620. https://doi.org/10.18653/v1/D19-1371

[6] Syed, A. A., Gaol, F. L., & Matsuo, T. (2021). A Survey of the State-of-the-Art Models in Neural Abstractive Text Summarization. In *IEEE Access*, 9, 13248-13265.

[7] Hassani H., Silva E. S. (2023). The Role of ChatGPT in Data Science: How AI-Assisted Conversational Interfaces Are Revolutionizing the Field. In *Big Data and Cognitive Computing*, 7(2), 62. https://doi.org/10.3390/bdcc7020062

[8] Pedreschi, D., Giannotti, F., Guidotti, R., Monreale, A., Ruggieri, S., & Turini, F. (2019). Meaningful explanations of black box AI decision systems. In *Proceedings of the AAAI conference on artificial intelligence*, 33(01), 9780–9784. https://doi.org/10.1609/aaai.v33i01.33019780

[9] Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. In *ACM computing surveys (CSUR)*, 51(5), 1-42. https://doi.org/10.1145/3236009

[10] Godin, F., Demuynck, K., Dambre, J., De Neve, W., & Demeester, T. (2018). Explaining Character-Aware Neural Networks for Word-Level Prediction: Do They Discover Linguistic Rules? In *2018 Conference on Empirical Methods in Natural Language Processing*, 3275-3284. https://doi.org/10.18653/v1/D18-1365

[11] Mullenbach, J., Wiegreffe, S., Duke, J., Sun, J., & Eisenstein, J. (2018). Explainable Prediction of Medical Codes from Clinical Text. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 1101-1111. https://doi.org/10.18653/v1/N18-1100

[12] Xie, Q., Ma, X., Dai, Z., & Hovy, E. (2017). An Interpretable Knowledge Transfer Model for Knowledge Base Completion. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 950-962. https://doi.org/10.18653/v1/P17-1088

[13] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135-1144. https://doi.org/10.1145/2939672.2939778

[14] Abujabal, A., Roy, R. S., Yahya, M., & Weikum, G. (2017). QUINT: Interpretable Question Answering over Knowledge Bases. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 61-66. https://doi.org/10.18653/v1/D17-2011

[15] Wang, H., Gao, Y., Bai, Y., Lapata, M., & Huang, H. (2021). Exploring Explainable Selection to Control Abstractive Summarization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(15), 13933-13941. https://doi.org/10.1609/aaai.v35i15.17641

[16] Schäffer, S. (2024). *Summarizing Long Scientific Documents: Leveraging Llama2-7B-Chat with Explainable AI* [Unpublished Master's Thesis]. Graz University of Technology.

[17] Chopra, S., Auli, M., & Rush, A. M. (2016). Abstractive Sentence Summarization with Attentive Recurrent Neural Networks. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 93-98. https://doi.org/10.18653/v1/N16-1012

[18] Spinner, T., Schlegel, U., Schäfer, H., & El-Assady, M. (2019). explAIner: A visual analytics framework for interactive and explainable machine learning. In *IEEE Transactions on Visualization and Computer Graphics*, 26(1), 1064-1074. https://doi.org/10.1109/TVCG.2019.2934629

[19] Norkute, M., Herger, N., Michalak, L., Mulder, A., & Gao, S. (2021). Towards Explainable AI: Assessing the Usefulness and Impact of Added Explainability Features in Legal Document Summarization. In *Extended Abstracts of the 2021 CHI Conference on Human Factors in Computing Systems*. https://doi.org/10.1145/3411763.3443441

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13863880



# IMPACT OF TOKENIZATION TECHNIQUES ON URL CLASSIFICATION

M. Al-Maamari, M. Istaiti, S. Zerhoudi,
M. Dinzinger, M. Granitzer, J. Mitrovic
University of Passau, 94032 Passau, Germany

*Abstract*

Web crawling can be improved by the accurate classification of URLs to ensure relevant content is indexed and harmful content is filtered out. In this study, we examined the impact of various tokenization techniques on URL classification, a task integral to the development of intelligent web crawlers. Our investigation was conducted using a large-scale dataset of over one million URLs, categorized into 'Malicious', 'Benign', and 'Adult' classes, with detailed sub-labels for in-depth analysis [1]. We explored a range of tokenization methods, including Byte Pair Encoding (BPE), Enhanced BPE with a GPT-4 generated keyword dictionary, punctuation-based splitting, and character-level n-grams, to assess their effect on the classification accuracy and computational efficiency [2, 3]. The results indicated that while simple tokenization methods like Char 1-gram offered rapid prediction times, they were inadequate in correctly identifying more complex 'Malicious' URLs. More sophisticated techniques such as BPE and WordPiece achieved a better balance of precision and recall for 'Benign' and 'Adult' content, yet they, along with other methods, struggled with the 'Malicious' category. The findings highlight the nuanced challenges of URL classification and underscore the need for advanced tokenization approaches that can compete with the nature of malicious content while maintaining computational efficiency. Future work should focus on integrating diverse tokenization strategies and enhancing semantic comprehension within the tokenization process to improve classification performance, particularly for detecting malicious content within the vast and dynamic landscape of the web.

## INTRODUCTION

Web crawlers are fundamental tools used by search engines to collect data from the Internet, which demands the classification of URLs to improve efficiency and filter out irrelevant or harmful content. Efficient web crawling is contingent upon the avoidance of resource expense on unneeded or harmful URLs, such as those that are malicious, spam, or not relevant to the crawler's purpose. The incentive for developing robust URL classification systems is to support these intelligent crawling strategies.

The process of URL classification is a form of text classification, which involves categorizing text into organized groups. In this domain, tokenization plays a important role by breaking down text into smaller units, or tokens, that serve as input for machine learning algorithms. The choice of tokenization technique is a critical decision that can significantly influence the effectiveness of a classification model. [4] [5] Tokenization affects not only the granularity of the data but also the ability of the model to recognize patterns and make accurate classifications.

This paper aims to clarify the impact of various tokenization techniques on the task of URL classification. Given the diverse nature of URLs, which may include various structures and subcomponents, selecting an appropriate tokenization method is not trivial. We compare several tokenization methods, including Byte Pair Encoding (BPE), an enhanced version of BPE supplemented with a initial keyword dictionary, character-level n-grams, and a method based on splitting at punctuation marks.

Our investigation is grounded in the analysis of a comprehensive dataset of approximately one million URLs [1]. We employ a suite of evaluation metrics to assess the efficacy of each tokenization strategy, with a focus on accuracy, precision, recall, and the F1 score.

By concentrating on tokenization as a fundamental aspect of the URL classification process, our study provides granular insights into the influence of different tokenization approaches. The main objective is to identify the most effective tokenization technique, balancing high classification performance while being mindful of computational efficiency. Such insights are invaluable for the development of web crawlers that are more selective, sparing resources by avoiding the retrieval and indexing of unwanted URLs.

In the context of creating a more open web search ecosystem, this paper also contributes to the larger project aimed at developing an Open Web Index (OWI). As outlined in the recent work of [6], an OWI would promote a more open search ecosystem, offering genuine choice among alternative search engines and fostering a fair and collaborative information space. Our research supports this vision by enhancing the technology that supports web crawling, an essential component of search engine infrastructure. The classification of URLs based on reliable tokenization methods is a step towards enriching the open index with quality data, thereby enabling the development of declarative search engines and innovative web data products.

## BACKGROUND AND RELATED WORK

The classification of URLs has emerged as a task for enabling web crawlers to efficiently process the growing data on the World Wide Web. A web crawler, by definition, systematically navigates the web to index content for search engines and data retrieval applications [7]. With the sheer volume of web pages, it is essential to employ intelligent crawling strategies, such as focused crawlers, which aim to selectively retrieve pages relevant to specific topics or areas. URL classification facilitates this selective approach by iden-

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863880



tifying and filtering out URLs likely to lead to irrelevant or malicious content, thus optimizing the crawling process [8].

Tokenization, as the process of segmenting text into tokens, represents the first and a foundational step in any Natural Language Processing (NLP) pipeline. While the simplest approach to tokenization is to use whitespace-separated words, this can result in an inordinately large vocabulary, especially in the context of extensive corpora such as the web, additionally, this method does not work for URLs since there are no whitespaces in them. To address the inefficiencies associated with large vocabularies, subword tokenization algorithms have been developed. These algorithms, including Byte Pair Encoding (BPE), create subwords or tokens that can significantly limit the vocabulary size while retaining meaningful linguistic units, it also works with text that does not contain whitespace (e.g. URLs) [3, 4]. Tokenization strategies can significantly alter linguistic understanding and, thus, are crucial in the composition of input features for machine learning models, particularly in languages with rich morphology [5].

Previous studies have studied the impact of tokenization on machine learning model performance. In the context of text classification, various tokenization algorithms have been evaluated, demonstrating that the performance of these algorithms is contingent on multiple factors. These factors include the size and nature of the dataset, the specific classification task at hand, and the morphological complexity inherent to the language of the dataset [4]. Tokenization has also been shown to play a significant role in the context of named entity recognition (NER), where the choice of tokenization strategy can either enhance or impair model performance based on how it copes with the linguistic challenges posed by the target language [5]. In the domain of web page classification, character n-gram based features extracted from URLs have been successfully employed, showcasing the utility of tokenization techniques that do not rely on the actual content or the hyperlink structure of the pages [8]. This approach highlights the influence of tokenization in addressing the challenges associated with URL classification.

Collectively, these studies form the background against which we examine the effectiveness of various tokenization methods, with a particular emphasis on their application in URL classification for web crawlers. This exploration aims to contribute to the ongoing discussion on the optimal integration of tokenization techniques within machine learning frameworks for the enhancement of web crawling and indexing efficiency.

## METHODOLOGY

### *Data*

Our investigation utilized a comprehensive dataset comprising 1,069,715 URLs, each annotated with labels denoting its classification into 'Malicious', 'Benign', or 'Adult' categories, and further specified into 20 sublabels for detailed analysis [1]. This dataset is constructed to facilitate the development and comparative assessment of machine learning models. The dataset was curated to enable research in enhancing webpage classification, one component in optimizing web crawling and content filtering systems.

### *Tokenization Techniques*

The tokenization methods explored in this paper include:

- **Byte Pair Encoding (BPE):** BPE is a hybrid between character-level and word-level tokenization. It iteratively merges the most frequent pair of characters or character sequences, thereby reducing vocabulary size and capturing more information than individual characters [3]. We applied BPE to URLs to examine its effect on capturing token patterns significant for classification tasks.
- **Enhanced BPE:** This method extends BPE by integrating an initial dictionary of keywords generated by GPT-4 for each class [2]. The keywords enrich the BPE token dictionary, expected to refine the granularity with which URLs are tokenized and enhance classification performance.
- **Punctuation Split:** Utilizing regular expressions, specifically the pattern "(\w+|\S)", we tokenize on punctuation. This approach recognizes the structural nuances of URLs, which often contain meaningful delimiters such as periods and slashes.
- **Character-level N-grams:** We analyzed the performance of various n-gram levels, ranging from unigrams to longer spans of characters (1-gram, (1 to 3)-grams, and (3 to 6)-grams). This analysis aims to understand the impact of n-gram granularity on model performance, examining the trade-offs between the specificity of longer n-grams and the broader context captured by sequences.

### *Machine Learning Model*

Given the scope of this paper is to examine the impact of tokenization on URL classification, we selected the SGDClassifier from SKLearn as our machine learning model [9]. The choice of SGDClassifier is motivated by its computational efficiency and moderate performance across various text classification tasks. The SGDClassifier is well-suited for handling large-scale data and provides a consistent benchmark to evaluate the influence of different tokenization methods. By fixing the variable of the machine learning model, we isolate the effects of tokenization techniques on classification outcomes, thereby ensuring the focus of this study remains on the comparative analysis of the tokenization strategies employed.

## RESULTS

The heatmap visualization in Figure 1 shows the comparative performance of various tokenization techniques utilized for URL classification across three primary content categories. A key observation is the uniform struggle among

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863880



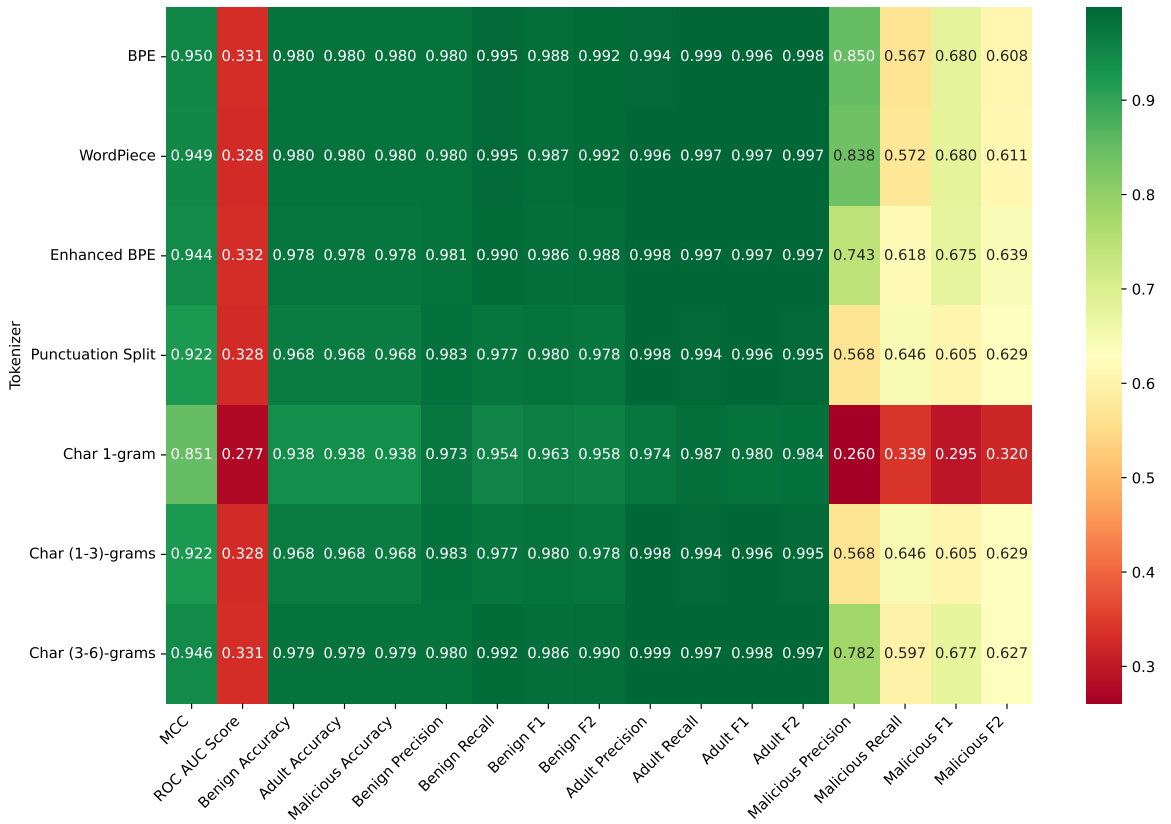



Figure 1: Performance of the tokenizers

all tokenization strategies to accurately classify 'Malicious' URLs. Despite this common challenge, certain tokenizers emerged with relatively superior performance in the 'Malicious' class, with **Byte Pair Encoding (BPE)**, **WordPiece**, and **Char (3-6)-grams** positioned as the frontrunners, respectively. Their ability to capture longer subword structures or sequences may attribute to their marginally better performance, suggesting a nuanced but high impact of token granularity on classification outcomes.

On the other hand, using **Char 1-gram** tokenization manifests as the least effective, particularly pronounced in its inability to classify 'Malicious' URLs. The results signify the insufficiency of singular characters to encapsulate the contextual complexity required for the identification of malicious content.

Furthermore, the ROC AUC Score, a probabilistic measure indicating a model's capability to discriminate between classes, is markedly low for all tokenization techniques. This uniform underperformance emphasizes a broader issue in the classification model's capacity to distinguish 'Malicious' URLs from others, reflecting a pivotal limitation within the current scope of tokenization approaches.

In contrast to the 'Malicious' class, tokenization techniques exhibit an excellent performance in classifying 'Benign' and 'Adult' URLs. This great performance indicates that the nature of tokens common in these categories is well-captured by the tokenizers, facilitating reliable classification. The differential success across the content categories underscores a key conclusion: while tokenization methods adeptly handle general content, they stumble in reliably identifying content with potentially harmful intent, where context and semantic complexity play an instrumental role, additionally, it is known that malicious URLs usually try to be similar to benign URLs to avoid being detected.

In light of the findings, it is obvious that the pursuit of enhanced tokenization strategies remains necessary. The quest entails refining the balance between token granularity and the semantic richness essential for the robust classification of web content, particularly for ensuring web crawlers' efficacy and safety in their navigational endeavors.

## DISCUSSION

The comparative analysis reveals significant insights into the performance landscape of various tokenization techniques in URL classification. Notably, the **Char 1-gram** tokenizer, despite its operational speed at a mere 0.02 milliseconds per URL Figure 2, demonstrates suboptimal performance metrics, with MCC values and F2 scores for the 'Malicious' class indicating insufficient precision and recall balance. This finding highlights the trade-off between prediction speed and classification robustness, particularly underlining the tokenizer's insufficiency in complex URL categorization that demands a richer contextual understanding.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863880

Meanwhile, the **Punctuation Split** tokenizer exhibits improvement in critical areas, including ROC AUC and MCC scores, over the Char 1-gram. At 0.09 milliseconds per URL, it encapsulates meaningful URL delimiters, hinting at the value of structural tokens in distinguishing between content categories. Similarly, the **Char (1-3)-grams** tokenizer maintains the same prediction time but advances in balancing precision and recall, except in the classification of 'Malicious' URLs, suggesting a need for an enhanced tokenization strategy to address URLs with malicious intent.

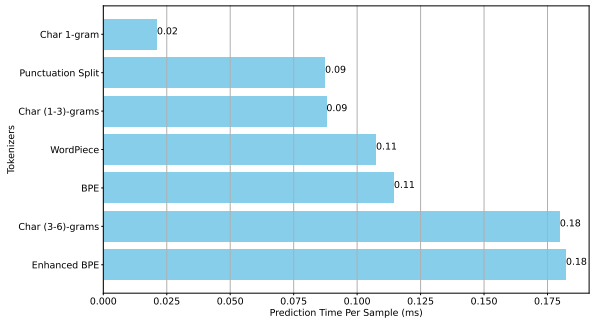



Figure 2: Comparison of Prediction Time Per URL Across Tokenizers

The **WordPiece** and **BPE** tokenizers, both clocking prediction times at 0.11 milliseconds per URL, achieve a admirable balance across evaluation metrics. However, their limitations become apparent in the 'Malicious' class, showing a challenge in detecting URLs of harmful web pages.

With a prediction time of 0.18 milliseconds per URL, the **Char (3-6)-grams** tokenizer shows potential in classifying 'Adult' and 'Benign' URLs but experiences a decline in performance when it comes to 'Malicious' URLs. This pattern suggests that while extended n-gram ranges might improve context capture, they may also result in overly specific tokens that lack generalizability.

Lastly, the **Enhanced BPE** tokenizer, also with a prediction time of 0.18 milliseconds, reveals a nuanced performance. It slightly improves upon BPE in the 'Adult' precision metric yet falls behind in critical areas such as 'Malicious' recall and F2 scores. The addition of GPT-4 generated keywords does not seem to uniformly enhance classification, particularly of 'Malicious' URLs, which remain challenging for all tokenizer models under study.

The practical application of these tokenization techniques within web crawlers has far-reaching implications. The efficiency of web crawlers is pivotal, as is their capability to sieve through the vast web content accurately. In this light, the findings of our study point to the necessity for carefully calibrated tokenizers that can adeptly handle URLs across varying content types without costing prohibitive computational costs.

In real-world applications, the decision to employ a particular tokenizer must be informed by the specific requirements of the web crawling task. The analysis highlights the need for a tokenizer that not only provides computational efficiency but also maintains high classification accuracy, especially for detecting 'Malicious' URLs. As web content continues to expand, the advancement of tokenization strategies will remain an essential area of research, with the objective of refining web crawlers to operate with enhanced precision and efficiency.

## CONCLUSION

Our comprehensive evaluation of tokenization techniques in URL classification has yielded several key findings. The study confirms that while faster tokenizers like **Char 1-gram** offer computational expediency, they fall short in effectively classifying URLs, particularly those that are malicious. In contrast, more complex tokenization strategies such as **BPE** and **WordPiece** demonstrate a commendable balance of speed and accuracy for 'Benign' and 'Adult' classes but exhibit limitations in discerning 'Malicious' URLs. Enhanced tokenizers like **Enhanced BPE**, despite incorporating domain-specific keywords, do not consistently improve classification outcomes, indicating the complex challenge of URL classification.

The pursuit of an optimal tokenization technique is complex and context-dependent. Our findings suggest that there is no one-size-fits-all solution; the choice of tokenizer must be tailored to the specific nuances of the classification task, with considerations for both computational efficiency and accuracy. For instance, while **Char (3-6)-grams** and **Enhanced BPE** offer detailed token representations, their slower prediction times may not be suitable for all web crawling contexts.

### *Suggestions for Future Research Directions*

Future research should explore the integration of multiple tokenization techniques, potentially leveraging the strengths of each to improve classification performance, especially for the elusive 'Malicious' class. Additionally, investigating the incorporation of semantic analysis and contextual understanding into the tokenization process could yield significant advancements. Another promising direction is the application of deep learning models that could learn optimal token representations in an end-to-end manner, potentially overcoming the limitations of predetermined tokenization schemes.

Continued exploration in tokenization techniques is critical as web content evolves. The development of more adaptive, context-aware models could greatly enhance the precision of web crawlers and their utility in navigating the ever-growing expanse of the internet.

https://doi.org/10.5281/zenodo.13863880



## ACKNOWLEDGEMENTS

Federal Ministry of Education and Research

This work is part of the OpenWebSearch.eu project, funded by the EU under the GA 101070014, and part of the CAROLL project, funded by the German Federal Ministry of Education and Research (BMBF) under the funding code 01|S20049.

## REFERENCES

[1] Mohammed Al-Maamari, Mahmoud Istaiti, Saber Zerhoudi, Michael Dinzinger, Michael Granitzer, and Jelena Mitrovic. A Comprehensive Dataset for Webpage Classification (Part 1: Adult & Malicious), March 2024.

[2] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[3] Philip Gage. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38, 1994.

[4] Zaid Alyafeai, Maged S Al-shaibani, Mustafa Ghaleb, and Irfan Ahmad. Evaluating various tokenizers for arabic text classification. *Neural Processing Letters*, 55(3):2911–2933, 2023.

[5] Gyeongmin Kim, Junyoung Son, Jinsung Kim, Hyunhee Lee, and Heuiseok Lim. Enhancing korean named entity recognition with linguistic tokenization strategies. *IEEE Access*, 9:151814–151823, 2021.

[6] Michael Granitzer, Stefan Voigt, Noor Afshan Fathima, Martin Golasowski, Christian Guetl, Tobias Hecking, Gijs Hendriksen, Djoerd Hiemstra, Jan Martinovič, Jelena Mitrović, et al. Impact and development of an open web index for open web search. *Journal of the Association for Information Science and Technology*, 2023.

[7] Duygu Taylan, Mitat Poyraz, Selim Akyokuş, and Murat Can Ganiz. Intelligent focused crawler: Learning which links to crawl. In *2011 International Symposium on Innovations in Intelligent Systems and Applications*, pages 504–508. IEEE, 2011.

[8] R Rajalakshmi and Chandrabose Aravindan. Web page classification using n-gram based url features. In *2013 fifth international conference on advanced computing (ICoAC)*, pages 15–21. IEEE, 2013.

[9] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13871624

# ENRICHING SCIENCE SEARCH WITH THE OPEN SEARCH FRAMEWORK MOSAIC

A. Nussbaumer[1], S. Gürtl[1], J. Honeder[1], T. Hecking[2], C. Gütl[1]
[1]Graz University of Technology, Graz, Austria
[2]German Aerospace Center, Cologne, Germany

*Abstract*

This paper presents a concept how the search in digital science libraries can be enriched with contextual science information. Digital science libraries often consist of collections of scientific publications that can be explored and searched in a structured way by using metadata to search and filter. We propose a concept of how the search can be extended with web content that includes related information, such as conferences, researchers, research projects, research data, or general knowledge on the topic. Thus, contextual information not included in a digital library can be added to a search result. This concept is demonstrated by an application in the field of earth observation that integrates earth observation catalogues with web content related to environmental emergency events.

## INTRODUCTION

Digital libraries are important in the academic field, as they enable users and especially researchers to search and find scientific publications. Their core functionalities include the storage and digital objects and a retrieval system for the stored objects [1]. Digital libraries in the science field can be classified according to the operator or publisher, the amount, and topical domain of the digital publications they offer. For example, university libraries offer books and publications in the fields relevant to the institution. Journals and scientific publishers offer their published content. General digital libraries, such as Google Scholar, have indexed scientific literature available at other digital libraries.

Basic features of digital libraries include effectiveness (hosting of required documents), efficiency (retrieval accuracy), accessibility (low access barriers), usability, software quality, and satisfactory (meeting the users' information need) [1]. Information retrieval and indexing key information is an essential issue of digital libraries [2]. Furthermore, the semantic web and social networking technologies help improve the usage of digital libraries [3].

Scientific Digital Library Systems aim to manage, search, and retrieve scientific data, such as publications, research data, and other scientific outcomes. Usually, they contain digital objects that have been approved and created in the past. In contrast, web data related to science contain more recent and up-to-date information, such as events and news. However, researchers often need information from both scientific publications and active research activities. Thus, it becomes obvious that these types of information should be combined into an integrated system where the user can search and retrieve all types of information.

In order to address this issue, this paper presents an approach and implementation that enriches the search in digital libraries with information from the web related to the knowledge domain of the digital library. The main contribution of this paper consists in a concept based on semantic information and metadata that connects digital libraries with web-based information sources and increases efficiency and user satisfaction in the search process.

## SCIENCE SEARCH ENRICHMENT

The Modular Search Application Based on Index Fraction (MOSAIC) [4] is a framework and generic search application that makes index partitions searchable. Index partitions are small or medium sized indices containing web documents related to a certain topic or for a particular purpose. These partitions are being created using the OpenWebSearch.eu infrastructure and contain various metadata, such as geocoordinates, topics, and genres of the web documents [5]. MOSAIC is built upon the Prototype Search Application [6] and provides an Application Programming Interface (API) to search index partitions and to filter the results according to the above mentioned metadata.

The overall approach of integrating scientific digital libraries with web search is depicted in Figure 1. On the one hand, an existing digital library system is used that provides an API for searching and retrieving digital objects. On the other hand, the MOSAIC framework is used to make web content available in the search and retrieval process. On a conceptual level the integration is realised by connecting search results with joint metadata.

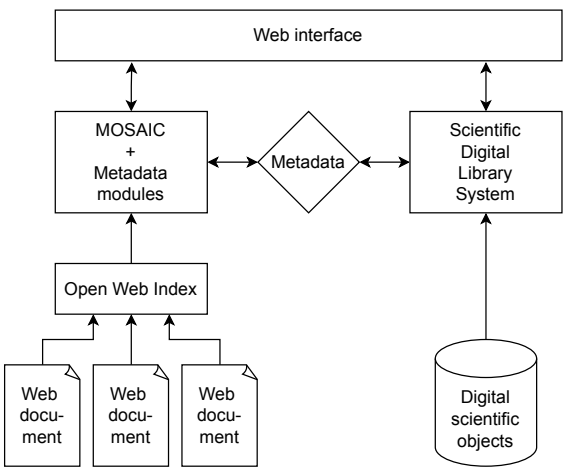



Figure 1: Conceptual design of science search context.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13871624



The selection of the joint metadata depends on the metadata provided by the digital library. Typical metadata of scientific digital libraries are author, date, and keywords of publications, but also geographic and temporal information. In order to semantically join search results by metadata, the same description language must be supported by MOSAIC. Either respective metadata are provided by datasets from the Open Web Index (OWI) or they have to be generated in a pre-processing step. Currently, the OWI datasets provide geo-locations and domain topics.

### *Metadata handling in MOSAIC*

The management of metadata in MOSAIC is conceptually built around a modular architecture, which allows for the separation of concerns and the independent processing of different metadata aspects. Central to this system are various specialised modules, each designed to handle distinct types of metadata. This modular design, technically realised using Apache Maven modules, allows users to enable or disable modules easily via a configuration file.

In MOSAIC, metadata is utilised through a dual approach involving metadata filtering and enrichment. Metadata filtering allows users to apply specific criteria to refine search results based on metadata columns of the OWI partition. This process can be executed directly using the metadata columns or through more advanced algorithms, which leverage specific metadata columns for sophisticated filtering. Once the filtering is complete, MOSAIC enriches the search results from the Lucene index by adding the specified metadata. In its current version, MOSAIC encompasses various modules which function within a structured API framework to facilitate interaction with the central index and search components.

The *core* module in the MOSAIC framework manages fundamental metadata fields such as document titles, URLs, and languages. It ensures that basic information is consistently available for all search results. This module interacts with other specialised modules, and provides a foundation for further metadata processing. Geographical metadata is managed by the existing *geo module*, which enriches search results with detailed location-based information. This module includes metadata fields for locations such as coordinates, location names, and administrative regions. For instance, a search result might include the exact latitude and longitude of a place, the city or region name. Filtering is achieved through a bounding box mechanism, where users specify the western, eastern, northern, and southern boundaries to limit search results to a specific geographic area. Additionally, the geographical information is represented in the search results to offer users a clear spatial context that enhances the relevance of the data. Eventually, the *keywords* module focuses on integrating relevant keywords extracted from the full text of the respective document.

Additional metadata can be incorporated by either extending an existing module if it aligns thematically or creating a new metadata module. If the new metadata is related to an existing module's theme, it can be added by updating the module's configuration and processing logic. Alternatively, users can define a new module that specifies the additional metadata fields and processing logic required. If filtering using the additional metadata is desired, users need to handle the processing of HTTP query parameters, particularly when the filtering goes beyond straightforward equality comparisons. Subsequently, advanced filtering logic can be implemented by overriding the provided methods if necessary. Moreover, to include additional metadata columns in search results, they must be specified in the module, with advanced processing for enrichment implemented as needed.

A simple web interface demonstrating the use of metadata with MOSAIC is shown in Figure 2. Beside a search term, the user can enter a geographic region and keywords to filter the search result. The geographic filter allows to specify a geographic rectangle using longitudinal and latitudinal boundaries. The location information (coordinates) of the indexed web documents is used to filter the search result, so that only documents show up in the result that have locations within the specified geographic rectangle. Similarly, the user can specify keywords that are used to compare and filter web documents that have keywords in their metadata. The web documents in the search result include both geographic locations and keywords.

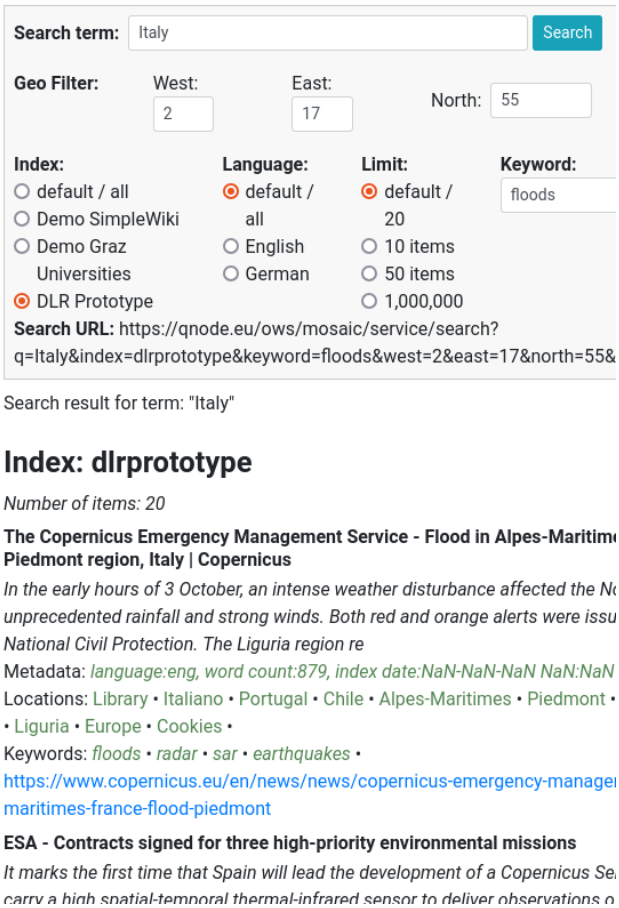



Figure 2: The web interface of MOSAIC showing filters for geo-locations and keywords.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13871624

### *Integration of search results*

In order to enrich search in digital libraries with web documents, an integration consisting of several steps has to be undertaken. First a web index has to be created consisting of web documents that are related to the digital library. The OpenWebSearch.eu project provides a technical infrastructure to create a web index that can be imported into MOSAIC. The specification of the web index is currently done by specifying a list of URLs of the web documents that should be included. In the future, the OWI will provide means to specify the content of an index with metadata, such as topic, language or domain.

The second step consists in the alignment of the joint metadata. This depends on the metadata that the digital library provides in its search API and the possibilities to integrate the same metadata into the web index and MOSAIC. For example, keywords used in a digital library might be structured according to a classification scheme. The keywords used in MOSAIC need to have the same structure. Thus, in a pre-processing step the web documents are analysed and tagged with keywords of the same classification scheme. Similarly, authors of publications can be extracted from web documents and tagged. Geographic information can be used as joint metadata if provided by the digital library.

In the third step a joint user interface has to be created that integrates the search over both the digital library and MOSAIC. A straightforward approach consists in the adaptation of the existing web interface of MOSAIC. While the built-in web interface queries the MOSAIC service, an additional query can be performed to a digital library. Thus, a federated search approach is realised. The web interface should then support the use of joint metadata implementing a kind of faceted search. For example, a keyword or author available in an item of the search result can be used for the new search query that retrieves results from both sources.

## APPLICATION

The concept has been tried out in a use case in the field of Earth Observation (EO) and environmental search [7]. This use case includes a digital library of the German Aerospace Center (DLR) that contains publications in this research field. Furthermore, EO Catalogues are included that contain environmental information generated by satellites in the form of geographic maps. Items of these data sources are tagged with keywords and geo-coordinates. An integrated dashboard has been created that coordinates the search over these data sources and bundles the search results.

An index partition has been created that contains app. 500 news web pages related to natural disasters. This index has been integrated into MOSAIC and made available through the MOSAIC web service. The web documents are tagged with geographic information, namely coordinates of the mentioned locations in the documents. The dashboard of the application displays the search result of the web documents retrieved from MOSAIC (see Figure 3). This view also includes information of the contained locations and a button to show them in a map.

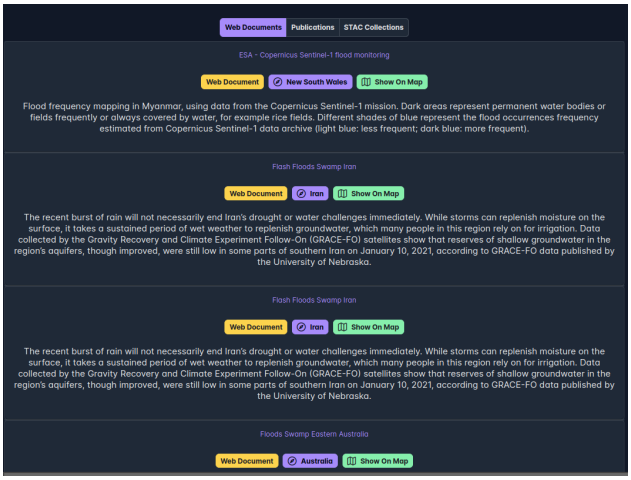



Figure 3: Search results of web pages retrieved from MOSAIC and displayed in the dashboard of the application.

The corpus of scientific publications included in the prototype application is provided by the German Aerospace Center (DLR). An internal snapshot of a scientific literature database which includes scientific publications in the domain of EO research has been imported into a Knowledge Graph Database. The initial data corpus is reduced in size (app. 12 000 documents) by filtering out non-relevant scientific publications that are not connected to the domain of natural disaster. In a pre-processing step keywords and authors are extracted from the raw attributes and added as explicit metadata of the documents. Thus, the Knowledge Graph can be queried by using search terms, but also keywords and authors. The search result is displayed in Figure 4 on the left side, where keywords and authors of each publication are highlighted.

The dashboard shown in Figure 4 displays query fields on top, the search results on the left side, and the map on the right side. The query fields accept general search terms, but also keywords and authors. The search result area on the left side includes three tabs to switch between the results of the three sources. By clicking on a keyword or author of an item in the search results, the respective word is automatically added to the query fields and used for a federated search over all sources. The map on the right side shows the locations of the results geographically. This map also enables to select a geographic area to filter the search.

The third information source of this application consists of geographical information coming from EO systems. Data is extracted from existing EO catalogues, such as Planetary Computer [1] and Terrabyte STAC API [2], and imported into the Knowledge Graph. These items contain metadata, such as title, keywords, origin, spatial extent, and time interval.

[1] https://planetarycomputer.microsoft.com/
[2] https://stac.terrabyte.lrz.de/browser/

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13871624



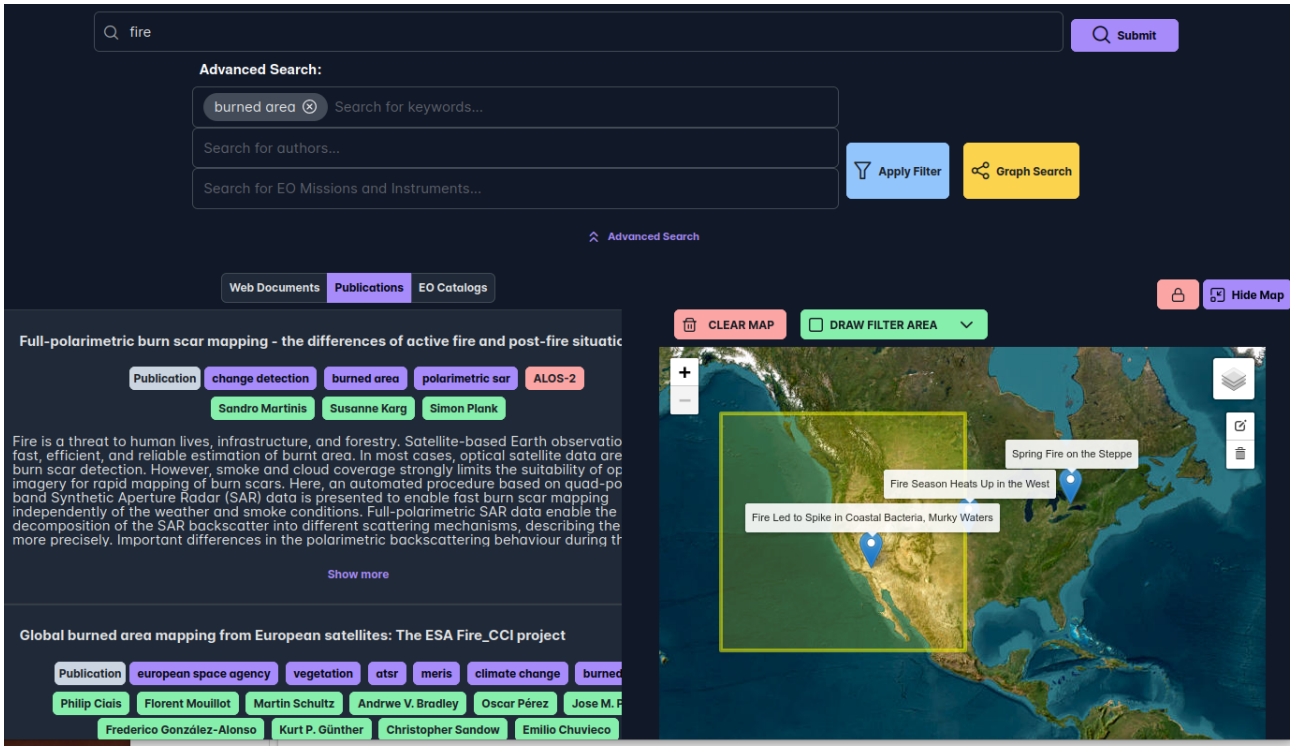



Figure 4: The Dashboard of the example application. The search fields are entered on top, the search results are displayed on the left side, and the locations of the search results are displayed on a map on the right side.

## CONCLUSION AND OUTLOOK

The main contribution of this paper consists in an approach to enrich search in a digital scientific library by adding web documents to the search result. Similar to a federated search approach, digital libraries and web documents are queried with search terms, keywords, geo-location, and other metadata. A semantic relation established through the same taxonomies of metadata is used to harmonise the search over different data sources. This approach brings the advantage to supplement past information of scientific publications with more recent knowledge, such as information of events, conferences, or research activities. In order to put this approach into practice, the software MOSAIC has been developed that allows to use indexed web data and integrate the search for metadata. The applicability of this approach is demonstrated with the example of an environmental search application.

Future work will include further development of the example application. Keywords will be structured to a taxonomy, so that the same set of keywords is used by all data sources. Second, focused crawling will be developed that allows to automatically create index partitions related to the science field.

## ACKNOWLEDGEMENTS

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

## REFERENCES

[1] V. K. Sharma and S. K. Chauhan, "Digital library challenges and opportunities: An overview," *Library Philosophy and Practice*, 2019. https://digitalcommons.unl.edu/libphilprac/3725

[2] E. A. Fox, M. A. Gonçalves, and N. A. Kipp, "Digital libraries," in *Handbook on Information Technologies for Education and Training*. 2002, pp. 623–641.

[3] A. Tella, V. Okojie, and O. T. Olaniyi, "Social bookmarking tools and digital libraries," in *Handbook of Research on Managing Intellectual Property in Digital Libraries*. 2018, pp. 396–409.

[4] S. Gürtl, *MOSAIC: Empowering a Modular Framework for Configurable and Tailored Web Search based on an Open Web Index*, 2024. https://diglib.tugraz.at/diplomaTheses

[5] M. Granitzer *et al.*, "Impact and development of an open web index for open web search," *Journal of the Association for Information Science and Technology*, 2023. 10.1002/asi.24818

[6] A. Nussbaumer, R. Kaushik, G. Hendriksen, S. Gürtl, and C. Gütl, "Conceptual Design and Implementation of a Prototype Search Application using the Open Web Search Index," 2023. 10.5281/zenodo.10636166

[7] J. Honeder, *Bridging Science and Web: An Open Search Application for Earth Observation and Environmental Research*, 2024. https://diglib.tugraz.at/diplomaTheses

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13871740

# NeutrinoReview: CONCEPT PROPOSAL FOR AN OPEN SOURCE REVIEW MANAGEMENT TOOL

E. Sandner*[1,3], I. Jakovljevic[1], A. Simniceanu[2], L. Fontana[2], A. Henriques[1], A. Wagner[1], C. Gütl[3]
[1]CERN, 1211 Geneva, Switzerland
[2]WHO, 1211 Geneva, Switzerland
[3]Graz University of Technology, 8010 Graz, Austria

*Abstract*

This paper proposes the design of NeutrinoReview, a tool that aims to guide scientists through the systematic review process. Based on an extensive literature review and discussions with experienced scientists from WHO, a conceptual design is suggested that combines state-of-the-art solutions and further reduces human effort by seamlessly integrating them into a single solution. This concept serves as an initial step towards a comprehensive solution that automates the whole systematic review process with a continuous user interface, thereby significantly improving user experience. This paper aims to present the idea of NeutrinoReview to the broader research community at a very early stage, with the aim of further refining and expanding this promising concept in an open and collaborative manner.

## INTRODUCTION

In 1753, the medical researcher James Lind conducted the first systematic literature review [1]. Since then, systematic reviews (SRs) have become a common practice in supporting evidence-based medicine and are widely used as a research methodology in various fields beyond healthcare. For example, Kitchenham constructed specific guidelines for performing SR in software engineering research. He describes a systematic literature review as a method for identifying, evaluating, and synthesizing all research relevant to a specific research question, topic area, or phenomenon of interest [2].

Since SRs aim to identify all relevant research, this research methodology demands significant human effort. Conducting a SR requires a substantial amount of time, typically ranging from 6 months to 2 years [3, 4].

To minimize this workload, several research efforts aimed to apply information technology, especially artificial intelligence (AI) and natural language processing (NLP) to support experts in the SR process. While certain steps have been effectively automated, others require additional research and advancement [5].

To facilitate collaboration and enable different automation tools to work together more effectively, the International Collaboration for the Automation of Systematic Reviews (ICASR) was formed. In their initial meeting, they formulated the 'Vienna Principles' to propose that solutions should be offered as specialized modules designed for specific tasks, compatible with standardized file types, enabling researchers to assemble a sequence of tools that best fit their specific needs [6].

This approach offers adaptability but requires exporting results from each stage and then re-importing them into subsequent tools, which represents a redundant extra effort. Additionally, it requires that experts familiarize themselves with a variety of tools and their distinct user interfaces.

To address these challenges, this paper suggests a conceptual design of a comprehensive SR automation tool that eliminates the need for additional tools to conduct SRs. This novel tool, named NeutrinoReview, will offer intuitive and continuous user interaction throughout the entire process, starting from the development of the project protocol and continuing through to data extraction. Its objective is to automate a maximum number of tasks and enhance human efficiency in areas where complete automation is not feasible. By integrating the most advanced automation approaches for each step and eliminating transitions between them, this method aims to contribute to faster evidence-based research.

One use case may be the ARIA project, a joint venture between CERN and WHO. The objective of this project is to develop an online tool[1] to quantify the risk of SARS-CoV-2 airborne transmission to inform nonpharmaceutical risk reduction measures in residential, public and health care settings. The underlaying model, as estimated internally, is based on more than 100 different parameters which should be supported with evidence based research. Here, the suggested tool could help to execute the required SRs in a feasible timeframe.

However, NeutrinoReview will not be developed to be limited to internal use but rather should become an open-source web application, freely accessible to researchers worldwide and designed for use in SRs across various fields. Furthermore, the motivation for developing the proposed tool includes the fact that with minor adjustments, it can be used for free and unbiased science searches. By removing the restriction to academic papers, a fully automated SR process could greatly contribute to an unbiased, free, and open web search pertinent to an audience beyond academia.

The remainder of the paper continues by describing existing automation approaches for several steps of the SR process, followed by an outline of the applied methodology. Subsequently, the design decisions and conceptual architecture for NeutrinoReview are described. Then, the existing limitations and paths for future work are explained. The pa-

* elias.sandner@cern.ch

[1] https://partnersplatform.who.int/tools/aria

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13871740



per concludes by encouraging interested parties to contribute to this endeavor.

## BACKGROUND AND RELATED WORK

Van Dinter et al. systematically reviewed existing literature about the automation of SRs [5]. They identified 41 primary studies focusing on the automation of one or more steps of the SR. Most of the identified studies targeted the (semi-)automation of the screening process, and no solution aimed to cover more than three steps simultaneously. Thus, combining tools designed for specific steps is a common practice. This approach can allow a team of experienced researchers to reduce the duration of a full SR from several months to less than two weeks [7].

Based on the insights stated above, the remainder of this section outlines the SR process and aims to highlight how automation tools can support the human expert at each step.

### *Project Initiation and Data Retrieval*

This phase starts after the research question is defined and an SR is determined as the preferred research methodology. First, a project protocol is developed which offers a detailed description of the research objective, hypothesis, search terms, and criteria to include and exclude studies. The protocol is designed to guarantee reproducibility, making sure that following its prescribed guidelines leads to consistent results, irrespective of the researcher. Templates can help reduce human effort in creating project protocols. This approach was also adopted by Clark et al. to streamline this stage of the process [7].

Search terms, typically determined through collaboration with a librarian or information manager, are then used to formulate a search strategy that prioritizes broad sensitivity and comprehensive data retrieval, rather than being overly specific and risking omission of relevant documents. Subsequently, the formulated strategy is employed to conduct searches across multiple databases. To reduce the time effort at this stage, two tools from the Systematic Review Accelerator (SRA)[2], a software suite developed at Bond University, were utilized [7]:

- A word frequency analyzer, assisted in developing the search string.
- A tool called 'polgot search translator' was instrumental in adapting the search syntax for compatibility with various databases.

Although their approach involved manually querying multiple databases, amalgamating searches from various source databases could further streamline the process. Metta [8] and EBM Search [9] serve as examples of meta search engines that facilitate SRs.

Prior to further processing, the retrieved studies must be deduplicated, a task that can be accomplished using one of the various available software solutions. The tool Deduklick, for instance, excels in this task with an average recall of 99.51% and a precision of 100% [10].

### *Screening*

To determine studies that comply with the specified inclusion and exclusion criteria, each retrieved study is subjected to a thorough screening process. Initially, the researcher assesses only the title and abstract of each paper to eliminate the bulk of the nonrelevant papers and then proceeds to a detailed examination of the full text for comprehensive evaluation. To streamline the screening process Clark et al. utilised RobotSearch [11] during the title and abstract screening phase to filter out documents that are definitely not randomized controlled trials (RCTs), the specific study type to which this SR was limited [7]. However, it is important to consider that the exclusive focus on RCTs notably simplified the screening process. In addition to prefiltration, review management tools are widely utilized for the screening phase of SRs. Their key feature is a user-friendly interface enabling experts to include or exclude studies quickly using hotkeys.

As analysed in a previous publication, most existing tools integrated supervised machine learning approaches like classification and priority ranking to reduce the workload in the screening process [12].

Due to their ability to classify text without requiring fine-tuning or labeled data, general-purpose Large Language Models (LLMs) represent a promising technology for automating literature screening in systematic reviews. Reference [13] evaluated OpenAI's GPT-3.5 Turbo for title and abstract screening, finding that assigning the model roles such as an "experienced researcher" improved performance. Although the model's sensitivity increased with certain prompts, none reached Cochrane's required sensitivity (0.99) [14]. The model performed comparably to less experienced human screeners, despite an expert missing 19% of relevant papers. Similarly, [15] assessed GPT-3.5 on six datasets, finding a weighted average accuracy of 0.907 and a sensitivity for included papers of 0.764. Considering the rapid developments in this field, further performance improvements also in the use case of screening automation can be expected.

While many review management tools provide features for uploading full-text PDF documents, tools like LiteRev [16] automatically retrieves full texts, either from the metadata directly or via a link in the metadata. In the latter case, the text is automatically extracted from the PDF file. Additionally, most reference management software, such as EndNote[3], offers semi-automated features for full-text retrieval.

### *Data Extraction*

In [7] the sole technical tool utilized for data extraction was a digital spreadsheet, which functioned as a structured form to systematically capture the necessary data. ExaCT [17] is a tool that uses an information extraction engine to

[2] https://sr-accelerator.com

[3] https://endnote.com

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13871740

extract fragments that best describe the characteristics of the trial to support the expert in the data extraction phase.

Ultimately, these gathered insights are documented, culminating in the SR being prepared for submission and publication.

### *Comprehensive Solutions*

To conclude this section, automation tools are available for various stages of the SR process. However, to the best of our knowledge, no existing tool comprehensively facilitates the integration of automation technologies across the entire SR process. Therefore, this paper proposes the development of a tool that combines the most advanced automation approaches and leverages the opportunity to further streamline the process by integrating them into a single, comprehensive tool.

## METHODOLOGY

Extensive literature research and discussions with experienced scientists from CERN and WHO, as well as observing their working methods, allowed a detailed analysis of the currently very labor intensive process, highlighted the pain points, and motivated the development of a more advanced software tool that guides scientists through the whole process of a SR. By combining best-practice solutions for individual tasks, as discussed in the previous section, a conceptual architecture was developed for a tool that further streamlines the process and enhances user experience by providing an all-in-one solution. The emphasis was placed on offering a generic solution that is both domain-independent and modular. In addition, tasks that require further research for automation have been identified.

## CONCEPTUAL DESIGN

Based on existing automation solutions targeted at certain tasks, this section describes the concept of a software tool that addresses the whole process, starting from the project protocol development and continuing through to data extraction. Only the initial review of the literature and the presentation of the results in a scientific paper are not included in the concept. This is due to two reasons. First, for the ARIA project, no initial literature research is required to identify a suitable research question for an SR. The topic is predetermined by the necessary parameters used in the model. Second, within the ARIA project, the goal is to retrieve evidence-supported numerical values for those parameters. The publication of the results will be considered, but it is not a priority. Furthermore, addressing these steps for a wide audience may be challenging due to domain-specific requirements.

Figure 1 illustrates the conceptual designs suggested for a novel review management tool. It consists of five components that will be detailed subsequently.

### *Project protocol Development*

After creating a new SR instance, the user is presented with a user interface that guides them through the creation of the project protocol. Based on the field in which the SR is created, the input form adapts to the respective domain. For instance, in health-related fields, it may be based on the registration form of the International Prospective Register of Systematic Reviews (PROSPERO) [18] to ensure that all information is collected at the point of potential registration.

Furthermore, the concept includes a feature that assists the user in creating the search string, which is one of the most critical elements of the protocol regardless of the applied field. This feature could be based on a word frequency analyzer similar to the SRA tool or on generative AI technology.

### *Data Retrieval*

To reduce the chance of omitting relevant studies, various academic libraries are queried in SRs. The choice of libraries is determined by the responsible researcher and varies across different SRs. For example, PubMed[4], Medline[5], Embase[6], and Cochrane[7] are the most relevant academic libraries for conducting the SRs planned for the ARIA project. Therefore, while these libraries will be prioritized, NeutrinoReview will not be limited to them.

Provided that the academic libraries of interest offer the option to query the database through an Application Programming Interface (API), NeutrinoReview will amalgamate the search across various sources. Since the search string is already known to the tool, this can streamline the database search task to a single click. Additionally, the integrated deduplication software immediately begins preparing the retrieved citations for the screening phase.

### *Title and Abstract Screening*

As previously described in [12], a wide range of tools to support the title and abstract (TiAb) screening phase is available. However, it was also emphasized that the reliability of existing tools is not always transparent and further automation could lead to additional time savings, as this is the most critical task of the SR process.

Given the rapid evolution of foundational LLMs and their demonstrated performance in automating title and abstract screening, this technology emerges as the most promising approach for streamlining the screening phase. An additional advantage is that these models facilitate automation without being restricted to specific research questions or eligibility criteria.

Therefore, NeutrinoReview will reduce the human workload in the TiAb screening phase by integrating LLM-based prefiltration, leaving only those records for human review

[4] https://pubmed.ncbi.nlm.nih.gov
[5] https://www.medline.com
[6] https://embase.com
[7] https://www.cochranelibrary.com

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13871740



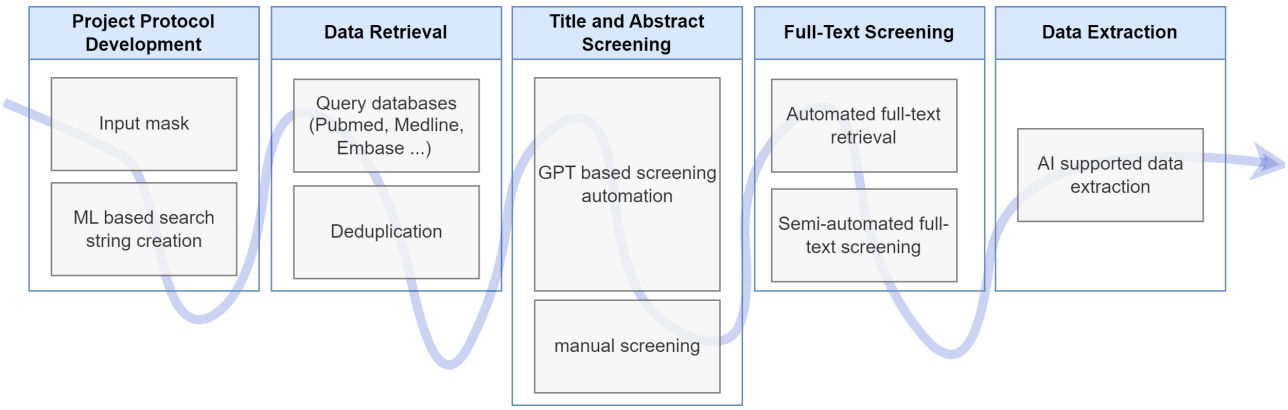



Figure 1: Conceptual architecture of a review management system encompassing the entire systematic review process.

where the underlying model cannot make a definitive exclusion decision. This approach minimizes the risk of excluding relevant papers while still significantly reducing the human workload.

The development of this screening automation necessitates extensive prompt engineering and the comparison of multiple models to identify the most effective one. Additionally, before integrating LLM-based automation, thorough evaluation on human-annotated data is essential to ensure its reliability in replacing human screening.

Finally, the remaining citations have to be manually screened. In this step, the user will be supported with semi-automation features, already approved in existing solutions as summarized in a previous publication [12].

### *Full-Text Screening*

Following the TiAb screening phase, NeutrinoReview will retrieve the full text of included citations, provided that they are open access and available online. Full-text versions purchased by the user can be uploaded for further processing in NeutrinoReview.

By utilizing various information retrieval and NLP methods, including foundational LLMs, relevant passages from the documents will be extracted, and recommendations for inclusion or exclusion decisions will be provided based on the defined criteria. However, in the full-text screening phase, which typically involves a smaller number of citations, the final decision will always be made by the human expert to ensure that no relevant citation is missed at this final stage.

After completing the screening phase, NeutrinoReview will display the results through various visualizations, including diagrams showing the distribution of different exclusion reasons and flow diagrams reporting the applied filtration process.

### *Data Extraction*

In addition to facilitating the data filtration process, NeutrinoReview will be able to assist scientists in extracting data from relevant studies. The software helps create custom templates for data extraction tailored to the specific SR. In addition, it will automatically extract pertinent information from the full text in the form of value, unit, and context, and will offer suggestions for completing the form. To maintain the quality standards of a SR, these suggestions require validation and approval from a human expert.

As a novel tool, NeutrinoReview guides users through the entire SR process and facilitates the extraction of relevant data at all stages. This includes generating the protocol in both .docx and .pdf formats, maintaining lists of citations in RIS and .csv formats at all stages, and providing the results of the data extraction in at least .csv format.

### *Comprehensive Solution*

Offering the described (semi-)automation systems within a single tool allows for a smooth transition between individual steps. It completely eliminates the need to export results from one tool and import them into another, which not only saves time but also reduces the potential for human error. Furthermore, providing one comprehensive tool is expected to improve the user experience, as all steps are integrated within a consistent UI. Additionally, this approach is anticipated to decrease the training effort required to efficiently utilize the integrated automation technologies.

## LIMITATIONS

The presented concept, while promising, confronts several limitations that could affect its feasibility and further development. The reliance on libraries that provide APIs and accessible full-text articles may limit its seamless application, potentially leading to increased human effort. To ensure the quality of SRs, human oversight remains necessary for interpreting complex data and making final decisions. Moreover, ethical considerations and potential biases introduced by AI-based automation require continuous monitoring and adjustment to ensure fair and unbiased reviews. Addressing these issues is crucial for the development of a robust and user-friendly review management system.

https://doi.org/10.5281/zenodo.13871740



## FUTURE WORK

Regardless of the presented concept, continuous improvement in the automation of all steps of the SR process remains necessary. Tsafnat et al. outlined potential directions for each step [17]. To achieve complete automation of the process, enhancing the automation of full-text screening and data extraction is particularly important, with recent advancements in large language models (LLMs) offering promising approaches.

To realize the concept presented, organizing the implementation as an open-source project could be instrumental. This approach would encourage contributions from institutions interested in utilizing such a tool and establish partnerships with academic institutions and providers of academic libraries, potentially accelerating development and addressing existing limitations.

Furthermore, once a prototype is developed, its real-world applicability must be evaluated across various research domains. This evaluation should include user experience studies and assessments of time savings and effects on the quality of SRs, providing valuable insights for further refinement of the tool.

## CONCLUSION

In conclusion, this paper suggests the architecture of a novel tool called NeutrinoReview, an open-source software designed to streamline the entire SR process within a single tool. By utilizing state-of-the-art automation technologies and integrating them seamlessly, the proposed concept aims to significantly reduce the substantial human effort currently required to conduct SRs. The CAiMIRA team at CERN, in collaboration with their partners at the WHO, will continue their research and development efforts, with the goal of presenting a first prototype in the near future. This paper invites experts from various fields to challenge the proposed design, suggest improvements, or contribute in any capacity to the implementation and evaluation of this open-source project.

## ACKNOWLEDGEMENTS

The study was done as part of the joint CERN and WHO ARIA[8] project, that is funding the PhD project, in the context of which this paper was written. Furthermore, we greatly thank OpenWebSearch.EU[9] project and the members for their help and support with this publication.

## REFERENCES

[1] M. Bartholomew, "James Lind's treatise of the scurvy (1753)", *Postgraduate Medical Journal*, vol. 78, no. 925, pp. 695-696, 2002, Oxford University Press.

[2] S. Keele and others, "Guidelines for performing systematic literature reviews in software engineering", Technical report, ver. 2.3 EBSE Technical Report. EBSE, 2007.

[3] R. Ganann, D. Ciliska, H. Thomas, "Expediting systematic reviews: methods and implications of rapid reviews", *Implementation Science*, vol. 5, pp. 1-10, 2010, Springer.

[4] S. Khangura, K. Konnyu, R. Cushman, J. Grimshaw, D. Moher, "Evidence summaries: the evolution of a rapid review approach", *Systematic Reviews*, vol. 1, pp. 1-9, 2012, Springer.

[5] R. Van Dinter, B. Tekinerdogan, C. Catal, "Automation of systematic literature reviews: A systematic literature review", *Information and Software Technology*, vol. 136, pp. 106589, 2021, Elsevier.

[6] E. Beller, J. Clark, G. Tsafnat, C. Adams, H. Diehl, H. Lund, M. Ouzzani, K. Thayer, J. Thomas, T. Turner, et al., "Making progress with the automation of systematic reviews: principles of the International Collaboration for the Automation of Systematic Reviews (ICASR)", *Systematic Reviews*, vol. 7, pp. 1-7, 2018, Springer.

[7] J. Clark, P. Glasziou, C. Del Mar, A. Bannach-Brown, P. Stehlik, and A. M. Scott, "A full systematic review was completed in 2 weeks using automation tools: a case study," Journal of Clinical Epidemiology, vol. 121, pp. 81-90, 2020.

[8] N. R. Smalheiser, C. Lin, L. Jia, Y. Jiang, A. M. Cohen, C. Yu, J. M. Davis, C. E. Adams, M. S. McDonagh, W. Meng, "Design and implementation of Metta, a metasearch engine for biomedical literature retrieval intended for systematic reviewers", *Health Information Science and Systems*, vol. 2, pp. 1-9, 2014, Springer.

[9] P. J. Bracke, D. K. Howse, S. M. Keim, "Evidence-based Medicine Search: a customizable federated search engine", *Journal of the Medical Library Association: JMLA*, vol. 96, no. 2, pp. 108, 2008, Medical Library Association.

[10] N. Borissov, Q. Haas, B. Minder, D. Kopp-Heim, M. von Gernler, H. Janka, D. Teodoro, P. Amini, "Reducing systematic review burden using Deduklick: a novel, automated, reliable, and explainable deduplication algorithm to foster medical research", *Systematic Reviews*, vol. 11, no. 1, pp. 172, 2022, Springer.

[11] I. J. Marshall, A. Noel-Storr, J. Kuiper, J. Thomas, and B. C. Wallace, "Machine learning for identifying randomized controlled trials: An evaluation and practitioner's guide," Research Synthesis Methods, vol. 9, no. 4, pp. 602-614, 2018.

[12] E. Sandner, C. Gütl, I. Jakovljevic, and A. Wagner, "Screening automation in systematic reviews: Analysis of tools and their machine learning capabilities," in *dHealth 2024*, pp. 179-185, 2024, IOS Press.

[13] O. K. Gargari, M. H. Mahmoudi, M. Hajisafarali, and R. Samiee, "Enhancing title and abstract screening for systematic reviews with GPT-3.5 turbo," BMJ Evidence-Based Medicine, vol. 29, no. 1, pp. 69-70, 2024.

[14] J. Thomas, S. McDonald, A. Noel-Storr, I. Shemilt, J. Elliott, C. Mavergames, I. J. Marshall, "Machine learning reduced workload with minimal risk of missing studies: development and evaluation of a randomized controlled trial classifier for Cochrane Reviews", *Journal of Clinical Epidemiology*, vol. 133, pp. 140-151, 2021.

[15] E. Guo, M. Gupta, J. Deng, Y.-J. Park, M. Paget, and C. Naugler, "Automated paper screening for clinical reviews using large language models: Data analysis study," Journal of Medical Internet Research, vol. 26, p. e48996, 2024.

[8] https://partnersplatform.who.int/tools/aria

[9] https://openwebsearch.eu

https://doi.org/10.5281/zenodo.13871740



[16] E. Orel, I. Ciglenecki, A. Thiabaud, A. Temerev, A. Calmy, O. Keiser, A. Merzouki, "An Automated Literature Review Tool (LiteRev) for Streamlining and Accelerating Research Using Natural Language Processing and Machine Learning: Descriptive Performance Evaluation Study", *Journal of Medical Internet Research*, vol. 25, e39736, 2023, JMIR Publications Toronto, Canada.

[17] G. Tsafnat, P. Glasziou, M. K. Choong, A. Dunn, F. Galgani, E. Coiera, "Systematic review automation technologies", *Systematic Reviews*, vol. 3, pp. 1-15, 2014, Springer.

[18] J. H. Schiavo, "PROSPERO: an international register of systematic review protocols", *Medical Reference Services Quarterly*, vol. 38, no. 2, pp. 171-180, 2019, Taylor & Francis.

[19] J. Knafou, Q. Haas, N. Borissov, M. Counotte, N. Low, H. Imeri, A. M. Ipekci, D. Buitrago-Garcia, L. Heron, P. Amini, et al., "Ensemble of deep learning language models to support the creation of living systematic reviews for the COVID-19 literature", *Systematic Reviews*, vol. 12, no. 1, pp. 94, 2023, Springer.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13863828



# SCIENTIFIC QA SYSTEM WITH VERIFIABLE ANSWERS

Adela Ljajić*[1], Miloš Košprdić[1], Bojana Bašaragin[1], Darija Medvecki[1], Lorenzo Cassano[2]
Nikola Milošević[12]
[1]Institute for Artificial Intelligence Research and Development of Serbia, Novi Sad, Serbia
[2]Bayer A.G., Berlin, Germany

*Abstract*

In this paper, we introduce the **Verif.ai project**, a pioneering open-source scientific question-answering system, designed to provide answers that are not only referenced but also automatically vetted and verifiable. The components of the system are (1) an Information Retrieval system combining semantic and lexical search techniques over scientific papers (PubMed), (2) a Retrieval-Augmented Generation (RAG) module using fine-tuned generative model (Mistral 7B) and retrieved articles to generate claims with references to the articles from which it was derived, and (3) a Verification engine, based on a fine-tuned DeBERTa and XLM-RoBERTa models on Natural Language Inference task using SciFACT dataset. The verification engine cross-checks the generated claim and the article from which the claim was derived, verifying whether there may have been any hallucinations in generating the claim. By leveraging the Information Retrieval and RAG modules, Verif.ai excels in generating factual information from a vast array of scientific sources. At the same time, the Verification engine rigorously double-checks this output, ensuring its accuracy and reliability. This dual-stage process plays a crucial role in acquiring and confirming factual information, significantly enhancing the information landscape. Our methodology could significantly enhance scientists' productivity, concurrently fostering trust in applying generative language models within scientific domains, where hallucinations and misinformation are unacceptable.

## INTRODUCTION

The introduction of large language models (LLMs) in recent years has marked a transformative phase across numerous sectors, providing advanced capabilities in understanding, generating, and interacting with natural language [1–6]. Within the scientific realm, these models present an exceptional opportunity to expedite research methodologies, streamline the retrieval of information, and sophisticate the creation of intricate scientific discourse [7, 8]. Nevertheless, as these models become more embedded in scientific endeavors, they encounter a pivotal challenge: the phenomena of hallucinations, or the unintended creation of incorrect or misleading content.

In alignment with findings from prior studies on large language models (LLMs) such as those by [9–11], ChatGPT also encounters the issue of hallucination. The issue of extrinsic hallucination pertains to the creation of unverifiable facts, sourced from its internal memory, across various tasks, without the capability to cross-reference information with external databases. [12].

The issue of hallucinations is particularly critical in scientific settings, where the utmost accuracy and dependability are required. Such occurrences, not only present a barrier to the broader acceptance of LLMs within the scientific community [13], but also instigate a trust deficit, restricting the full utilization of generative language models due to fears of misinformation. To leverage the full spectrum of advantages offered by these models, it is essential to directly confront and mitigate these concerns, safeguarding the integrity of scientific data.

To address this vital issue, we present **Verif.ai**, an innovative open-source project designed to minimize the risk of hallucinations in scientific generative question-answering systems. Our strategy employs a multifaceted approach to information retrieval, utilizing both semantic and lexical search methods across extensive scientific databases like PubMed[1]. This is complemented by a Retrieval-Augmented Generation (RAG) process, employing the fine-tuned generative model, Mistral 7B, to produce answers with directly traceable references. Additionally, our system employs an extra layer of fact-checking, or vetting of generated responses. The verification engine, powered by fine-tuned DeBERTa and RoBERTa models on the SciFACT dataset for the natural language inference task, scrutinizes the congruence of generated claims with their source materials, further solidifying trust in the generated content.

By indicating potential hallucinations and employing advanced hallucination reduction techniques, our system, supported by its open-source framework and the backing of the scientific community, is bridging the trust gap in utilizing LLMs for scientific applications. Through this endeavor, **Verif.ai** underscores the importance of generating accurate, verifiable information, thereby instilling renewed confidence in the use of LLM-based systems for scientific inquiry.

## METHODOLOGY

Our methodology employs a toolbox to discover relevant information and provide context to the question-answering system. Currently, the primary component of this toolbox is the information retrieval engine based on indexed documents from PubMed database. The question-answering system utilizes a fine-tuned LLM to generate answers using retrieved documents in context. A fact-checking or verification engine examines the generated answer within the toolbox, identifying any potential hallucinations in the system. The final

* adela.ljajic@ivi.ac.rs

[1] https://pubmed.ncbi.nlm.nih.gov/

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863828



component of the system is a user interface, enabling users to ask a questions, review answers, and offer feedback functionality, so they can contribute to the improvement of the **Verif.ai** project. The overview of the methodology is depicted in Figure 1. In the following subsections, we provide details of the methods envisioned for each of the components.

### *Toolbox and Information Retrieval*

The major component that has been implemented so far in our toolbox is the information retrieval engine. Our information retrieval engine has two components, lexical search and semantic, or vector-based search. The information retrieval component for lexical search is based on OpenSearch[2], an open-source engine that was forked from Elasticsearch and is under the Apache 2 license. The information retrieval component for the semantic search component is based on the Qdrant vector database [3].

In the compilation of the PubMed corpus from the Medline repository, which encompasses 36,797,469 articles, a deliberate decision was made to exclude articles missing the abstracts. Consequently, the corpus for indexing was refined to 69 % or exactly to 25,488,790 documents. For indexing the content from PubMed articles, we opted to concatenate the title and abstract into a single field, named "text," which serves as the basis for both lexical and semantic searches. To generate embeddings of the "text" field for vector search, the model trained on the MSMARCO dataset ('sentence-transformers/msmarco-distilbert-base-tas-b') was chosen due to its capability to manage asymmetric searches, such as those involving discrepancies in length between queries and the texts being searched [14].

To navigate the tokenization limit of 512 tokens imposed by the selected model for text vectorization, the texts that exceed this threshold underwent a segmentation process. The "text" field is divided into several overlapping segments (with overlap window of 100 tokens) to ensure the preservation of essential information, with each segment being independently indexed to enhance semantic search functionality. This allows for the comprehensive indexing of content that would otherwise be truncated. As a result, the final number of indexed segments is 27,795,286, significantly expanding the scope of searchable content and facilitating more accurate and relevant search results within the indexed corpus.

In the query post-processing, we employ a strategy that integrates the outcomes of both lexical and semantic searches by normalizing their respective retrieval scores to a unified scale range between 0 and 1. This will support direct matches, thereby enhancing the discovery of semantically similar phrases and textual segments where direct text matches are absent. Currently, equal weight is accorded to both lexical and semantic searches. However, recognizing the potential for optimization, we plan to adjust these weights by fine-tuning them during the evaluation phase in the future. This adjustment aims to optimize the balance between lexical and semantic search components, enhancing the overall effectiveness and precision of the retrieval system in identifying the most relevant documents. The selected documents in the information retrieval phase are then conveyed to the RAG component, responsible for generating the appropriate answer.

### *RAG for Question-Answering with References*

The integration of retrieval components with generative models facilitates the generation of text that is both rich in context and also referenced by the sourced articles. This ensures that the generated claims or responses are not only relevant but also verifiable, drawing directly from the content of the articles retrieved. The RAG framework allows the generative model to reference multiple sources, thereby enriching the response with diverse perspectives and insights. Conversely, it also empowers the model to exclude references to any articles whose content is non-essential or irrelevant to the question, underscoring the model's capacity for critical evaluation and selective synthesis of information. To integrate the principles of RAG with our methodology, we tested two novel LLMs - Mistral-7B-Instruct-v0.1, a Mistral 7B parameter model with instruction fine-tuning[4] and Phi-2 model[5]. We fine-tuned both models for the task of question-answering with references using a dataset of 10,000 examples containing randomly selected questions from PubMedQA dataset [15]. The answers in the dataset were generated using GPT-3.5 with the most relevant documents from PubMed passed as context. The following prompt was used to generate answers from GPT-3.5:

> Please carefully read the question and use the provided research papers to support your answers. When making a statement, indicate the corresponding abstract number in square brackets (e.g., [1][2]). Note that some abstracts may appear to be strictly related to the instructions, while others may not be relevant at all.

Fine-tuning was performed using the QLoRA methodology [16]. For training we used a rescaled loss, a rank of 64, an alpha of 16, and a LoRA dropout of 0.1, resulting in 27,262,976 trainable parameters. The input to the training has the following structure: the question, retrieved documents (between one and 10 documents), and the answer.

We then tested the fine-tuned models on the task of answer generation. Using the exactly same input as in the training did not produce the expected results, and therefore, we added the instruction at the beginning of the prompt for both models:

[2] https://opensearch.org/
[3] https://qdrant.tech/
[4] https://huggingface.co/filipealmeida/Mistral-7B-Instruct-v0.1-sharded
[5] https:huggingface.co/microsoft/phi-2

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863828

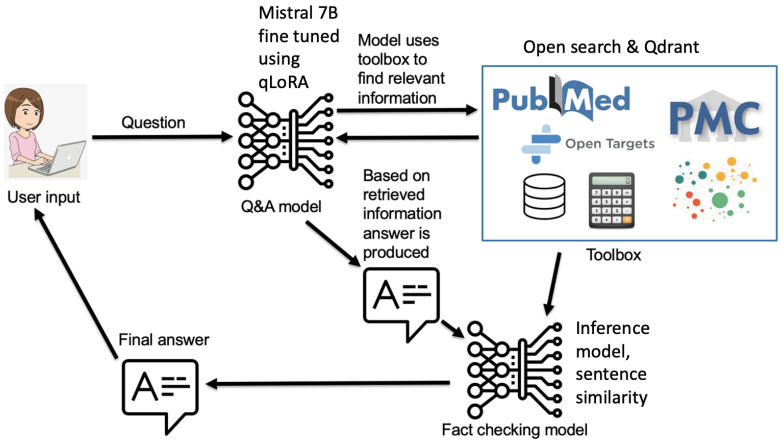



Figure 1: Methodology overview of the **Verif.ai** project

prompt = f"""Respond to the Instruction using only the information provided in the relevant abstracts in ‘‘‘Papers‘‘‘ below.
Instruction: {query_articles}
Answer:"""

The beginning instruction was followed by the question asked by the user and 10 relevant documents obtained by querying OpenSearch (lexical search) and Qdrant (semantic search) to retrieve results ranked by this hybrid combination. The instruction is formatted the same way as in the training set. To prompt both models, we use the mentioned template and default parameters with only two differences: we set max_new_tokens to 1000 and repetition_penalty to 1.1.

We made the preliminary generated QLoRA adapter for Mistral available on Hugging Face[6].

### *Verifying Claims from the Generated Answer*

The aim of the verification engine is to parse sentences and references from the answer generation engine and verify that there are no hallucinations in the answer. Our assumption is that each statement is supported by one or more references. For verification, we compare the XLM-RoBERTa-large model[7] and DeBERTa model[8], treating it as a natural language inference problem. The selected model has a significantly different architecture than the generation model and is fine-tuned using the SciFact dataset [17]. The dataset is additionally cleaned (e.g., claims were deduplicated, and instances with multiple citations in no-evidence examples were split into multiple samples, one for each reference). The input to the model contains the CLS token (class token), the statement, a separator token, and the joined referenced article title and abstract, followed by another separation token. The output of the model falls into one of three classes: "Supports" (the statement is supported by the content of the article), "Contradicts" (the statement contradicts the article) and "No Evidence" (there is no evidence in the article for the given claim).

The fine-tuned model serves as the primary method for flagging contradictions or unsupported claims. However, additional methods for establishing user trust in the system will be implemented, including presenting to the user the sentences from the abstracts that are most similar to the claim.

### *User Feedback Integration*

The envisioned user interface would present the answer to the user's query, referencing documents containing the answer and flagging sentences that contain potential hallucinations. However, users are asked to critically evaluate answers, and they can provide feedback either by changing a class of the natural language inference model or even by modifying generated answers. These modifications are recorded and used in future model fine-tuning, thereby improving the system.

## PRELIMINARY EVALUATION

In this section, we present the results based on our preliminary evaluation. At the time of writing of this article, the project was in the 5rd month of implementation, and we are working on improving our methodology and creating a web application that integrates all the described components.

### *Information Retrieval*

The outcomes derived from employing OpenSearch for lexical search and Qdrant for semantic search were subjected to a qualitative assessment. This examination focused on a subset of indexed articles from the PubMed database, aiming to discern the efficacy and relevance of the search results provided by these distinct methodologies. We compared lexical search, semantic search, and a hybrid combination of both lexical and semantic search. We observed that lexical

[6] https://huggingface.co/BojanaBas/Mistral-7B-Instruct-v0.1-pqa
[7] https://huggingface.co/xlm-roberta-large
[8] microsoft/deberta-v3-large

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863828



search may perform better when the search terms can be exactly matched in the documents, while semantic search works well with paraphrased text or synonymous terms. Hybrid search managed to find documents containing terms that could be exactly matched, as well as ones that were paraphrased or contained synonyms. While semantic search would also find documents that contained an exact match of the terms, it often happened that they were not prioritized. Hybrid search helped in putting such documents at the top of the search results. Based on several user discussions, we have concluded that users expect the top results to be based on exact matches and later to find relevant documents that do not contain the searched terms.

One of the main challenges in creating hybrid search for large datasets, such as PubMed, is storing the index for the semantic part of the engine. While OpenSearch has support for vector search, by integrating the FAISS vector store, it stores all the vectors in the memory. In case the dataset is large (PubMed contains over 120GB of data with 768 dimensions of embedding vectors), and computational resources are limited, it is necessary to find a performant implementation that would store part of the index on a hard disk. Storing part of the index on a hard drive sacrifices to a certain degree performance, but there are implementations, such as by using memory mapped files [18] in Qdrant that have acceptable performance. While it requires us to perform two queries and post-process the results ourselves, it enables the implementation of a large and performant index on limited computational resources.

In our evaluation, we implemented a compression strategy informed by the latest advancements in embedding optimization for vector search, which prioritizes compression over dimensionality reduction for managing large datasets effectively. This approach involved compressing vector embeddings to significantly reduce memory usage 4x by allocating only 1 byte per dimension, thereby retaining 99.99% of the search quality [19]. Utilizing Qdrant's Scalar Quantization feature allowed us to compress the precision of each dimension from a float 32-bit float to 8-bit unsigned integer. As a result, we not only achieved a substantial reduction in storage and memory requirements but also observed an acceleration in the performance of our semantic search operations, effectively overcoming memory and computational constraints.

## *Answer Generation*

We were able to obtain comparable answers from both models (Mistral 7B and Phi2) when prompting them with a smaller number of documents, but the context length of the models played a deciding role when prompting the models with 10 documents. While Mistral-7B-Instruct-v0.1 can process up to 32,000 tokens of input text, the context length obtained by its instruct fine-tuning, the Phi-2 model can only process up to 2,048 tokens. Since our current goal is to generate answers based on no less than 10 abstracts, this automatically left Phi-2 unfit for our needs. Phi-2 was also prone to over-generation, which was not an issue with Mistral. This issue could only artificially be prevented by lowering the max_new_tokens size, but this would also leave the answers unfinished. The combination of these two factors excluded Phi-2 from further testing.

We have manually compared the answers generated by our fine-tuned Mistral-7B-Instruct-v0.1 to answers from GPT-3.5 and GPT-4 on a test set of 50 questions and extracted abstracts. No model showed a clear advantage over the others. The quality, referenced abstracts, and length of the answers varied within each model and among the models. In terms of referenced abstracts, most of the time all three models referenced the same abstracts as relevant. This evaluation indicated that the fine-tuning of the Mistral 7B model improved the model's performance, making the generated answers comparable to those of much larger GPT-3.5 and GPT-4 models for the referenced question-answering task.

We have also tested our model on a preliminary output of the information retrieval module which consisted of user queries and 10 relevant documents along with their PubMed IDs. The model showed decreased performance, which seemed to be related to both a different ID format (realistic PMID as opposed to 0-9 numeration of documents in the train set) and a consistently high number of documents. The model was fine-tuned on a varied number of documents for each query, where the highest number of inputs consisted of four documents so this is an expected behavior. Furthermore, in terms of quantitative analysis, fine-tuning of Mistral-7B-Instruct-v0.1 using the original dataset of PubMedQA questions and GPT-3.5 generated answers showed a continuous decrease of evaluation loss, as can be seen in Figure 2. This behavior leaves space for further improvement so we plan to fine-tune Mistral-7B-Instruct-v0.1 once again using a PMID-like ID format and 10 documents for each query, which could potentially improve the model performance for our purpose. At the time of writing this paper, the dataset is under construction..

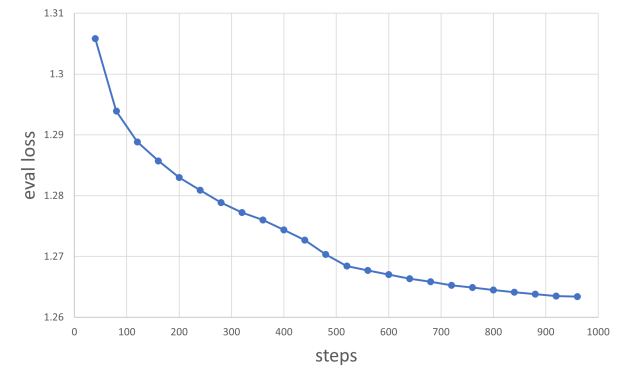



Figure 2: Evaluation loss for fine-tuning of Mistral 7B model on PubMedQA questions with generated and referenced answers

## *Verification and Hallucination Detection*

The evaluation of the fine-tuned XLM-RoBERTa and DeBERTa model on the SciFact dataset that can be used for

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13863828

hallucination detection can be seen in Table 1. The model used 10% of the data for validation and 10% of the dataset for evaluation (test set). All three sets have homogenous distribution of the classes (36%:42%:22% for NO_EVIDENCE, SUPPORT and CONTRADICT classes respectively).

Table 1: The evaluation of the entailment model fine-tuned from XLM-RoBERTa-large and DeBERTa-large model using SciFact dataset

| | XLM-RoBERTa | | |
|---|---|---|---|
| | **Precision** | **Recall** | **F1-score** |
| **NO_EVIDENCE** | 0.91 | 0.96 | 0.95 |
| **SUPPORT** | 0.91 | 0.75 | 0.82 |
| **CONTRADICT** | 0.59 | 0.81 | 0.68 |
| **Weighted Avg** | 0.87 | 0.85 | 0.85 |
| | **DeBERTa** | | |
| **NO_EVIDENCE** | 0.88 | 0.86 | 0.87 |
| **SUPPORT** | 0.87 | 0.92 | 0.90 |
| **CONTRADICT** | 0.88 | 0.81 | 0.85 |
| **Weighted Avg** | 0.88 | 0.88 | 0.88 |

As can be seen from the table, the models exhibited state-of-the-art performance, surpassing the reported scores in [17] for the label prediction task, and DeBERTa-large model showed superior performance compared to the RoBERTa-large. We use fine-tuned DeBERTa-large model for verification and hallucination detection. We also evaluated the SciFact label prediction task using the GPT-4 model, resulting in a precision of 0.81, recall of 0.80, and an F-1 score of 0.79. Therefore, our models outperformed GPT-4 model in zero-shot regime with carefully designed prompt for label prediction for the claims and abstracts in the SciFact dataset. It is important to note that the SciFact dataset contains challenging claim/abstract pairs, demanding a significant amount of reasoning for accurate labeling. Thus, in a real-use case where answers are generated by Mistral or another generative model, the task becomes easier. We believe that this model provides a good starting point for hallucination detection, as supported by our qualitative analysis of several pairs of generated claims and abstracts, which demonstrated good performance.

However, this model has some limitations. While it is capable of reasoning around negations, detecting contradicting claims, differing in just a few words switching the context of the claim compared to the text of the abstract, proves to be a challenge. Additionally, we observe that neither model handles well situations where numerical values in claims are slightly different from the ones in the abstract.

## CONCLUSION

In this paper, we present the current progress on the **Verif.ai** project, an open-source generative search engine with referenced and verifiable answers based on PubMed. We describe our use of OpenSearch and Qdrant to create a hybrid search, an answer generation method based on fine-tuning the Mistral 7B model, and our first hallucination detection and answer verification model based on fine-tuned DeBERTa-large model. However, there are still some challenges to be addressed and work to be done.

LLMs are rapidly developing, and performant, smaller LLMs, with larger context sizes are becoming more available. We aim to follow this development and use the best available open-source model for the task of referenced question-answering. We also aim to release early and collect user feedback. Based on this feedback, we aim to design an active learning method and incorporate user feedback into the iterative training process for both answer generation and answer verification and hallucination detection.

We also aim to improve our answer generation model, by creating better dataset, using GPT4Turbo and manual labelling. We plan to release this dataset in the future.

The model for hallucination detection and answer verification exhibits some limitations when it needs to deal with numerical values or perform complex reasoning and inference on abstracts. We believe that a single model may not be sufficient to verify the abstract well, but it may be the case that a solution based on a mixture of experts may be required [20, 21]. To build user trust, we aim to offer several answer verification methods, some of which should be based on explainable AI and be easy for users to understand. In the future, this may include, for example, verification based on sentence similarity scores.

Currently, the system is designed for use in the biomedical domain and provides answers based on scientific articles indexed in PubMed. However, we believe that the system can be easily extended to other document formats and become a base for a personal, organizational, or corporate generative search engine with trustworthy answers. In the future, our version may incorporate additional sources, contributing to the trust and safety of the next generation internet.

## AVAILABILITY

Code created so far in this project is available on GitHub[9] under AGPLv3 license. Our fine-tuned qLoRA adapter model for referenced question answering based on Mistral 7B [10] is available on HuggingFace[11]. The verification models are available on HuggingFace[12] [13]. More information on the project can be found on the project website: `https://verifai-project.com`.

## ACKNOWLEDGMENT

The **Verif.ai** project is a collaborative effort of Bayer A.G. and the Institute for Artificial Intelligence Research and Development of Serbia, funded within the framework of the NGI Search project under Horizon Europe grant agreement No 101069364.

[9] `https://github.com/nikolamilosevic86/verif.ai`

[10] `https://huggingface.co/filipealmeida/Mistral-7B-Instruct-v0.1-sharded`

[11] `https://huggingface.co/BojanaBas/Mistral-7B-Instruct-v0.1-pqa`

[12] `https://huggingface.co/nikolamilosevic/SCIFACT_xlm_roberta_large`

[13] `https://huggingface.co/MilosKosRad/DeBERTa-v3-large-SciFact`

https://doi.org/10.5281/zenodo.13863828



# REFERENCES


[1] OpenAI. GPT-4 Technical Report; 2023.

[2] Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, Casas Ddl, et al. Mistral 7B. arXiv preprint arXiv:231006825. 2023.

[3] Bubeck S, Chandrasekaran V, Eldan R, Gehrke J, Horvitz E, Kamar E, et al. Sparks of artificial general intelligence: Early experiments with gpt-4. arXiv preprint arXiv:230312712. 2023.

[4] Park JS, O'Brien J, Cai CJ, Morris MR, Liang P, Bernstein MS. Generative agents: Interactive simulacra of human behavior. In: Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology; 2023. p. 1-22.

[5] Touvron H, Martin L, Stone K, Albert P, Almahairi A, Babaei Y, et al. Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:230709288. 2023.

[6] Katz DM, Bommarito MJ, Gao S, Arredondo P. Gpt-4 passes the bar exam. Available at SSRN 4389233. 2023.

[7] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems. 2020;33:9459-74.

[8] AI4Science MR, Quantum MA. The impact of large language models on scientific discovery: a preliminary study using gpt-4. arXiv preprint arXiv:231107361. 2023.

[9] Radford A, Wu J, Child R, Luan D, Amodei D, Sutskever I. Language Models are Unsupervised Multitask Learners; 2019. Available from: https://api.semanticscholar.org/CorpusID:160025533.

[10] Muennighoff N, Wang T, Sutawika L, Roberts A, Biderman S, Scao TL, et al. Crosslingual Generalization through Multitask Finetuning. In: Annual Meeting of the Association for Computational Linguistics; 2023. Available from: https://api.semanticscholar.org/CorpusID:253264914.

[11] Workshop B, :, Le Scao T, Fan A, Akiki C, Pavlick E, et al. BLOOM: A 176B-Parameter Open-Access Multilingual Language Model. arXiv e-prints. 2022 Nov:arXiv:2211.05100.

[12] Bang Y, Cahyawijaya S, Lee N, Dai W, Su D, Wilie B, et al. A Multitask, Multilingual, Multimodal Evaluation of ChatGPT on Reasoning, Hallucination, and Interactivity. In: Park JC, Arase Y, Hu B, Lu W, Wijaya D, Purwarianti A, et al., editors. Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers). Nusa Dua, Bali: Association for Computational Linguistics; 2023. p. 675-718. Available from: https://aclanthology.org/2023.ijcnlp-main.45.

[13] Boyko J, Cohen J, Fox N, Veiga MH, Li JI, Liu J, et al. An Interdisciplinary Outlook on Large Language Models for Scientific Research. arXiv preprint arXiv:231104929. 2023.

[14] Craswell N, Mitra B, Yilmaz E, Campos D, Lin J. Ms marco: Benchmarking ranking models in the large-data regime. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval; 2021. p. 1566-76.

[15] Jin Q, Dhingra B, Liu Z, Cohen WW, Lu X. Pubmedqa: A dataset for biomedical research question answering. arXiv preprint arXiv:190906146. 2019.

[16] Dettmers T, Pagnoni A, Holtzman A, Zettlemoyer L. Qlora: Efficient finetuning of quantized llms. arXiv preprint arXiv:230514314. 2023.

[17] Wadden D, Lin S, Lo K, Wang LL, van Zuylen M, Cohan A, et al. Fact or fiction: Verifying scientific claims. arXiv preprint arXiv:200414974. 2020.

[18] Tevanian A, Rashid RF, Young M, Golub DB, Thompson MR, Bolosky WJ, et al. A UNIX Interface for Shared Memory and Memory Mapped Files Under Mach. In: USENIX Summer; 1987. p. 53-68.

[19] Reimers N. Cohere int8 & binary Embeddings - Scale Your Vector Database to Large Datasets; 2024. [Accessed on 20-03-2024. Available from: https://txt.cohere.com/int8-binary-embeddings/.

[20] Jacobs RA, Jordan MI, Nowlan SJ, Hinton GE. Adaptive mixtures of local experts. Neural computation. 1991;3(1):79-87.

[21] Jiang AQ, Sablayrolles A, Roux A, Mensch A, Savary B, Bamford C, et al. Mixtral of Experts. arXiv preprint arXiv:240104088. 2024.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13863133

# DESIGN SCIENCE RESEARCH FOR THE DEVELOPMENT OF A UNIVERSITY COURSE ON THE INFORMED USE OF SEARCH ENGINES

M. Platz, Saarland University, Saarbrücken, Germany

*Abstract*

From primary school age at the latest, children need the skills to search for information correctly, evaluate search results, and assess the potential risks of disclosing personal data in school and everyday life. In this paper, the question of how primary school teacher trainees can be prepared to incorporate the topic of search engines into their lessons is addressed. Design Science Research is applied, and the course structure and design principles are presented. The course intends to give the students confidence in dealing with and the relevance of the informed use of search engines in primary education to increase the probability that they include the topic in their future teaching. However, no statement can yet be made about the seminar's influence on future teacher behavior.

## THE INFORMED USE OF SEARCH ENGINES

Search engines are essential for participation in (not only digital) life. However, the lack of transparency of algorithms for filtering information, user profiling, and the design of the user interfaces of popular platforms are leading to increasing immaturity in user behavior and a low level of risk awareness regarding privacy. Many of these effects are amplified by AI chatbots.

The KIM-Study 2022 [1], a basic study on media use by 6 to 13-year-olds, showed that 70% of children use the Internet. At the same time, it demonstrates that more and more children are using media independently and without adult supervision [1]. Concerning the miniKIM study 2020 [2], Textor [3] emphasizes that even 2 to 5-year-olds' leisure time is already intensively characterized using media (devices). Even primary school children use search engines like Google [4; 5] without knowing or questioning how they work: "The simplicity and clean interface of Google conceal a complexity that is not understood by users" [6, p. 4]. This applies not only to children and young people but also to adults.

To promote an informed use of search engines, search engine literacy must be promoted among users to promote search and information literacy. Search engine literacy is an aspect of search literacy, which is an aspect of information literacy. Information literacy refers to the ability to understand that information is needed, to search for it effectively and efficiently, to evaluate it appropriately, and to use it. It also includes integrating new information into previous knowledge and utilizing it legally, economically, socially, and ethically to achieve goals [7]. Search literacy refers directly to obtaining information and describes the ability to find and access the desired information to satisfy information needs efficiently and effectively [7]. Developing search engine literacy means gaining knowledge about the basic functioning of search engines (such as ranking, filtering, and sorting algorithms) and the following aspects: findability, linguistic functions and query language [7; 8].

In this context, the overarching research question of the project is: How can a sensible use of search engines be promoted? The following subordinate research questions, among others, will be analyzed:

- (RQ1) How can teaching-learning arrangements be developed to promote the informed use of search engines?
- (RQ2) How can primary school teacher trainees be prepared to incorporate the topic of search engines into their lessons?
- (RQ3) How should information materials be designed to promote an informed approach to search engines?

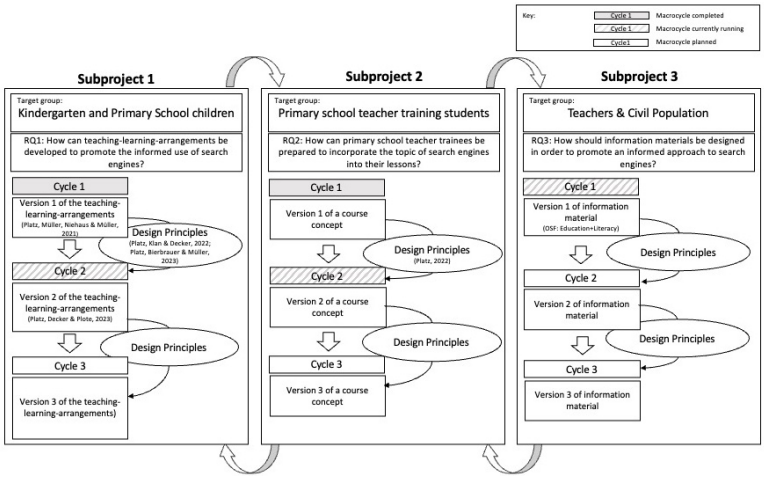



Figure 1: Process, status, and planning of the project.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863133



Fig. 1 shows the process, the status, and the further planning of the project. Design Science Research (see next section) will be used to answer the research questions. The results of the three sub-projects (Fig. 1) influence each other. RQ2 is addressed in this paper. To this end, the current draft of a seminar concept is described, intended to enable student teachers to develop teaching-learning arrangements for the informed use of search engines.

## DESIGN SCIENCE RESEARCH

Design Science Research (DSR) is a paradigm rooted in the philosophy of pragmatism. It has its foundations in the sciences and the construction of the artificial ('The Sciences of the Artifical' [9], first edition 1970). DSR involves problem-solving research to answer research questions about human problems, producing useful artifacts [10], e.g., models, concepts, instantiations, and methods [11].

There are many different DSR models that have been developed and are used in various disciplines. In this paper, models from the fields of 'education' and 'information systems' are used to visualize synergy effects and derive suitable models for the research described.

What the models in the field of 'education' all have in common is the emphasis on cycles or iterations [12, p. 59]. The terminology used to describe the different phases within such cycles varies. What seems to be consistent across design research projects are the following phases (or work areas, cf. [13]) of each so-called macrocycle of design research ([12, p. 59], cf. also [14, p. 35]): (1) preparation and design, (2) implementation and (3) analysis and re-design. These can also be found in the models from the field of 'information systems' (e.g. [15]).

The 'Four Cycle View of Design Science Research' [10, p. 5] is applied in this project (see Fig. 2). It is described below, linked to didactic development research, and guiding questions for the different cycles are formulated based on [10; 13; 16; 17] to support researchers in the implementation of DSR projects.

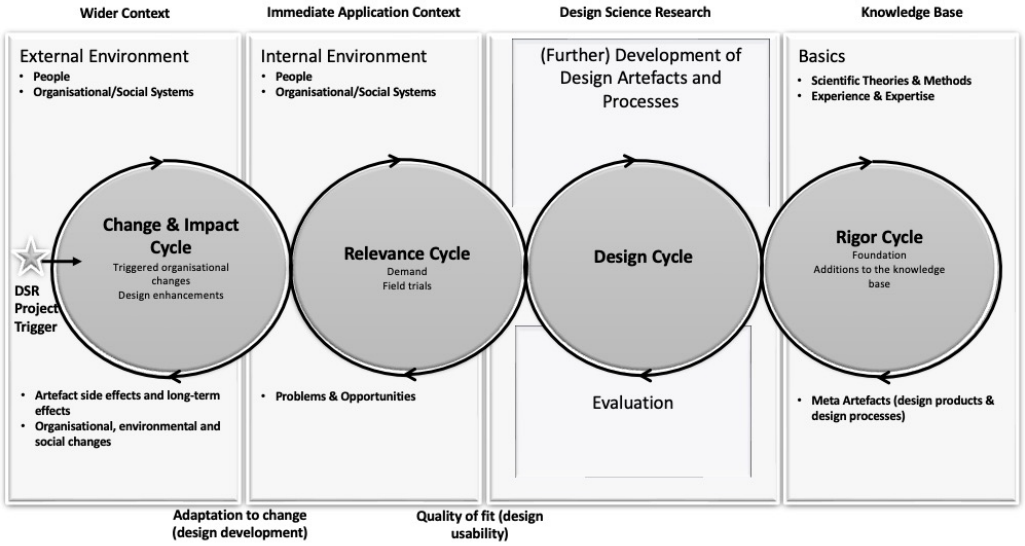



Figure 2: The Four-Cycle View [10] (adapted by the author of this paper).

The Four Cycle View [10] (see Fig. 2) is an extension of the Three Cycle View [16]: "The Relevance Cycle bridges the contextual environment of the research project with the design science activities. The Rigor Cycle connects the design science activities with the knowledge base of scientific foundations, experience, and expertise that informs the research project. The central Design Cycle iterates between the core activities of building and evaluating the design artifacts and processes of the research. I posit that these three cycles must be present and clearly identifiable in a design science research project." [16, p. 88]. The fourth cycle was implemented to better capture the dynamic nature of artifact design for dynamic real-world contexts. It encompasses the impact of design artifacts on their broader organizational and social context [10].

### *Relevance Cycle*

"Good design science research often begins by identifying and representing opportunities and problems in an actual application environment." [16, p. 89]. The Relevance Cycle initiates the DSR with an application context that not only provides the requirements for the research (e.g., the issue/problem to be addressed) as input but also defines acceptance criteria for the final evaluation of the research results. The design research results must be fed back into the environment for investigation and application in the environment. Field trials can be used to determine whether further iterations of the cycle are required in the research project [16].

Table 1: Relevance Cycle - Guiding Questions

| | |
|---|---|
| **1** | **How can the problem space for the DSR project be defined and visualized?** |

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863133

| | |
|---|---|
| 2 | **What does the solution space look like?** (Delimitation of the research project; [17]). The following central questions from the work area specifying and structuring learning objects [13] can help determine the solution space:<br>• What is to be learned first, and with what focus?<br>• What educational content should be emphasized when formulating the topics?<br>• How exactly must the learning object be constructed to enable adequate mediation between subject-specific and individual perspectives?<br>• What is to be learned first, and with what focus? |
| 3 | **Does the design artifact improve the environment, and how can this improvement be measured?**<br>• Through which contexts, views or perspectives can contexts, views or perspectives be used to find links to the learners' previous experiences that promote learning?<br>• In what way might the learning object itself need to be changed or restructured so that it can be learned?<br>• Which sequencing into sub-subjects enables learning paths according to which design principles that are accessible to many learners? [13] |

## *Rigor Cycle*

Design science relies on an extensive knowledge base of scientific theories and methods that form the foundation for rigorous design research. Equally important, the knowledge base contains two types of additional knowledge: The experience and expertise that defines the current state of the field of application of the research and the pre-existing artifacts and processes (or meta-artifacts) found in the field of application. The Rigor Cycle provides the research project with pre-existing knowledge to ensure its innovation. Researchers must thoroughly research and reference the knowledge base to ensure that the designs produced are research contributions and not routine designs based on the application of known processes [16].

Table 2: Rigor Cycle - Guiding Questions

| | |
|---|---|
| 1 | **Are the drafts created research contributions and not just routine drafts?** |
| 2 | **Which extensions of the original theories and methods were made during the research, which new meta-artifacts (design products and processes), and which experiences were gained during the implementation of the research and the testing of the artifact in the application environment?** |
| 3 | **How can the local teaching-learning theory be modified, further differentiated, and increasingly empirically validated based on the results of the design experiments? [13]** |

## *Design Cycle*

"The internal design cycle is the heart of any design science research project." [16, p. 90]. The design cycle moves more quickly between the construction of an artifact, its evaluation, and the subsequent feedback for further design refinement. Simon [9] describes the nature of this cycle as generating design alternatives and evaluating (Rigor Cycle) the other options against the requirements (Relevance Cycle) until a satisfactory design is achieved. During the execution of the design cycle, it is essential to balance the design effort and the evaluation of the evolving design artifact [16].

Table 3: Design Cycle - Guiding Questions

| | |
|---|---|
| 1 | **Which DSR process model fits the research?** |
| 2 | **How can the research processes used in the project be documented and justified?** |
| 3 | **How can the design artifacts be (further) developed?**<br>• Which learning activities should be initiated according to which design principles and tasks for which objectives?<br>• Which teaching and learning materials can be used to support the processes?<br>• How can typical hurdles on the pupils' learning paths be avoided or overcome? [13] |
| 4 | **How can the developed artifact be evaluated?** |

## *Change and Impact Cycle*

The goals of many DSR projects are typically driven by factors in the external environment in which the designed artifacts are to be embedded. From a dynamic perspective, the Change and Impact Cycle allows researchers to become more aware of the dynamics in the broader organizational or societal context and to understand and manage these dynamics within the context of a research project. Today's organizations and societies are constantly changing. Such external dynamics highlight that the aims, rationale, and requirements for DSR projects can change throughout a research project. A particular change in the broader context may also create a problem that did not previously exist (or was perceived) and thus be the trigger for the entire DSR project. The introduction of the artifact into its immediate context of use may, in turn, also lead to changes in the broader environment, resulting in new or changed requirements for the artifact and thus new subsequent iterations through the cycles or, in the worst case, even making the artifact no longer usable [10].

Table 4: Change and Impact Cycle

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935
https://doi.org/10.5281/zenodo.13863133



| | |
|---|---|
| 1 | **What organizational, environmental, and social changes could influence the DSR project?** |
| 2 | **What artifact side effects and long-term effects could arise?** |

In the next section, a university course that is developed with DSR addressing search engines is described.

## THE COURSE

The course 'Computer Science Education in Primary School' is anchored in the curriculum as a mandatory course for primary school teacher students at Saarland University (Germany) since 2021 [18]. It is assigned to mathematics. The learning objectives are to acquire basic knowledge in the field of computer science education in the use of digital media for mathematical teaching and learning processes in primary school and to determine mathematics didactically meaningful applications of digital media, create didactic concepts, and reflect on them critically. The topics addressed in the course are contents of computer science education at the primary level from a mathematics didactics perspective (e.g., algorithms (coding); languages & automata (robots & co.); computer science, people, and society (cryptology)), the design of teaching-learning arrangements with the use of digital media and its use in teaching practice. A unique feature of the course is its practical relevance: The developed teaching-learning arrangement is implemented as part of a teaching experiment in the school internship of the students, and afterwards, they analyze and reflect on the experiment.

Search Engines are chosen as a topic for this course because many areas of computer science are used to ensure the functionality of the search engine, and most of them can be made tangible through CS Unplugged [27] and linked to the competences for computer science education in primary education formulated by the Society for Computer Science Germany [19], e.g.:

- **Algorithms** – How do you make a search engine scalable? And why are search engines so fast? (Search Algorithms, Sorting Algorithms, Graphs, Search Index, etc.)
- **Languages and automata** – How do you communicate with a search engine? (Search Literacy) How to provide a user-friendly GUI? (Human-Computer-Interaction)
- **Information and Data** – Computer security and encryption: How do we protect the user?
- **Computer science, people, and society** – What opportunities and risks does using a search engine have? (Ranking, Filter Bubbles, Privacy, AI)

To develop coherent teaching-learning arrangements to promote the informed use of search engines, several aspects must be considered: According to the Conference of Education Ministers' (KMK) strategy [20], developing and acquiring the necessary skills for life in a digital world goes far beyond basic computer skills and affects all subjects. They can, therefore, not be assigned to an isolated learning area (p. 12). In the strategy, the KMK formulates competencies for the development of which each subject with its specific approaches to the digital world should contribute. Consequently, concepts that can be implemented in regular primary school lessons in combination with traditional teaching topics without taking up additional teaching time must be developed [21]. In the basic curriculum Media education and computer education [22] of the Saarland, which is based on the KMK strategy paper [20], in the competence area 'problem solving and modeling', in addition to developing problem-solving strategies with the help of algorithms, particular emphasis is also placed on reflecting on the influences of algorithms and the impact of the automation of processes in the digital world [22, p. 6]. Teaching concepts for promoting the informed use of search engines can be connected here. The subject-specific and didactic reference links to the traditional content of subject teaching – in this case, maths – the reference to computer science education and the topic of search engines and, finally, reflection to conclude dealing with search engines.

In the first two course cycles (winter semester 2021/22 and summer semester 2022), it was observed that the trainee teachers developed teaching units in which search engines were addressed but not linked to the content of regular lessons, or conversely, that the subject was linked to computer science education, but the search engine context was not addressed. Only the use of the search engine was addressed so that it remains a 'black box' and, e.g., the algorithms for filtering remain invisible [23]. Hevner and vom Brocke [17] note that it cannot be assumed that students have comprehensive knowledge of research processes and methods and that many also lack practical experience with real problems in the workplace. They, therefore, recommend involving students in practical projects where they can apply what they have learned. The seminar aims to develop teaching-learning arrangements to promote the informed use of search engines, involving them in the overarching research project (see Fig. 1). Through practical experience and reflection on their teaching activities as well as acting as DSR researchers, students should acquire specialized, didactic, and practical knowledge and a sense of confidence in dealing with and the relevance of the informed use of search engines in primary education.

Although the iterative nature of design science research is emphasized, this does not mean that design researchers must always repeat macrocycles; so-called microcycles can also be used [12; 24; 25; 26]. Microcycles occur within a design experiment as researchers attempt to adapt both the instructional activities and their underlying theory. Each microcycle consists of an anticipatory thought experiment, the execution of teaching activities, and the analysis that leads to the adaptation or revision of subsequent activities. In practice, design research can vary in how much it builds on either microcycles, macrocycles, or both [24, p. 879]. To guide the students, they go through a DSR microcycle with support and focus mainly on Relevance Cycle – Question 2 (Table 1) and Design Cycle – Questions 3 & 4 (Table 3).

The following design principles were used to re-design the course: If you want to design a course for teacher training students on the development of teaching-learning

https://doi.org/10.5281/zenodo.13863133

arrangements to promote the informed use of search engines, then you are best advised to:

- Include course elements to create Awareness of and concern for the risks of internet searches (Table 5, session 2).
- Give the students an already developed teaching-learning arrangement that they can optimize (e.g., those from CS Unplugged [27]; sessions 3, 4, and 7).
- Provide the (prospective) teachers with a model to support them in lesson planning. One crucial part is the theoretically sound understanding of a concept (originally on the topic of 'algorithms' [28] transferred to 'search engines' and expanded to include a reflection [29] (sessions 5 and 6).
- Let the students become active participants in and designers of the course (sessions 8 to 15).

Table 5 provides an overview of the course structure.

Table 5: Course Structure

| Session | Topic | Mainly addressed cycle |
|---|---|---|
| 1 | Organizational matters & introduction: Computer Science Education in Primary School | Relevance Cycle |
| 2 | Search better and safer on the Internet | Relevance Cycle |
| 3 | CS Unplugged | Rigor Cycle |
| 4 | Search Engines in Mathematics Education | Rigor Cycle |
| 5, 6 | Teaching-Learning-Arrangements in Mathematics in Primary School | Rigor Cycle |
| 7 | Analysis of existing Teaching-Learning-Arrangements to promote the informed use of search engines | Rigor Cycle |
| 8, 9, 10, 11 | Development of teaching-learning arrangements to promote the informed use of search engines as further development of CS Unplugged units | Design Cycle |
| 12, 13, 14 | Testing with the course participants and further development | Design Cycle |
| 15 | Reflection & Conclusion, Term Paper & Wikiversity | Rigor Cycle |

Practical experience and reflection on one's teaching activities, as well as acting as a DSR researcher, are intended to impart specialized didactic and practical knowledge and a sense of security in dealing with search engines at the primary level.

In the course evaluations (as part of the SaLUt II project at Saarland University, part of the quality offensive Teacher training program funded by the BMBF), the students rated the course very positively. This is reflected in the above-average scores on the scales of satisfaction, event quality, lecturers, usefulness, and relevance.

A more detailed evaluation can be done through the reflection the students sent to the docent after their trial in the school intern. However, no statement can yet be made about the seminar's influence on future teacher behavior. For this, the participants would have to be interviewed again much later, when they have completed their studies and are working as teachers.

## CONCLUSIONS

To assess the current state of DSR presented in this paper, guidelines to clarify the requirements for effective design science research and to be able to evaluate DSR projects are taken from the fields of 'information systems' [30] and 'education' [31] (Table 6).

Table 6: Guidelines for the evaluation of DSR projects

| | |
|---|---|
| 1 | **Design as an Artifact/ Consistency (construct validity)/ Practicality – Expected:** The artifact 'concept of the seminar computer science education in primary school' is a method (March & Smith, 1995), i.e., a series of steps to counter the difficulties of integrating the informed use of search engines into the primary classroom. |
| 2 | **Problem Relevance/ Relevance (content validity)/ Practicality – Actual/ Effectiveness – Expected:** The relevance of integrating the informed use of search engines into primary school lessons was described in the first section. Scientific findings were used to design the artifact. The seminar concept can be employed in the primary school teaching degree program, and it is hoped that the students can be given a sense of confidence in dealing with and feel the relevance of dealing with search engines in primary education so that they can integrate it into their future teaching. |
| 3 | **Design Evaluation/ Effectiveness – Actual:** The usefulness, quality, and effectiveness of the artifact were checked, among other things, by the student contributions in the term paper and by teaching evaluations. On this basis, changes were made to the seminar concept. However, whether the seminar impacts the students' future teaching behavior has not yet been investigated. |
| 4 | **Research Contributions:** This paper presents the seminar concept of computer science education in primary school, which other universities can adopt (in whole or in part) (possibly in an adapted form). |

https://doi.org/10.5281/zenodo.13863133



| | |
|---|---|
| 5 | **Research Rigor:** To assess the artifact, the student contributions in the term paper were evaluated, and teaching evaluations were considered. Within the framework of the macrocycle, qualitative evaluation methods will also be used, and a possibility of the actual effects on the students' subsequent teaching behavior is to be developed. (For the construction of the artifact, see guideline 2). |
| 6 | **Design as a Search Process:** The teaching-learning arrangements are developed so that they can be used directly in school. The students publish their developments in a fact sheet format [32] as OER at Wikiversity: https://de.wikiversity.org/wiki/Open-Source4School/Lernumgebungen_zur_Informatischen_Bildung_im_Mathematikunterricht_der_Primarstufe#Search_Engine_Literacy |
| 7 | **Communication of Research:** This paper shares the seminar concept with a scientific audience. Challenges were presented in [23]. |

## REFERENCES


[1] KIM (2022). *Kindheit, Medien, Internet.* Mpfs.

[2] miniKIM (2020). *Kleinkinder und Medien.* Mpfs.

[3] Textor, M.R.: Mediennutzung von Kleinkindern: die miniKIM-Studie 2020. *Das Kita-Handbuch, 2021.*

[4] JIM (2018). *Jugend, Information, Medien.* Mpfs.

[5] Feil, Ch., Gieger, Ch., & Grobbing, A. (2013). *Projekt: Informationsverhalten von Kindern im Internet – eine empirische Studie zur Nutzung von Suchmaschinen.* Deutsches Jugendinstitut.

[6] Le Deuff, O. (2017). Search engine literacy. In *European Conference on Information Literacy* (pp. 359–365). Springer.

[7] Karatassis, I. (2015). A gamification framework for enhancing search literacy. In *FDIA 2015, 6* (pp. 3–6).

[8] Fuhr, N. (2014). *Internet search engines – Lecture script for the course in SS 2014.* http://www.is.inf.uni-due.de/courses/ir_ss14/ISMs_1-7.pdf

[9] Simon, H. A. (1996). *The sciences of the artificial.* MIT Press.

[10] Drechsler, A., & Hevner, A. (2016). A four-cycle model of IS design science research: capturing the dynamic nature of IS artifact design. *DESRIST 2016.*

[11] March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision support systems, 15*(4), 251–266.

[12] Bakker, A. (2018). *Design research in education: A practical guide for early career researchers*. Routledge.

[13] Prediger, S., Link, M., Hinz, R.; Hußmann, S., Thiele, J., Ralle, B. (2012). Lehr-Lernprozesse initiieren und erforschen – Fachdidaktische Entwicklungsforschung im Dortmunder Modell. *MNU 65*(8), 452–457.

[14] Gravemeijer, K., & Prediger, S. (2019). Topic-specific design research: An introduction. In G. Kaiser & N. Presmeg (Eds.), *Compendium for early career researchers in mathematics education* (S. 33–57). Springer.

[15] Peffers, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems, 24*(3), 45–77.

[16] Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian journal of information systems, 19*(2), 87–92.

[17] Hevner, A. R., & vom Brocke, J. (2023). A Proficiency Model for Design Science Research Education. *Journal of Information Systems Education, 34*(3), 264-278.

[18] UdS, (2021). *Modulhandbuch der Studienfächer und der Profilfächer im Studiengang Lehramt für die Primarstufe.*

[19] GI (2019). *Kompetenzen für informatische Bildung im Primarbereich.* Empfehlungen der Gesellschaft für Informatik e.V.

[20] KMK (2016). *Strategie der Kultusministerkonferenz „Bildung in der digitalen Welt*" (Beschluss der Kultusministerkonferenz vom 08.12.2016)

[21] Platz, M., Müller, L., Niehaus, E. & Müller, S. (2021). Modules for Open Search in Mathematics Teaching. In *Proceedings of 3rd OSSYM*, CERN.

[22] MBK (2019). *Basiscurriculum Medienbildung und informatische Bildung.* Ministerium für Bildung und Kultur Saarland.

[23] Platz, M. (2022). Search Engine Literacy – mehr als Kompetenzen zum Recherchieren mit Suchmaschinen. In T. Irion, M. Peschel & D. Schmeinck (Eds.), *Grundschule und Digitalität. Beiträge zur Reform der Grundschule 155* (S. 298–307). Grundschulverband.

[24] Prediger, S., Gravemeijer, K. & Confrey, J. (2015). Design research with a focus on learning processes: An overview on achievements and challenges. *ZDM, 47,* 877–891.

[25] Gravemeijer, K., & Cobb, P. (2006). Design research from a learning design perspective. In J. van den Akker, K. Gravemeijer, S. McKenney, & N. Nieveen (Eds.), *Educational DesignResearch: The design, development and evaluation of programs, processes and products* (S. 17–51). Routledge.

[26] Cobb, P., McClain, K., & Gravemeijer, K. (2003). Learning about statistical covariation. *Cognition and instruction, 21*(1), 1–78.

[27] CS Unplugged (n.d.). *Topics.* https://www.csunplugged.org/en/topics/

[28] Etzold, H., Noack, S. & Jurk, A. (2019). *Algorithmen im Alltag. Leitfaden für Lehrerinnen und Lehrer. Teil 1: Hintergrund und Theorie.* Digitales Lernen Grundschule, Universität Potsdam.

[29] Platz, M., Klan, F. & Decker, A. (2022). Developing and promoting search engine literacy in primary education. In *Proceedings of 4th OSSYM*, CERN.

[30] Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *Management Information Systems Quarterly, 28*(1), 75–105.

[31] Plomp, T. (2013). Educational design research: An introduction. T. Plomp & N. Nieveen (Eds.), *Educational design research* (pp. 11–50). SLO.

[32] Platz, M. (2020). Ein Schema zur kriteriengeleiteten Erstellung und Dokumentation von Lernumgebungen mit Einsatz digitaler Medien. In F. Dilling & F. Pielsticker (Eds.), *Mathematische Lehr-Lernprozesse im Kontext digitaler Medien* (S. 29–56). Springer.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13882749

# FROM FREE SOFTWARE TO OPEN SOURCE: TRAVERSING THE VALUES AND ETHICS OF OPEN SEARCH INFRASTRUCTURES

Renée Ridgway, rridgway@cc.au.dk, Aarhus University, Aarhus, Denmark,

FOSS Research Group, BTECH, Aarhus University, Herning, Denmark

Once was free software valued for its 'peculiar form of potentiality', not per se a thing or technology or license (Kelty 2013) but in the words of Broca, a 'concrete utopia perhaps' (2012), something that had possibility. The original values imbued within free software ranged from 'experimentalism and creativity, provisionality and modifiability, rectification and refraction, dissent and critique, participation and obligation' (Kelty 2013). Moreover, free software embodied a politics of liberalism, with its roots anchored in Unix, situated within the hybrid academic-corporate culture of Bell-Labs. This was cross-fertilized by nascent computer science departments worldwide, which were critical of software development by corporations, organizations or consultancies (Kelty 2008). At that time, the ethics of open software creation was immanent to academic computer science departments and people came *to* free software as the solution, enacting 'disruption in the creation, circulation, distribution and control of knowledge' (Kelty 2013). The power of free software was to make these values, or principles, material and thereby able to be 'manipulated, reconfigured, tested and torqued'; in other words, radically open to change and promoting process over product as an infrastructural and material strategy for a new world (Kelty 2013).

Yet was Free Software a dispositief (Foucault 2009), or an assemblage (Rabinow 2003), a tactic or an open knowledge infrastructure on its own, or as with open source today, a strategy or underlying practice that is not understood enough? The paper delves into the values and ethics of free software to what is now called, since 1998, 'open source', beginning with its negotiations of knowledge and power in regard to intellectual property. Additionally, it investigates the techno-infrastructural reform of 'the potentiality, modifiability, and portability of the tools and code' (Kelty 2013) as well as questions of infrastructure, corporate continuity, competition, maintenance and support (Jackson et al. 2011). It traces how Silicon Valley behemoths Google, Facebook, Microsoft, Amazon, Apple are all built on foundations of Linux servers and open source software, networking tools and programming languages. The evolving genre of FOSS (Free Open Source Software) and FLOSS (Free Libre Open Source Software) has been offered as a solution to the problems of (corporate) infrastructure; simultaneously it opens the door to software developers worldwide, who build new infrastructures and technical solutions based on shared resources and open code.

In order to better understand FOSS, semi-structured ethnographic interviews will be conducted with developers who have been awarded EU funding for NGI (Next Generation Internet) search projects, which have to be made 'open source' upon completion. These will address a specific technology, method, tool or problem related to information retrieval, search, indexing, discovery and exploration of information or incorporating search engine evaluation and ethics in search and discovery. This cultural analysis of the domesticated forms open source is taking will contribute to comprehending the values of the search industry and (open) infrastructures. On the one hand it will demonstrate that open source has become an instrumentalized kind of politics, where the power to 'fork' the software has disappeared into centralized (closed) knowledge infrastructures and how misnomer companies such as OpenAI, 'generates within itself the very tools for its analysis, transformation and reconstruction' (Kelty 2013). On the other, it puts forth the value 'infrastructuring openness' applied to interoperability, integration and the construction of 'recursive publics' within some of these search projects, going beyond the dominance of conservative-libertarian ideologies.



ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13882876



# RETRIEVAL AUGMENTED GENERATION AND SCIENTIFIC KNOWLEDGE GRAPHS TO SUPPORT SCIENTIFIC HYPOTHESES GENERATION

O. Bensch *, T. Hecking †, German Aerospace Center (DLR), Germany
J. N. Kutz ‡, University of Washington, Seattle, USA

*Abstract*

We propose to combine a Large Language Model (LLM) based conversational agent using Retrieval Augmented Generation (RAG) with methods from Literature-Based discovery (LBD) that rely on scientific knowledge graphs to create an LLM based research assistant. A LLM based conversational agent connected to a scientific knowledge graph using RAG can facilitate hypotheses generation by uncovering novel insights and connections across a verity domains, thus aiding exploratory research. Furthermore, such a system could aid the research process by providing an explanation and answering questions about the relation between two research concepts that are not directly related, rather via intermediate concept relations extracted from multiple papers. However, challenges such as ensuring the accuracy of retrieved information and managing the complexity of scientific terminology must be addressed to fully harness RAG's potential in this domain. Our proposed approach promises to accelerate scientific research by supporting both the generation of new research hypotheses and the efficient review of existing knowledge, marking a significant step forward in LBD and therefore automated scientific discovery.

## *Retrieval-Augmented Generation (RAG)*

Large Language Models (LLM) pre-trained on a generation task, also known as generative pre-trained transformer (GPT) [1] models, are gaining in popularity due to the uprising availability of GPT based conversational agents like Gemini [2], GPT-4 [3] or open source alternatives like Mixtral 8x7B [4]. However, GPT models face the major issue of generating incorrect or even false information, when asked about information not available during the training phase [5]. Therefore, RAG combines traditional information retrieval techniques, such as vector search, with GPT models to refine the text generation process [6]. Utilizing RAG, documents are indexed into a vector database via an embedding model for retrieval. This approach fetches relevant information from databases or document collections, using techniques like vector search to find documents that match a user's query [7]. The GPT model then uses these retrieved documents or segments alongside the user's query to generate text that is not only fluent but packed with accurate, relevant and recent information without the need of either a time- and cost-consuming pre-training or fine-tuning phase of the GPT model. RAG utilizes GPTs rather as an interface to access a knowledge databases than to provide this information itself and finds its application in various domains including question answering, content generation, and the enhancement of conversational AI, utilizing a connected corpora [6]. It therefore also addresses the issue of inaccuracies or "hallucinations" in GPT models [5] and provides similarity scores for the relevance of retrieved documents to the query, allowing users to assess the relevance of the generated responses.

* oliver.bensch@dlr.de
† tobias.hecking@dlr.de
‡ kutz@uw.edu

## *Literature-Based Discovery (LBD)*

LBD is a strategic approach aimed at discovering latent connections and insights within scientific texts [8]. It begins with constructing a scientific knowledge graph from literature, identifying key concepts and their relations. For instance, one scientific paper might contain the relation that "fish oil" reduces "blood viscosity," while another contains the relation that high "blood viscosity" can be associated with "Raynaud's disease". This explicit knowledge might lead to the implicit hypothesis that fish oil potentially aids in the cure of the disease. Using this LBD approach this relation was hypothesized by Swanson and later proven to be correct [8]. LBD is currently used in the biomedical domain, due to existing knowledge graphs and ontologies like UMLS [9] enabling the identification of new drug targets and therapeutic approaches with systems like LION-LBD[10]. Recently, efforts have been made to utilize LLMs to construct scientific knowledge graphs in areas such as AI (AIKG) [11] and computer science (CSKG) [12], which might extend LBD applications beyond the biomedical domain. First experiments have also shown that link prediction on non domain specific (open-domain) scientific knowledge graphs might also be used to automatically create novel hyptotheses [13].

Besides the automatic creation of novel hypotheses across research domains, LBD can assist research with open- or closed discovery. Open discovery begins with a search term and explores related concepts to forge new connections and therefore novel hypotheses. This approach does not only consider concepts directly related to the search term, and therefore extracted from scientific texts that mention the search term in combination with another concept, but also concepts extracted from scientific texts that do not mention the search term directly but are related to the search term via a related concept in the knowledge graph [8]. This might help to explore novel, potentially unexpected connections between concepts and aid in hypotheses generation.

https://doi.org/10.5281/zenodo.13882876



On the other hand, closed discovery starts with predefined hypotheses in form of the relation of two concepts. Paths between these two concepts from the scientific knowledge graph are returned to explore the literature across multiple scientific texts. These paths could be seen as either supporting or conflicting indicators for the given hypotheses.

### *Large Language Model (LLM) based Conversational Agent for Hypotheses Generation or Verification*

The development of a conversational agent utilizing LLMs with RAG to support research through hypotheses generation or verification necessitates the integration of methodologies capable of mapping user queries to corresponding concepts within a scientific knowledge graph. Techniques such as keyword search [14], wikification [15], and taxonomy tagging [16] could complement RAG's vector search [7] to align user queries with relevant concepts in a scientific knowledge graph. Another approach cloud be to map the user query to a corresponding concept in the scientific knowledge graph is to unify the embedding of a text with the corresponding graph relation [17]. LLMs can also transform concept relations into narrative text [18]. When combined with user queries in a RAG-like fashion this approach could aid in hypothesis generation in the case of open-discovery LBD or verification in the case of closed-discovery LBD. Such a system cloud provide not only the generated responses but also quantify the similarity between the user's query and the identified relationships, along with the relevant scientific texts underlying these connections. Additionally, language models' reasoning capabilities, illustrated in the Chain of Thought (CoT) method [19], could be employed to verify or confirm relations for hypotheses generation. Ultimately, a LLM based conversational agent that integrates scientific knowledge graphs with RAG could serve as a valuable research assistant that helps facilitating LBD across multiple domains and make a step towards automated scientific discovery.

## REFERENCES


[1] A. Radford and K. Narasimhan, "Improving language understanding by generative pre-training," 2018. https://api.semanticscholar.org/CorpusID:49313245

[2] M. Reid *et al.*, *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context*, 2024.

[3] OpenAI *et al.*, *Gpt-4 technical report*, 2024.

[4] A. Q. Jiang *et al.*, *Mixtral of experts*, 2024.

[5] M. Lee, "A mathematical investigation of hallucination and creativity in gpt models," *Mathematics*, vol. 11, no. 10, 2023. doi:10.3390/math11102320

[6] P. Lewis *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS)*, 2020. https://arxiv.org/abs/2005.11401

[7] V. Karpukhin *et al.*, "Dense passage retrieval for open-domain question answering," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020, pp. 6769–6781. doi:10.18653/v1/2020.emnlp-main.550

[8] D. R. Swanson, "Fish oil, raynaud's syndrome, and undiscovered public knowledge," *Perspectives in biology and medicine*, vol. 30, no. 1, pp. 7–18, 1986.

[9] O. Bodenreider, "The unified medical language system (umls): Integrating biomedical terminology," *Nucleic acids research*, vol. 32, no. suppl_1, pp. D267–D270, 2004.

[10] S. Pyysalo *et al.*, "LION LBD: a literature-based discovery system for cancer biology," *Bioinformatics*, vol. 35, no. 9, pp. 1553–1561, 2018. doi:10.1093/bioinformatics/bty845

[11] D. Dessı, F. Osborne, D. Reforgiato Recupero, D. Buscaldi, E. Motta, and H. Sack, "Ai-kg: An automatically generated knowledge graph of artificial intelligence," in *The Semantic Web – ISWC 2020: 19th International Semantic Web Conference, Athens, Greece, November 2–6, 2020, Proceedings, Part II*, 2020, pp. 127–143. doi:10.1007/978-3-030-62466-8_9

[12] D. Dessí, F. Osborne, D. R. Recupero, D. Buscaldi, and E. Motta, "Scicero: A deep learning and nlp approach for generating scientific knowledge graphs in the computer science domain," *Knowledge-Based Systems*, vol. 258, 2022. doi:10.1016/j.knosys.2022.109945

[13] O. Bensch and T. Hecking, "Towards open domain literature based discovery," in *3rd International Open Search Symposium*, 2021.

[14] G. Salton and M. J. McGill, "Introduction to modern information retrieval," *McGraw-Hill, Inc.*, 1986.

[15] R. Mihalcea and A. Csomai, "Wikify!: Linking documents to encyclopedic knowledge," in *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, 2007, pp. 233–242.

[16] J. Priem, "Linnaeus: Towards a global unified taxonomy for open scholarly data," *arXiv preprint arXiv:2102.09615*, 2011. https://arxiv.org/abs/2102.09615

[17] M. Yasunaga *et al.*, "Deep bidirectional language-knowledge graph pretraining," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 37 309–37 323. https://proceedings.neurips.cc/paper_files/paper/2022/file/f224f056694bcfe465c5d84579785761-Paper-Conference.pdf

[18] Z. Kasner and O. Dusek, "Neural pipeline for zero-shot data-to-text generation," in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022, pp. 3914–3932. doi:10.18653/v1/2022.acl-long.271

[19] J. Wei *et al.*, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 24 824–24 837. https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13882789



# SEARCH, FIND, CITE AND APPLY GRAMMAR RULES FOR TEXTGENERATION

E. Niehaus†, S. Müller, J. Rapp, R. Kastor,
University of Kaiserslautern-Landau (RPTU), Landau, Germany

*Abstract*

To implement an open, transparent and reproducible research culture becomes more challenging with Generative Artificial Intelligence (GenAI) [1]. Like all texts, texts produced by GenAI also have a syntactic and a semantic structure. This paper discusses on a conceptual level the application of grammars and grammar rules as citable units that can be used for transparent text generation for conferences and other application scenarios complementary to standard references, that are used e.g. in scientific publications.

## INTRODUCTION

In scientific works citations and references are used as standard to refer to previous publications and to new scientific results on the existing scientific knowledge. Scientific publications are products and milestones of an evolutionary process and a joined collaborative effort of the scientific community. Publications in proceedings share common requirements for the possible scientific structure of papers. The syntax can be described with a specific grammar. Submissions of papers can be compliant with a specific grammar or may violate some of the rules. GenAI is more and more used and serves as an accelerator for text production.

In computer science a compiler converts a programme written in source programming language A into a functional equivalent code in a programming language B. During a compilation a tokenizer as the first phase is applied to convert the source code in language A as strings of symbols into an array of tokens. The list of tokens is transformed into an Abstract Syntax Tree (AST) [2]. A parser reduces the string of the source code, which is written in language A, to a start symbol S. A specific set of rules of a specific grammar is implemented for this purpose. If that reduction was successful the source code in language A is regarded as syntactically correct. In this context a grammar *(N, T, R, S)* consists of a set of non-terminal symbols *N*, a set of terminal symbols *T*, a system of rules *R* and a start symbol *S*.

A generated AST represents the parsing process of the source code as semantic analysis. In the next step, the generated (and semantically attributed) AST is used to create an output source code in language B. The output language B can be binary/executable or a low-level programming language. So far the

1. parsing of source code in A and
2. generation in code B

can be distinguished conceptually.

In contrast to programming languages, in which the acceptance of source code in language A is boolean, we will apply fuzzy logic for describing the acceptance of natural language elements and their matching with grammar rules [3].

## BASIC EXAMPLE

As a basic example the start symbol will be a general text represented as non-terminal symbol *S*. Rule (1)

$$S \rightarrow SCI|MA \qquad (1)$$

includes an Open Resource (OR) expression, that allows to replace the non-terminal *S* either by *SCI* or by *MA*. The non-terminal symbol *SCI* represents a scientific article and *MA* is a manual e.g. for application of a workflow. This rule is branching into two different allowed text types.

$$SCI \rightarrow S_A S_I S_M S_R S_C \qquad (2)$$

Rule (2) defines the grammar structure of a scientific paper. A scientific paper in this definition of the grammar consists of five sections represented by the non-terminal symbols $S_A, S_I, S_M, S_R$ and $S_C$.

- $S_A$ is the section *Abstract*,
- $S_I$ the section *Introduction*,
- $S_M$ the section *Methodology*,
- $S_R$ the section *Results* and
- $S_C$ represents the section *Conclusion.*

For a high-level programming language designed for computers a missing section or an extra section may lead to a violation of the grammar rules due to the defined set of rules. In natural language we can decompose a

- document *SCI* into sections, e.g. defined by rule (2),
- sections into paragraphs,
- paragraphs into sentences,
- sentences into words and other symbols classified by the lexical analysis into tokens like

† email: niehaus@rptu.de

https://doi.org/10.5281/zenodo.13882789

nouns, verbs, adjectives, mathematical and other symbols, etc.

The non-terminal symbol $S_M$ for the section *Methodology* might be decomposed in other deterministic or optional substructures defined by the rules of the grammar.

## CITATION OF GRAMMAR RULES

As mentioned in the introduction a text can be parsed and tested, if the structure of the document with the section follows a specific rule (e.g. is compliant with the specific rule). A compiler for a programming language needs deterministic true/false assessment for syntactical correctness. In a natural language processing we extend that concept to a probability distribution of OR-expressions in the rules $S \rightarrow SCI|MA$ e.g. stating, that start symbol *S* is generating a scientific article *SCI* with a probability of 0.7 and a manual *MA* with a probability of 0.3 as optional choices for the generation of text.

For citation or reference to a grammar rule $SCI \rightarrow S_A S_I S_M S_R S_C$ probability theory is not the appropriate approach, because in this context the matching is the gradual assignment of truth with a value between 0 and 1 (as full match). Fuzzy logic allows this gradual assignment of a matching result to a cited rule.

## FUZZY LOGIC – MATCHING RULE

Fuzzy Logic extends the classical "true/false (1/0)" logic to a truth value of an expression (e.g. "the man is old") with real number between 0 and 1 (according to Lofti Zadeh [4]). Fuzzy logic can be applied to grammar and languages [5]. Assuming we have a document with a parsed section structure $S_A S_I S_M S_R S_C$ then the document matches fully with the grammar rule mentioned in (2). In a parser a sequence $S_A S_I S_M S_R S_C$ of non-terminal symbols for the different sections can be reduced to the non-terminal symbol *SCI*. In natural language context a source text might not have all the sections and e.g. the section $S_R$ is missing with $S_A S_I S_M S_C$ as section sequence. This leads to a fuzzy match with a value 4/5 = 80% with the grammar rule mentioned in (2). Furthermore conceptionally a clear distinction between "cover" and "exceed" fuzzy values can be considered. Assuming the document inserts an additional section $S_x$ at the end of the document, leading to a section structure $S_A S_I S_M S_R S_C S_x$ , then the fuzzy "cover" is 1 (100%) but the "exceed" value is defined as 1/5 (20%). Due to the fact that documents can have more than one extra section then the exceed fuzzy value can have the maximal value 1 for the exceed value.

The value 0 in "exceed" is a good match, because no extra terminal or non-terminal symbols exist in the text beyond the defined elements of the rule. If the section structure is exceeding more than 100% the given definition of the rule, the "exceed" value is cut to the maximum value of 1. This could be applied, because fuzzy values must be between 0 and 1. In the example rule above, the limitation to 1 would be applied if more than five extra non-terminal symbols appear in the sequence of symbols that is compared with the definition of the rule. It is recommended to apply the fuzzy-NOT to the linguistic value [6] "exceed" to assign good matches to 1 and poor matches to 0. A value of 1 for "not-exceed" means that the document contains e.g. no extra sections (no exceeding terminal or non-terminal symbols) in comparison to the rule mentioned in (2). The fuzzification of the "cover" and "exceed" resp. "non-exceed" fuzzy values might be more complex than counting the symbols matching with definition or do not comply with definition of a specific rule.

The first conceptual approach is done for citations of grammar rules in text generation with gradual matching parameters for "exceed" and "cover" with application of fuzzy logic. In this context matching has to deal with vague information, that is also applied for Semantic Web languages [7]. We may address this issue by either extending current Semantic Web languages to cope with.

## TEXT GENERATION – PROBABILITY

As mentioned above, text generation can be dependent of random experiments in an Abstract Syntax Tree, where the rule allows to have choices. A probability distribution on a finite set of options describes that mathematically. In this context text generation creates decision numbers that are used in the generative process when optional cases in a rule are possible. For the rule $S \rightarrow SCI|MA$ a random number *r* between 0 and 1 with a uniform distribution on the interval [0,1] determines the replacement for a start symbol *S* in the grammar. The non-terminal symbol *SCI* is selected if $r < 0.7$ and a non-terminal symbol for the manual *MA* is selected for replacement otherwise. For the selection process the interval [0,1] is decomposed in *n* subintervals for *n* different options in the grammar rule. To be transparent, in the derivation process from a tree node in AST to child nodes, these random numbers $r_1, r_2, \ldots, r_n$ for *n* different random experiments should be transparently assigned to the rules $R_1, R_2, \ldots, R_n$ of the grammar for which the random experiment is performed.

## DOCUMENT OBJECT IDENTIFIER

A rule as part of a grammar is a digital object, that can be selected for application in text generation. Generalizing the approach described above conceptually, a digital object identifier (DOI) can be used to create a unique and persistent identifier for a rule and/or the corresponding subtree of an AST. Due to the fact that DOI can handle various other digital objects as well [8], the generation of multimedia documents, that include digital objects like audio, video, animation, data, charts, etc. a rule for text generation could be referred to in a consistent way. DOI is standardized by the International Organization for Standardization (ISO), so a reference for grammar rules follows an established work to provide unique identifiers. DOIs are an existing implementation of the Handle System. A DOI for a grammar rule provides the opportunity to fetch the rule and the corresponding subtree, because the DOI operates also consistently within

https://doi.org/10.5281/zenodo.13882789



the Uniform Resource Identifiers (URI). The conceptual work of this paper allows generative models to be transparent, if they are applied not only on the root file of academic, professional, and government documents but also on the decomposition of the documents reflecting generative process in a transparent way.

## LIMITATIONS OF THE CONCEPT

Context-free or context dependent grammars represent the syntactical structure of the document or a language [9]. This is similar to the validation and transparent tracing of document generation and DOI referencing of rules. Similar to the compilation of the source the semantic analysis might require a digital element with DOI reference to have e.g. the logic of an argumentation incorporated in the document generation. As an example we could conclude the statement $C$ if the prerequisites $A_1, A_2, \ldots, A_k$ represent a semantics. $J_1, \ldots, J_i$ are the justification for this conclusion $C$ that are needed for the applicability of the conclusion. In general this is covered by the classical scientific citations, due to the fact that a statement in a scientific article should have references for justifications of the statement.

## CONCLUSION

In this conceptual approach we consider a grammar rule as an identifiable digital object, that can be applied for text generation. Recursive application of rules towards a final generated text creates an Abstract Syntax Tree. The DOI serves as a mechanism to search, find and identify these grammar rules and ASTs in a unique way and supports the text generator with Uniform Resource Identifiers to fetch and apply the grammar rule for a specific generative task. For transparency the DOIs for the grammar rules could be inserted in the document similar to a citation to allow tracing of the generative process and distinguish the generated product from the manual add-ons of the author to the generated text. These manual add-ons to the text can be associated with intellectual contributions to the generated output.

Parts of the generative process are depending on random experiments in decision trees, for which a sequence of random numbers $r_1, r_2, \ldots, r_n$ determines the selection of options $R_1, R_2, \ldots, R_n$ for $n$ different rules with decision options. Providing the sequence of random numbers $r_1, r_2, \ldots, r_n$ for the selection of options in rules $R_1, R_2, \ldots, R_n$ allows to reconstruct the process of text generation, because the previous randomized decision path is replaced by given static numbers, that are recorded during the randomized generation process. The application of rules does not mean that an author who applied generative models needs to accept the result of a generative model completely. The editor may change the generated result and remove or add content elements in comparison to generated content referred to with DOIs. The matching process with a rule requires to represent a partial matching with a DOI referenced grammar rule or AST. Fuzzy values for “cover” and “exceed” can add transparency to the presented conceptual framework by measuring the similarity or compliance with a given DOI referenced rule or subtree.

Finally, by the application of the proposed concept a new scientific article would create not only the classical text content but also a syntax tree with its start grammar rule that other scientists can refer to, search for and apply on their text generation in other publications with a DOI based reference to syntactical structures.

## REFERENCES

[1] B. A. Nosek *et al*., “Promoting an open research culture”, *Science*, vol. 348, no. 6242, pp. 1422-1425, 2015. doi:10.1126/science.aab2374

[2] C. Clark, “ASTs for optimizing compilers”, *ACM SIGPLAN Notices*, vol. 36, no. 9, *pp. 25-30*, Sep. 2001. doi.org/10.1145/609769.609773

[3] G. Satta and O. Stock, "Bidirectional context-free grammar parsing for natural language processing" *Artificial Intelligence*, vol 69, no. 1-2, pp. 123-164, 1994. doi: 10.1016/0004-3702(94)90080-9

[4] L. Zadeh, "Fuzzy logic." *Computer,* vol. 21, no. 4, pp. 83-93, 1988. doi:10.1109/2.53

[5] A. Torrens-Urrutia *et al.* "A fuzzy grammar for evaluating universality and complexity in natural language", *Mathematics*, vol. 10, no. 15, pp. 1-23, Jul. 2022. doi:10.3390/math10152602

[6] C. Nguyen *et al.*, “Hedge algebras, linguistic-value logic and their application to fuzzy reasoning.” *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 7, no. 04, pp. 347-361, 1999. doi:10.1142/S0218488599000301

[7] B. Fernando, and U. Straccia, "Fuzzy ontology representation using OWL 2.", *International journal of approximate reasoning*, vol 52, no. 7, pp. 1073-1094, Nov. 2010. doi:10.48550/arXiv.1009.3391

[8] R. Chandrakar, "Digital object identifier system: an overview.", *The Electronic Library*, vol. 24, no. 4, pp. 445-452, Jul. 2006, doi:10.1108/02640470610689151

[9] R. Simmons and Y. Yu, "The acquisition and use of context-dependent grammars for English", *Computational Linguistics*, vol. 18, no. 4, pp. 391-418, 1992. doi:10.5555/176313.176314

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13872960



# TOWARDS A SYSTEMATIC USE OF WEB-TEXT DATA TO SUPPORT GEOSPATIAL ANALYSIS OF MAJOR NATURAL DISASTER AND CRISIS EVENTS – EVIDENCE FROM THE AHRTAL 2021 FLOODING, GERMANY

V. Rittlinger†, S. M. Farzana[2], X. Hu[3], H. Pandey[3], S. Voigt[1], C. Geiß[1], H. Taubenböck[1]

*Abstract*

During and after natural disaster or crisis events, availability of relevant situation assessments, possibly in near real-time, are essential for many kinds of disaster and crisis relief activities. In this context, web-text data can be a valuable source of information, as they are often available in a timely manner. However, the challenge lies in the extraction and automated aggregation of meaningful insights from this unstructured data. In this talk we present ongoing work regarding an approach to extract geospatial disaster management information from web-crawl data. A special focus is put on the identification of event-related web information, including thematic information on the development and impact of a disaster event as well as the analysis of its geospatial context. As a proof of concept, we reanalyze a flood event that occurred in the Ahrtal valley, Germany in June 2021, resulting in significant damage and human and economic loss.

Crawled Web-text data is used to reconstruct temporal and spatial aspects of this disaster event. A combination of different natural language processing methods is used to filter relevant web-text documents, extract, analyze, and visualize event related information. The processing steps are subdivided into steps focusing on location and thematic-based filtering approaches as well as topic detection methods to classify meaningful information into event related classes. Additionally, a geospatial component is included in the information filtering and generation process and used for geographic characterization of the disaster impact . In this step, we will employ advanced geoparsing approaches that leverage mid-sized large language models like Mistral (7B) and geographic knowledge from OpenStreetMap to accurately extract geospatial information from texts, including fine-grained locations, such as streets, houses, and points of interest.

Apart from the geospatial and flood related information we also attempt to assess the temporal dynamics of the event. In an outlook we describe how web-text derived disaster information may be combined with other geospatial data sources such as satellite imagery to improve situational understanding and assess spatio-temporal dynamics of major natural disaster or crisis events.

† vanessa.rittlinger@dlr.de

[1]*German Aerospace Center (DLR), Earth Observation Center, Oberpfaffenhofen, Germany*

[2]*German Aerospace Center (DLR), Institute for Software Technology, Cologne, Germany*

[3]*German Aerospace Center (DLR), Institute of Data Science, Jena, Germany*

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13882805



# WEB PAGE CLASSIFICATION USING UNSUPERVISED AND SEMI-SUPERVISED CLUSTERING TECHNIQUES

H. Pandey*, T. Elssner†, J. Kersten‡, German Aerospace Center (DLR), Jena, Germany

*Abstract*

In light of ever increasing amounts of data, the efficient and robust search and retrieval of information from different open and closed data sources is a central task. The quality of search can be improved by metadata extraction and enrichment. Web page classification, i.e., assigning one or more thematic classes to web pages, is a central task in this regard. In this abstract, we address the adaptable and content-based classification of web pages in order to support specific and domain-focused search scenarios.

One example would be to find all relevant documents in a set of news pages reporting about flood events induced by heavy rainfall in a specific area. Since we would like to offer such search capabilities for a wide variety of domains and questions/requests, a flexible concept is required.

A common approach is to use different sets of features including links, neighbours, and contents for supervised classification. Semi-structured information embedded within the web pages such as hyperlinks of interconnected web-pages, other modalities such as images, audio, video etc. can be used to enrich and supplement the information presented by the actual content of web pages.

Different taxonomies for web site classification [1] along with ready-to-use software [2][1][3][2][4][3][5][4] already exist. For instance, it is common practice to utilize web page classifiers for focused crawl campaigns [6]. A plethora of existing methods use supervised techniques based on various features of the web pages for classification [1]. However, existing taxonomies tend to represent rather general and static classes, like entertainment, sports, people, travel etc. [7]. Additionally,pre-trained models tend to be insufficient for searching domain-specific content, as an alternative categorization with more fine-grained thematic classes may be desired. This poses a challenge, when the labels of the web pages to be classified are not a priori available. Motivated by this, we focus on unsupervised approaches for web page classification.

Unsupervised web classification methods using URL based features [8], content based features such as Multipurpose Internet Mail Extensions (MIME) content breakdown [9] and external subdomain connection have shown promising results with regards to processing speed, scalability [8] and Quality of Experience (QoE) perceived by the user-groups [9].

In this study, we outline our specific approach to web page classification. The core idea here is to offer flexibility in terms of topics, domains as well as granularity. Based on this, we propose a first concept, which involves unsupervised and semi-supervised clustering, and investigates the usefulness of URL-based, content based and other web page features to achieve general yet highly adaptable categorization of web page categories.

Finally, initial results from experiments using the proposed concept are presented and discussed. The evaluation of experiments will be carried out on data collected using the Owler crawler[5], and lastly, alleys for future work are pointed out.

* hema.pandey@dlr.de
† tobias.elssner@dlr.de
‡ jens.kersten@dlr.de

[1] https://www.whoisxmlapi.com/
[2] https://www.cyren.com/security-center/url-category-check
[3] https://developers.similarweb.com/docs/similarweb-web-traffic-api
[4] https://www.safedns.com/

[5] https://opencode.it4i.eu/openwebsearcheu-public/owler

## REFERENCES

[1] X. Qi and B. D. Davison, "Web page classification: Features and algorithms," *ACM Comput. Surv.*, vol. 41, no. 2, 2009, doi:10.1145/1459352.1459357

[2] H. Shirazi, K. Haefner, and I. Ray, "Fresh-phish: A framework for auto-detection of phishing websites," in *2017 IEEE International Conference on Information Reuse and Integration (IRI)*, 2017, pp. 137–143, doi:10.1109/IRI.2017.40

[3] R. Masri and M. Aldwairi, "Automated malicious advertisement detection using virustotal, urlvoid, and trendmicro," in *2017 8th International Conference on Information and Communication Systems (ICICS)*, 2017, pp. 336–341, doi:10.1109/IACS.2017.7921994

[4] J. Yang, E. Wittern, A. T. Ying, J. Dolby, and L. Tan, "Towards extracting web api specifications from documentation," in *Proceedings of the 15th International Conference on Mining Software Repositories*, 2018, pp. 454–464.

[5] P. Sriram, V. R. Datla, H. H. Gharakheili, and V. Sivaraman, "Enhancing visibility into home networks using sdn," in *2017 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, IEEE, 2017, pp. 1–6.

[6] H. Lu, D. Zhan, L. Zhou, and D. He, "An Improved Focused Crawler: Using Web Page Classification and Link Priority Evaluation," *Mathematical Problems in Engineering*, vol. 2016, pp. 1–10, 2016, doi:10.1155/2016/6406901

[7] D. Shen *et al.*, "Web-page classification through summarization," in *Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, 2004, pp. 242–249.

[8] I. Hernández, C. R. Rivero, D. Ruiz, and R. Corchuelo, "Cala: An unsupervised url-based web page classification system," *Knowledge-Based Systems*, vol. 57, pp. 168–180, 2014, doi:https://doi.org/10.1016/j.knosys.2013.12.019

[9] L. R. Jiménez, "Web page classification based on unsupervised learning using mime type analysis," in *2021 International Conference on COMmunication Systems & NETworkS (COMSNETS)*, 2021, pp. 375–377.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13882966



# AN OPEN SOURCE IMPLEMENTATION OF WEB CLUSTERING ALGORITHMS FOR SELECTIVE SEARCH

Gijs Hendriksen, Djoerd Hiemstra, and Arjen P. de Vries*
Radboud University, The Netherlands

*Abstract*

In distributed search, a document collection is partitioned across several *shards*, which can be queried independently to speed up query processing. Selective search builds upon this infrastructure, but reduces the required resources further by only querying a small number of the index shards. A resource selection algorithm is used to predict which shards are relevant for a given query. To ensure that this works effectively, the shards are usually created using a topic-driven clustering algorithm, so that different documents that are relevant for the same query are more likely to be assigned to the same shard. As a result, the resource selection step should become easier, and only a few shards need to be searched to obtain a high number of relevant results.

An effective clustering algorithm, *size bounded sampling-based K-means* (SB$^2$ *K*-means) [2], uses the lexical similarity of two documents as a proxy for their semantic similarity. A symmetric version of the negative Kullback-Leibler divergence is used to compare the unigram language models of two documents to determine how similar they are to one another. To scale the algorithm to larger document collections like Gov2[1] or ClueWeb09B,[2] the clustering step is performed on a smaller sample (e.g., 1%) of the documents, and the resulting centroids are used to assign the remaining documents to their respective shards. Additionally, in order to limit the number of shards that are much larger or much smaller than the average shard, large shards are repartitioned using the same clustering algorithm, and small shards are merged together. The SB$^2$ *K*-means algorithm has been shown to result in relatively balanced shard maps (in terms of size) that enable effective selective search.

Dai et al. [1] noticed that the topics of content-based partitions do not necessarily align with the intent or information needs of a user. To resolve this mismatch, they use query logs to bias the similarity function towards terms that are more likely to be used by users. They also applied the same bias to the centroids used to initialize the *K*-means algorithm, by clustering the word embeddings of frequently occurring query terms and using those clusters to create an initial set of topics. The authors show that both the new similarity function (`QKLD`) and the new centroid seed selection (`QInit`) improve the performance of a selective search system. Additionally, they published the resulting shard maps for Gov2 and ClueWeb09B, so other researchers could easily use these for their own research on shard maps or selective search.[3]

While the algorithms and shard maps were made publicly available, there is not yet a public implementation of either clustering algorithm. This makes it difficult to replicate their results on alternative datasets, or apply one of the algorithms in a practical setting. In order to make the algorithms usable by the general public, and make it easier for researchers or search engine developers to implement and experiment with selective search systems, we release an open source implementation of SB$^2$ *K*-means, including the extensions `QKLD` and `QInit`. We use `scikit-learn`'s `CountVectorizer` to tokenize and vectorize the input documents.[4] We use GloVe embeddings [3] and `scikit-learn`'s agglomerative clustering implementation to run `QInit`. The *K*-means algorithm is implemented in Cython,[5] in order to make it more efficient and usable on large collections. Our implementation will be published as a Python package on PyPI.[6]

Our ongoing work focuses on replicating the shard maps by Dai et al., and ensuring our implementation produces shard maps of similar quality. We will also generate and publish shard maps for other collections, to verify how well selective search performs for alternate settings and datasets.

## ACKNOWLEDGMENTS

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

## REFERENCES

[1] Zhuyun Dai, Chenyan Xiong, and Jamie Callan. Query-Biased Partitioning for Selective Search. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM '16, pages 1119–1128, New York, NY, USA, October 2016. Association for Computing Machinery.

[2] Anagha Kulkarni. *Efficient and Effective Large-scale Search*. PhD thesis, Carnegie Mellon University, April 2013.

[3] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global Vectors for Word Representation. In Alessandro Moschitti, Bo Pang, and Walter Daelemans, editors, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

* {gijs.hendriksen, djoerd.hiemstra, arjen.devries}@ru.nl

[1] `http://ir.dcs.gla.ac.uk/test_collections/access_to_data.html`

[2] `https://lemurproject.org/clueweb09/`

[3] `https://boston.lti.cs.cmu.edu/appendices/CIKM2016-Dai/`

[4] `https://scikit-learn.org/`

[5] `https://cython.org/`

[6] `https://pypi.org/`

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13882901



# SEARCH REPORTS AS A WAY TO MAKE THE (ACADEMIC) SEARCH PROCESS AND INFORMATION EVALUATION COMPREHENSIBLE AND TRANSPARENT

A. Neovesky, Institute for the History of the German Jews, 20144 Hamburg, Germany

*Abstract*

Bibliography and literature review are key elements and important quality criteria of scholarly publications. However, they do not provide contextual information on the search process that led to the used sources. Yet, the search process is in itself an essential element of academic work. The proposed paper discusses the importance of visibility of the information search process within scholarly work and suggests a search report for the integration of further contextual information.

## WHY LITERATURE REVIEW AND BIBLOGRAPHY IS NOT ENOUGH

Academic publications are unthinkable without references and footnotes, as they prove statements, acknowledge previous work, and enable verification and replication of research. They help to assess the up-to-dateness and depth of a contribution and are thereby also an important criterion to assess the quality of a publication. References also serve as a 'roadmap' for future research, guiding scholars to subsequent relevant literature.

In addition to the references, key publications are presented and discussed in the literature review, and thus also set in relation to each other.

However, these approaches do not provide any information on how the used resources relate to the research process and how they were found. This means, for example, that catalogs or (academic) search engines that were used to find central or particularly important publications cannot be identified.

## THE CONTEXT OF SOURCES AND (RE-) SEARCH PROCESS

### *The importance of the search process within scholarly work*

Especially in a scholarly context, the search process and the selection of the used sources are a central part of the research. Additional information on the choice of sources thereby helps to further strengthen the comprehensibility.

Furthermore, the (re-)search itself is also part of the results of scholarly work. Marchionini considers the search process as such as a result, as the experiences and mental reflections also become part of the seeker's knowledge. [1]

This lack of context of sources, search process and research process could be solved by a search report.

### *The search report*

In addition to the bibliography and the literature review, academic papers could be supported by a search report that provides details on the information search and names the main platforms, catalogs, repositories, and search engines, that have been used. This way, in addition to the sources themselves, the way and access to them would also be highlighted. [2]

Besides greater visibility of the search process and more transparency, this makes it also easier to assess the comprehensiveness of the literature research. Moreover, important repositories for a topic could be emphasized.

The use of AI tools could also be mentioned in a search report. For example, prompts that were used, could be listed, and be thereby presented in a comprehensible and visible way.

Overall, the inclusion of a search report would not only strengthen the integrity and transparency of a publication, but also contribute to advancing research standards. This would also be a more transparent approach to information search within a scholarly context and help to further promote search literacy.

## REFERENCES

[1] G. Marchionini, *Information Seeking in Electronic Environments*, Cambridge, MA, USA: Cambridge University Press, 1995, p. 47f.

[2] A. Neovesky, “Suche und Relevanz in digitalen wissenschaftlichen Sammlungen – Eine Untersuchung zu Suchstrategien, Auswahlverhalten und Digital Literacy von Historiker*innen”, Ph.D. dissertation, Department of Computer Philology and Medieval Studies, TU Darmstadt, 2023, urn:nbn:de:tuda-tuprints-240718, p. 280.

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13882975

# EXPLORING CURATION STRATEGIES FOR AN OPEN WEB INDEX

F. Douglas†, S. Krol, Hochschule der Medien, Stuttgart, Germany

*Abstract*

This project, conducted by a student group from Stuttgart Media University in collaboration with the Open Search Foundation, investigated various aspects of curating an Open Web Index. The primary goal was to explore methods for organizing, maintaining and ensuring the quality of web content.

The project focused on two main areas: content organization and thematic structuring, and the development of a transparent and participatory rating model for web resources. The team reviewed historically developed methods of information organization from information science, as well as existing classification systems and metadata schemes, such as Dewey Decimal Classification, Universal Decimal Classification, Dublin Core, and Schema, to evaluate their applicability for an OWI.

To address scalability, both supervised and unsupervised machine learning techniques were considered as methods for classifying and structuring web content dynamically and efficiently.

The proposed rating model aimed to ensure the quality and trustworthiness of indexed content. It included gamification elements to motivate user participation, such as badges, levels, and quests. The model featured a community-driven approach with user authentication and majority decision mechanisms to prevent misuse and ensure balanced perspectives. Additionally, the model incorporated a detailed rating interface that guided users through evaluating websites based on criteria like content quality, language, usability, trustworthiness, accuracy, and accessibility. Feedback mechanisms and reporting options were included to continuously improve the platform and address any violations or issues.

The project also researched the role of social annotation in enriching metadata and promoting transparency.

This study contributes insights into the curation of an Open Web Index, highlighting the integration of traditional information science techniques with modern, scalable and community-driven methods.

† fd051@hdm-stuttgart.de

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935

https://doi.org/10.5281/zenodo.13883055



# USER-DRIVEN RE-RANKING FOR ADAPTING THE VARIETY IN SEARCH RESULTS

D. F. Auer*, D. S. Naik†, B. König-Ries‡
Institute of Computer Science, Friedrich Schiller University Jena, Jena, Thuringia, Germany

*Abstract*

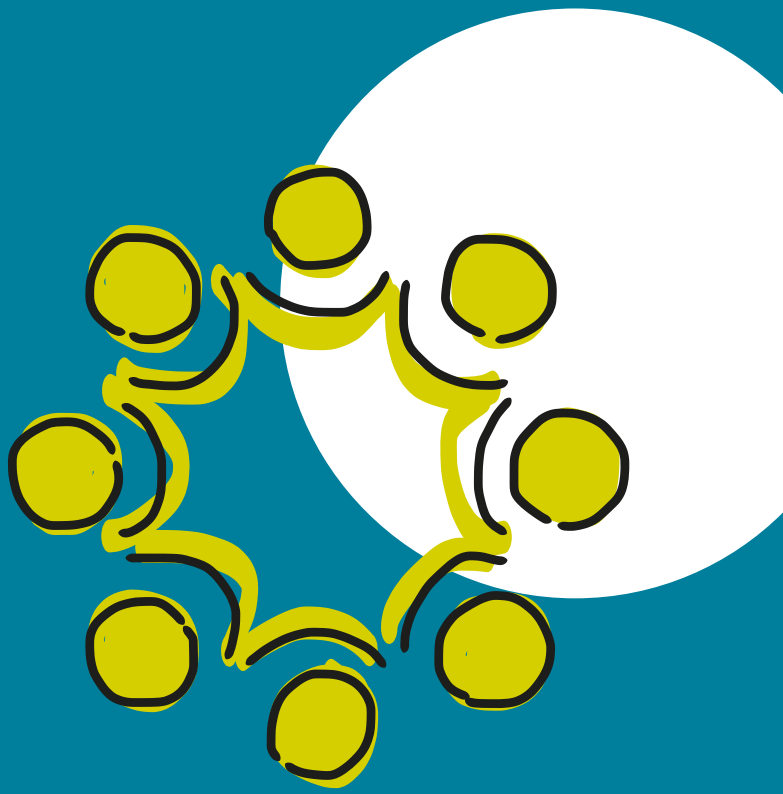
Search systems are a crucial means for users to access information. As only a tiny fraction of the information can be considered in the top search results, this naturally comes with biases that increase even further with personalization, biased databases, or intransparent retrieval systems. We believe that it is essential that users can a) easily understand the characteristics of their search results and b) control them. A prominent example is the presentation of contested political issues, where users may want to understand which view is presented to them most dominantly and change the setting to see more different perspectives. Similarly, when performing a literature search, a user might want to consciously opt for a ranking covering data from different continents or showing research overlaps with other scientific fields. Our work investigates, in the context of literature search, how different demands for variety can be reflected in search results. Therefore, we a) propose re-ranking methods that integrate user-defined variety settings inspired by fairness and diversity metrics and b) present a dataset with variety labels for the broad and multidisciplinary domain of water to test our methods. The experiments show that the proposed approaches effectively adjust rankings to match different variety preferences. With that, we demonstrate the potential of the approach to enhance users' control over search variety, contributing to transparency about entrenched biases and promoting a more user-centered search experience.

* daphne.auer@uni-jena.de
† divyasha.sunil.naik@uni-jena.de
‡ birgitta.koenig-ries@uni-jena.de

# Appendix

## List of Authors

ISBN: 978-92-9083-669-8 Open Search Symposium 2024 - #ossym2024 ISSN: 2957-4935



open search foundation

lrz

ISSN: 2957-4935
ISBN: 978-92-9083-669-8