OpenWebSearch.EU
"Piloting a Cooperative Open Web Search Infrastructure to Support Europe's Digital Sovereignty"

Final Report (M12)
TILDE – Trustworthy Access to Knowledge from the Indexed Web

Version 1.1

The Project is funded by the EC under GA 101070014

Funded by the European Union

Table of Contents

1 Period M1 – M6 **4**

1.1 Module: NLP 4
1.2 Module: Trustworthiness 5
1.3 Module: Visual Web Interface 7

2 Period M6 – M12 **8**

2.1 Module: NLP 8
2.2 Module: Trustworthiness 10
2.3 Module: Visual Web Interface 14

3 Data Availability **21**

4 Table of Figures **22**

5 Bibliography **23**

Preliminaries

i. Project Info

Project number	101070014
Project acronym	ows.eu
Project name	OpenWebSearch.eu – Piloting a Cooperative Open Web Search Infrastructure to Support Europe's Digital Sovereignty
Call	HORIZON-CL4-2021-HUMAN-01
Topic	HORIZON-CL4-2021-HUMAN-01-05
Type of action	HORIZON-RIA
Responsible unit	DG CNECT
Project starting date / Duration	01/09/2022
Project reporting period	2
Project Coordinator	Prof. Dr. Michael Granitzer, University of Passau

ii. Project Partners

Acronym	Partner
KNOW	Know Center Research GmbH

iii. Deliverable Info

Due Date / Delivery Date	08/09/2025
Deliverable Lead	Michael Jantscher
Deliverable type	Report
Dissemination level	SEN
Document Status / Version	V1
Work-package / Lead Partner	NN

iv. Deliverable Summary

This document describes the “2024FSTPC2PN35” TILDE OpenWebSearch.eu project funded by the EC under the GA 101070014 within a Horizon Europe Framework programme.

To facilitate access to the Open Web Index (OWI), we propose an AI-driven Open Web Search (OWS)-based component for data exploration, analysis, and aggregation - prototypically demonstrated by a use case in the health domain. TILDE thereby contributes to increasing the accuracy and trustworthiness of search results and aligns with the overall goal to foster a European ecosystem for web search infrastructure, emphasizing transparency, trustworthiness, and user empowerment.

The respective milestones in this reporting period are:

- Milestone 01 [Infrastructure setup] (M2)
- Milestone 02 [Algorithm selection process finished] (M3)
- Milestone 03 [First version of algorithms and application UI design available] (M6)
- Milestone 04 [Methods to evaluate algorithms available] (M9)
- Milestone 05 [Final set of algorithms available] (M12)
- Milestone 06 [Online demonstrator application available] (M12)

In agreement with the consortium (specifically with Shahab Khormali), a **cost-neutral extension** of the project was confirmed. The project has been extended **until the end of October.** The final demonstrator will be made available at this point.

1 Period M1 – M6

Our visual web-platform comprises three modules: 1.) The NLP module semantically enriches the OWI by extracting health-related concepts (such as symptoms, diseases as well as expressions of well-being level and mood, etc.) and relations connecting them (knowledge graph (KG)). The combination of open-source LLMs together with retrieval-augmented generation (RAG) techniques enables trustworthy interaction with OWI by content summarization and question answering 2.) The Trustworthiness module investigates "unfair" bias to promote fairness and accuracy. Within the component's RAG architecture, the focus is on prompting techniques across LLMs, and on applying benchmarks and metrics to measure bias through retrieval list comparison. This refinement of our prompting strategies will have a positive effect on detecting and reducing (information) inequalities. 3.) The Visual Web Interface module supports browsing evidence along the KG information and delivers fact-based answers to predefined question-patterns (e.g., "Which treatments are available for a disease?", “What are common symptoms of a disease?”, “Which diagnostic methods are used to diagnose a disease?”) using visualization methods and LLM-based summarizations.

1.1 Module: NLP

We analyzed health related content from the OWI by first scraping websites with respect to their *curlie* labels [1] (focusing on the “/en/Health/*” category): from the ~200.000 collected websites, 70.000 contained microdata, a structured data format supported by “Schema.org” that helps retrieving relevant parts of a website. Using the GliNER (Zaratiana, 2023) library, we extracted health-related concepts such as “disease", "symptoms", "medical procedure" and "drugs". An example overview of COVID-19 related, extracted concepts is visualized in Figure 1.

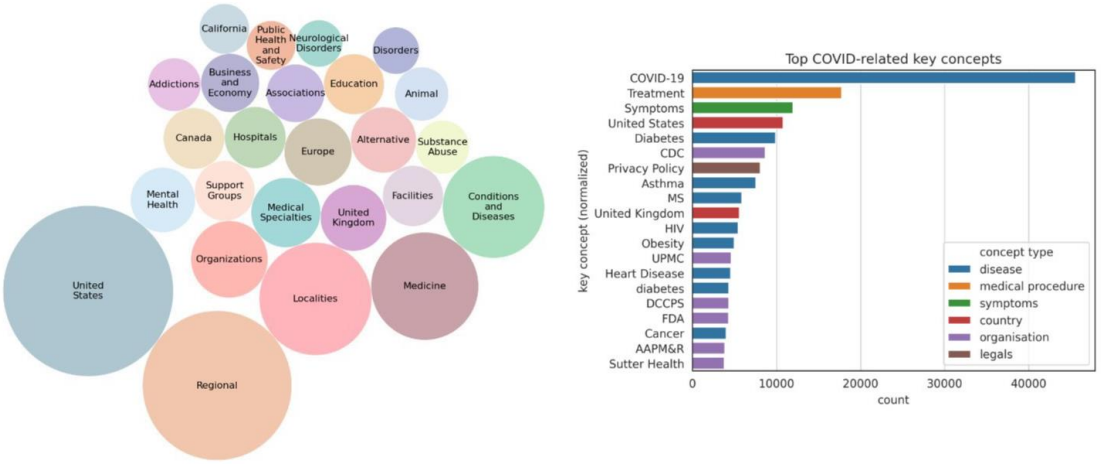

Figure 1 Explorative statistics of COVID-19 related websites from the OWI

In addition, we started to generate a medical knowledge graph, relating websites with each other as well as extracted concepts and structured information from the metadata section of the websites

[1] Curlie labels: https://www.curlie.org/ (Accessed on: 06.11.2025)

(Figure 2). To standardize health-related mentions of websites and to include expert knowledge, we further linked these extracted entities to the UMLS ontology (Medicine, 2025), a comprehensive and precise clinical healthcare terminology system. KG's structure in combination with encoded expert knowledge provides us with a solid basis to minimize the risk of hallucination in the upcoming RAG system.

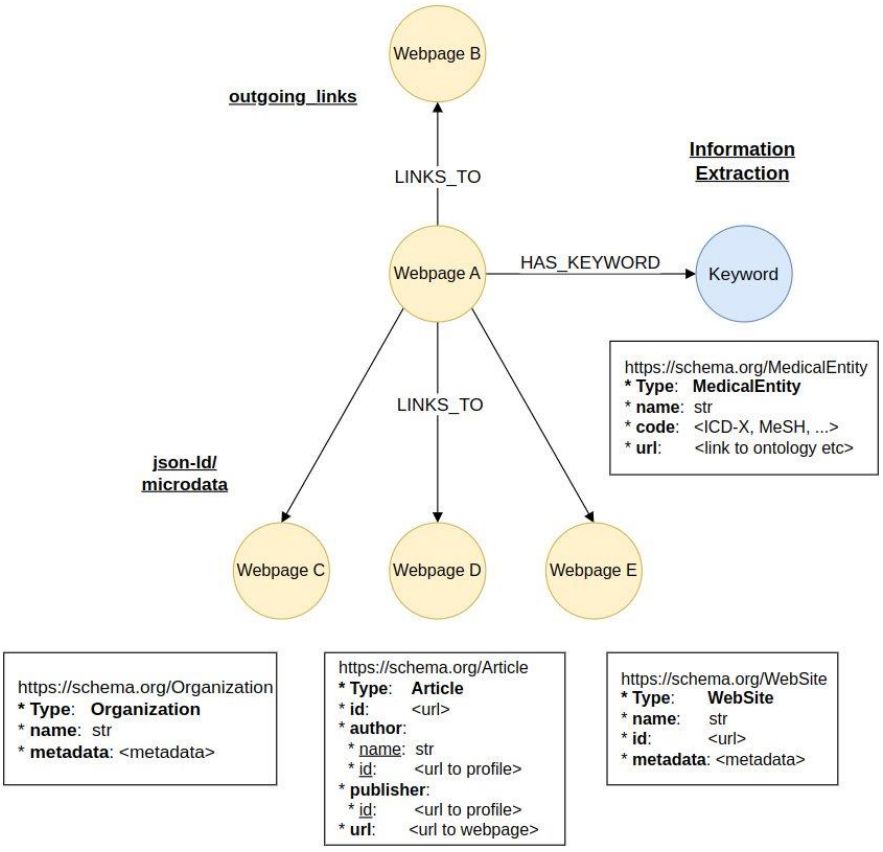

Figure 2 Health Knowledge Graph generated from a websites (structured) metadata and extracted and normalized entities from the website body.

1.2 Module: Trustworthiness

We concentrated on establishing a core set of metrics for evaluating bias through retrieval list comparison. Following the work of (Melchiorre, 2021) we selected key indicators such as document neutrality, normalized fairness of retrieval results (NFaiRR), and ranker-agnostic fairness of document sets (SetFaiRR). This provides the foundational capability to quantify bias within the document lists retrieved for user queries, which is crucial for informing the re-ranking and display mechanisms within our planned RAG architecture which illustration is presented below:

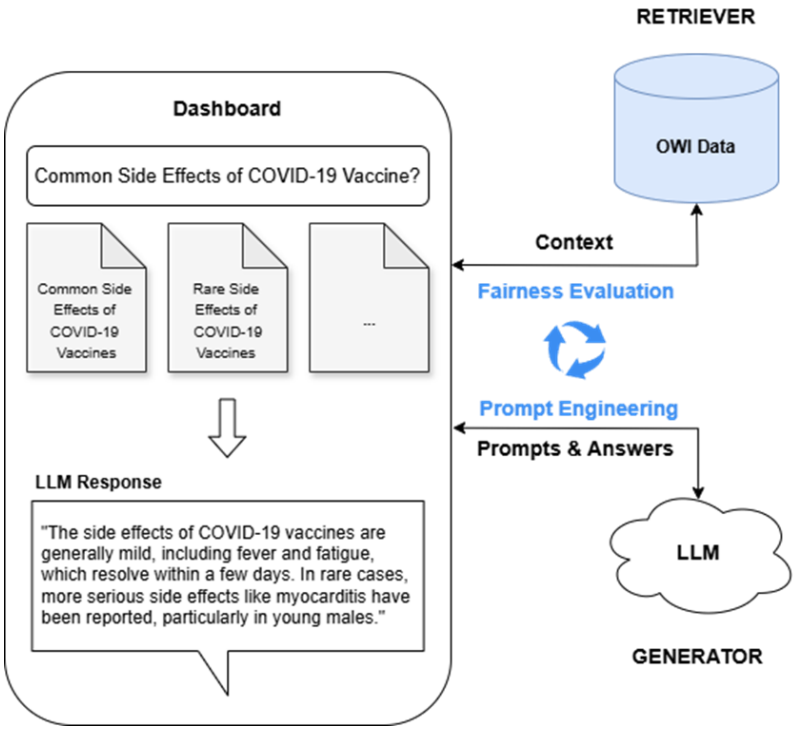

Figure 3 Illustration of RAG architecture and overview of all included components.

When evaluating and re-ranking an initial set of documents retrieved by a RAG architecture for a given user query, our approach centers on integrating document neutrality and the Normalized Fairness of Retrieval Results (NFaiRR) metric. First, we assess the neutrality of each document with respect to predefined protected attributes (e.g., gender), classifying it as neutral if it shows no indication of a protected attribute or if it presents a balanced representation of such attributes. This "document neutrality" score is crucial as it forms the basis for the subsequent fairness calculations. In the re-ranking step, we then compute the NFaiRR for the initial set of documents. This metric allows me to quantify the overall fairness of the retrieved list, considering both the neutrality of individual documents and their position in the ranked list. The NFaiRR score, normalized against an ideal fairness score, provides a clear measure of how well-balanced the results are, guiding the re-ranking process to mitigate biases and ensure a more equitable representation across protected attributes, without significant loss of utility.

During the fairness evaluation process, we specifically utilized the SetFaiRR metric because of its model-agnostic nature. Unlike NFaiRR, which assesses fairness for a *given ranked list* and thus inherently reflects a particular ranking model's output, SetFaiRR quantifies the inherent fairness of a *document set* independent of any specific ranking permutation. This was a deliberate choice. Our primary interest was not just to measure the fairness of a final ranked list, but rather to understand how our re-ranking mechanism—the "model" in a RAG architecture—*influences* the general desired outcome of fairness. By first establishing the baseline fairness of the document set itself using SetFaiRR, we could then more clearly discern the incremental impact and improvements achieved by our re-ranking strategies on the overall fairness of the results presented to the user.

Furthermore, our research has identified a supplementary set of benchmarks such as StereoSet (Nadeem, 2020) and Bias Benchmark for QA (Parrish, 2021). These benchmarks helped us to navigate more precisely towards the desired level of fairness in the final re-ranking step of initially retrieved documents relevant for user query.

1.3 Module: Visual Web Interface

To enable users to explore fact-based, trustworthy data, extracted website facts (concepts and relations) can be used to filter the data to only provide relevant information. Furthermore, concepts in the data can be highlighted, enabling users to gain evidence on their questions and easily identify relevant content (Figure 4). As access to OWI was delayed, other data sources were used for this mock-up as well as for the UI design in Figure 5.

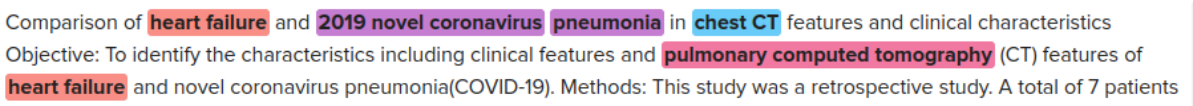

Figure 4: Mock-up highlighting facts extracted from a document.

Figure 5 illustrates a first version of our application UI design providing an 'aggregated' and 'visualized' answer to the question: "Which diseases are related to the symptom respiratory failure?". Users are provided with a summary of 44 results, highlighted facts extracted from the search results as well as a heat map to visually explore the relation strength of instances from the two categories "Diseases" and "Symptoms".

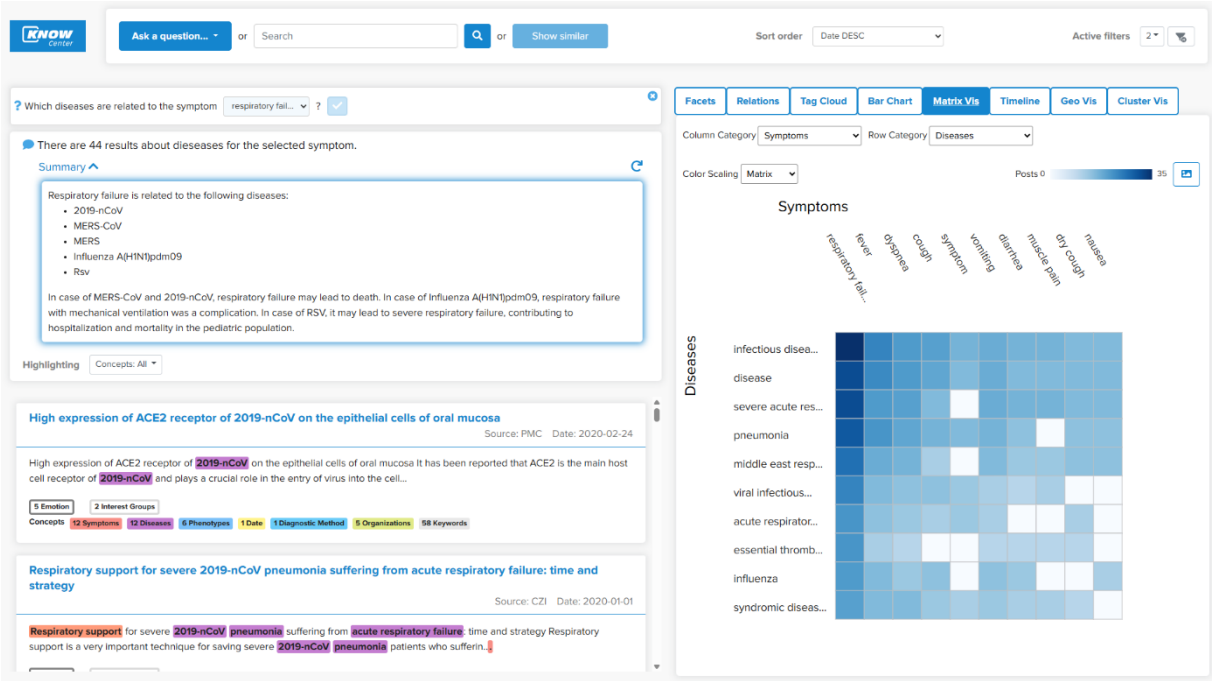

Figure 5: Application UI-Design showing highlighted text passages, extracted concepts as well as an aggregated summary to a sample question on the left side and a heat map visualization between instances of the two concepts "Diseases" and "Symptoms" on the right side.

2 Period M6 – M12

2.1 Module: NLP

Building on the work completed in the previous project phase, this module focuses on the development of a vertical, hybrid Retrieval Augmented Generation (RAG) search engine tailored for the healthcare domain. The goal of this module was to design and implement a robust information retrieval pipeline that leverages both structured entity-level search and dense semantic retrieval. Figure 6 visualizes the whole technology stack of this system.

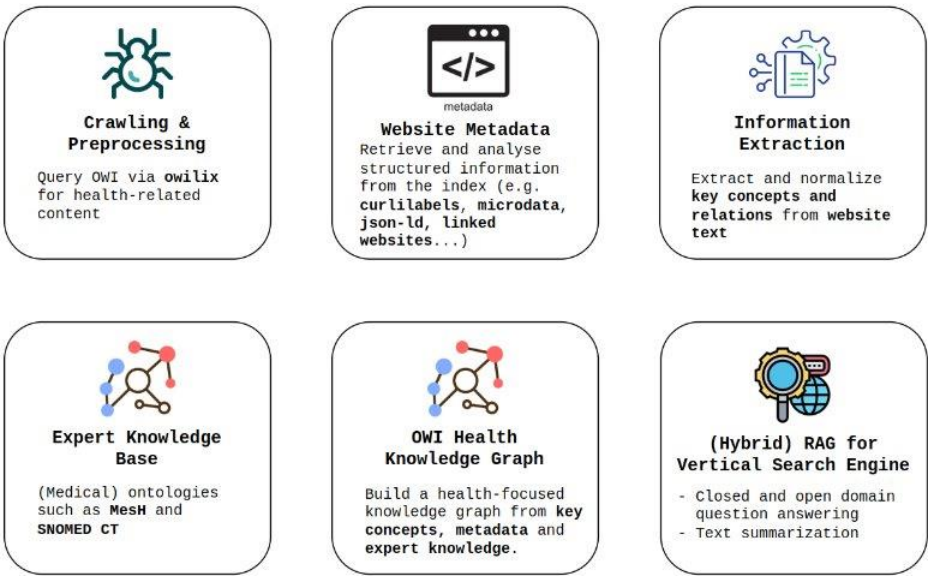

Figure 6 Technology stack for a vertical, hybrid search engine in the healthcare domain.

Data Indexing and Preprocessing

The scrapped health-related websites from the previous period are indexed into an Apache Solr database (Solr, 2025). During indexing, the following preprocessing steps were applied:

- **Entity extraction and normalization.** The extracted and normalized entities from the websites create a structured layer of metadata to support entity-based retrieval.
- **Chunking.** Website content was chunked into overlapping chunks of 500 tokens, facilitating more granular document representation and retrieval.
- **Embeddings.** The title of each webpage and each content chunk were embedded using a Sentence Transformer model, enabling semantic similarity search. For embeddings, the sentence transformer model *all-MiniLM-L6-v2* (Reimers, 2024) is utilized.

These steps then build the backbone of the hybrid RAG system.

Retrieval Architecture

Using the LangChain (Chase, 2025) framework, two primary retrievers were implemented:

- **Entity-Based Retriever.** First, user queries are analyzed with the GLiNER (Zaratiana, 2023) model to identify key entities, which are then normalized to corresponding concepts in the UMLS ontology. Using these standardized entities, the system performs a high-precision, entity-level search in Apache Solr (Solr, 2025), prioritizing documents with strong entity matches for improved retrieval accuracy.
- **Dense Retriever.** The user query is embedded using the same Sentence Transformer model (Reimers, 2024) as utilized for the indexed documents/websites. Cosine similarity is then used to retrieve the top-K relevant documents based on similarity between query embeddings and both title embeddings and content chunk embeddings.

Hybrid Fusion

To combine the strengths of both retrievers, we employ Weighted Reciprocal Rank Fusion (RRF) (Cormack, 2009). This approach merges the ranked lists by both retrievers. This means, documents that perform well in both retrieval paradigms are promoted, ensuring a balance between precision and semantic recall.
The final ranked list of websites is presented to the user as search results.

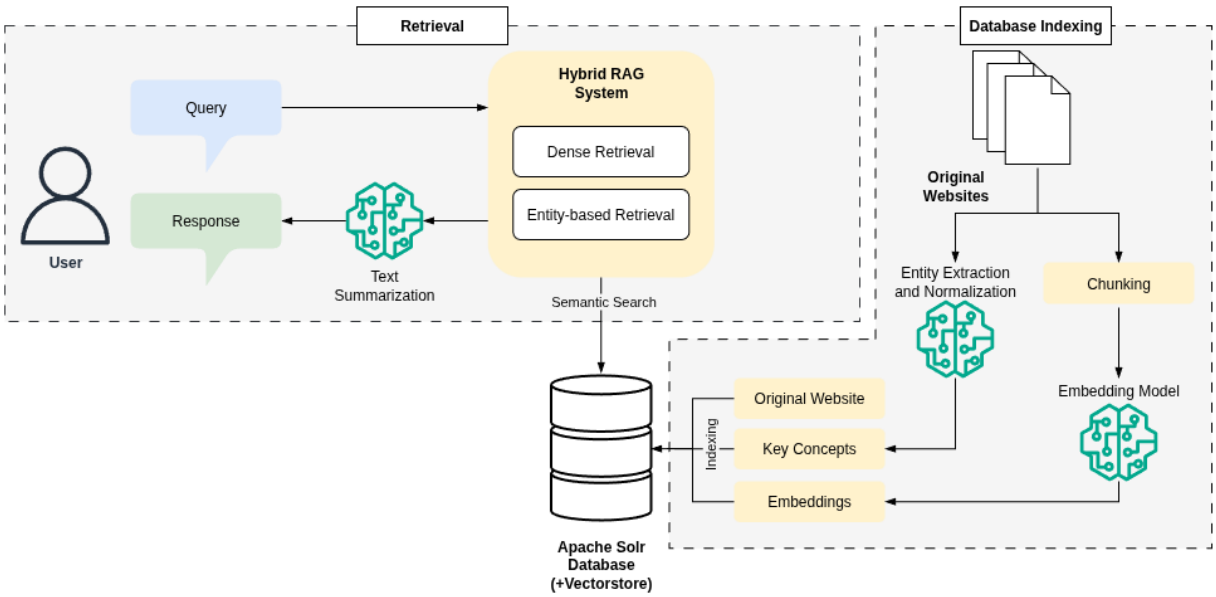

Figure 7 Hybrid RAG Architecture with (i) the database indexing phase and (ii) the retrieval step

Text Summarization

Once the top-ranked documents are identified, the system applies a query-focused text summarization step to extract and condense the most relevant information. Using a Langchain summarization pipeline, the retrieved content is summarized into a concise, user-friendly response

that highlights the key facts aligned with the query intent. The utilized model for summarization is *gpt-4o mini* (OpenAI, 2025).

Integration with Module: Trustworthiness and Visual Web Interface

The output of this module, a ranked list of retrieved websites, serves as the foundation for the trustworthiness module. In this upcoming module, these results will be re-ranked using additional trustworthiness and transparency scores, further improving the reliability of search outputs for users. This RAG system also forms the core of the interactive search module. Nevertheless, the output of this module can be used as a standalone service.

Deployment and Open Access

To ensure reproducibility and ease of integration, the entire RAG-based search system has been fully containerized using Docker. All components, including the Apache Solr database and retrieval services are encapsulated within separate containers. At the end of this project, these dockers will be openly released to facilitate adoption and further development.

2.2 Module: Trustworthiness

Modern search systems, especially those leveraging Large Language Models (LLMs), risk amplifying societal biases present in their vast training data. A simple query can yield results that are geographically skewed, lack viewpoint diversity, or reinforce harmful stereotypes. This project addresses this challenge by designing and implementing a robust, research-informed pipeline to detect and mitigate such information inequalities in real-time.

Our primary goal was to move beyond simple relevance ranking and create a system that intelligently balances search relevance with principles of fairness, credibility, and representational diversity. To achieve this, we developed a sophisticated three-stage re-ranking pipeline built upon the programmatic LLM framework called DSPy. Our methodology is directly informed by seminal academic research, operationalizing key principles from publications on bias measurement, including (Melchiorre, 2021), **BBQ** (Parrish, 2021), and **Stereoset** (Nadeem, 2020). In the continuation, DSPy framework with its components will be described in detail together with the research-driven logic behind its design, using the topic of "COVID-19 treatment options" as a running example.

Core Framework: DSPy for Programmatic Prompting

The foundation of our system is DSPy (Khattab, 2022), a framework from Stanford NLP that treats LLM pipelines not as manually tuned prompts but as programs that can be systematically optimized. This approach was chosen for its modularity, explainability, and ability to improve performance based on data and metrics. Key DSPy elements used in this project include:

- **Signatures:** These are declarative specifications that define the input and output fields of a task. They abstract the "prompt engineering" into a formal contract, making the LLM's role clear and predictable. For example, a simple signature might be question -> answer.

- **Modules:** These are the building blocks of a DSPy program, composing signatures into executable logic. We heavily utilize the *dspy.ChainOfThought* module, which instructs the LLM to generate a step-by-step reasoning process before providing its final answer, making its decisions transparent.

- **Optimizers (Teleprompters):** This is DSPy's most powerful feature. An optimizer, like *BootstrapFewShot*, can automatically find the best prompt for a given task. It works by running the program on a small set of training examples and evaluating the output against a custom metric. It then generates and refines few-shot examples and instructions to create a new prompt that maximizes the metric's score.

This programmatic approach allowed us to construct a complex, three-stage pipeline where each step is a well-defined, potentially optimizable module. Our core innovation is a pipeline that processes search results in three distinct stages: Enrichment, Re-ranking, and Auditing.

Stage 1: Parallel Document Enrichment

Before any re-ranking occurs, we first create a rich, structured representation of each candidate document. This is an efficient, parallelized "map" step where each of the top-N documents (e.g., top 50) are analyzed to extract a suite of fairness-related attributes.

The following attributes are extracted for each document:

1. Semantic Attributes:
    - *Viewpoint*
        - Classifies the document's perspective to enable viewpoint diversity (e.g., "Official Health Guidance," "Patient Experience Narrative")
    - *Mentioned Groups*
        - Identifies any specific demographic groups discussed, helping to surface marginalized voices
2. Trustworthiness and Clarity:
    - *Trustworthiness*
        - Classifies the source into predefined tiers (e.g., "Tier 1: High Authority" for who.int, "Tier 4: User-Generated" for a personal blog)
    - *Context Clarity*
        - Inspired by the BBQ paper, this attribute classifies the content as either "Factual/Disambiguated" or "Anecdotal/Ambiguous"

- This is crucial for distinguishing between evidence-based articles and opinion pieces, allowing the re-ranker to prioritize verifiable information

3. Neutrality Scores:
    - *Explicit gender neutrality score from* (Melchiorre, 2021)
        - We implemented an LLM-based module to count explicit gendered keywords (e.g., "he" vs. "she") and calculate a score from 0 (biased) to 1 (neutral)

Stage 2: Intelligent Re-ranking

Once the enrichment stage is complete, the pipeline transitions to its core decision-making phase: intelligent re-ranking. The list of enriched document objects, now containing a rich set of attributes for each candidate, is formatted into a single, detailed prompt. This prompt is then passed to our primary re-ranking module, which is powered by the *EnhancedFairnessReRankingSignature* and a Chain-of-Thought (CoT) process. This signature is not a simple instruction, but a detailed analytical rubric that guides the LLM through a multi-factor decision-making process, governed by a strict hierarchy of principles.

The LLM is instructed to use the enriched data for each document to perform a series of trade-offs, following these principles in order of importance:

1. **Maximize Fairness (The NFaiRR Principle):** Our critical innovation is the operationalization of the NFaiRR metric (Melchiorre, 2021) as a direct, actionable instruction. The primary instruction, given top priority, is:

   "1. Maximize Fairness (The NFaiRR Principle): Your primary objective is to create a ranked list that places documents with the highest neutrality_score at the top positions."

   In this step, the LLM scans the neutrality_score for every candidate document and identifies those with the highest scores as the primary contenders for the top ranks.

2. **Maintain Credibility & Clarity:** After identifying the fairest documents, the LLM is instructed to use the enriched data as a credibility filter. It analyzes the *trustworthiness* and *context_clarity* attributes for the top contenders. This step is crucial for preventing the system from promoting unfair but non-credible content. For example, the LLM will penalize or de-rank a document with a high neutrality score if it comes from a "Tier 4: User-Generated/Commercial" source or is tagged as "Anecdotal/Ambiguous," unless it provides a viewpoint that is critically important and unavailable in a higher-quality source.

3. **Ensure Viewpoint Diversity:** Finally, with a pool of candidates that are both fair and credible, the LLM uses the *viewpoint* attribute as a final balancing factor. If multiple documents have similarly high neutrality and credibility scores, the LLM is instructed to select a variety that covers different perspectives (e.g., "Official Health Guidance," "Patient Experience Narrative," "Investigative Journalism") to ensure the final list is not only fair and credible but also intellectually comprehensive.

To ensure these complex trade-offs are made reliably, the signature employs a structured Chain-of-Thought (CoT) process via a dedicated *ranking_reasoning* field. This forces the LLM to externalize its analytical process, effectively "showing its work" by first analyzing neutrality, then applying the credibility filter, and finally constructing the ranking based on this multi-layered analysis.

The final outputs of this stage are twofold: a machine-readable, structured JSON object containing the *re_ranked_list* for the dashboard, and a human-readable *justification* that explains the reasoning behind the new order. In essence, Stage 2 transforms the abstract goal of "fairness" into a concrete, data-driven task. The LLM is no longer making a subjective judgment; it is executing a defined procedure, using the rich metadata from Stage 1 to navigate the complex trade-offs between fairness, credibility, and diversity, and producing a fully justified and transparent outcome.

Stage 3: Two-Point Stereotype Auditing

Inspired by the Context Association Test in the Stereoset paper, we implemented a final auditing stage to act as a safety net. This stage checks for harmful stereotypes in the outputs of our own system. Recognizing that bias can be introduced at multiple points, we designed a two-point audit:

1. **Process Audit:** The re-ranker's justification text is audited. This checks the fairness of our system's internal reasoning, preventing it from introducing new biases (e.g., reasoning that a source is less credible due to its national origin).
2. **Output Audit:** The final, synthesized answer that will be shown to the user is audited. This is the last line of defense, catching cases where the answer-generation LLM might make a harmful stereotypical leap from fairly ranked but nuanced sources.

If an audit fails, the system can flag the content, display a warning, or trigger a re-generation attempt, ensuring a higher level of safety and responsibility.

To conclude, this project module successfully demonstrates the design of a comprehensive, multi-stage pipeline for enhancing fairness in information retrieval. By leveraging the programmatic power of DSPy and operationalizing key principles from foundational research papers, we have constructed a system that is not only effective but also transparent, auditable, and robust. The three-stage process of Enrichment, Re-ranking, and Auditing ensures that decisions are made based on rich, multi-faceted data and that the system's own reasoning is held to a high standard of fairness. This framework serves as a powerful blueprint for developing next-generation search systems that are committed to providing equitable access to information.

2.3 Module: Visual Web Interface

Figure 8 shows the visual web interface to retrieve and analyze health-related information. The upper area enables the user to provide a query to search for relevant information. Currently, the results are retrieved based on a lexical search. The integration with the hybrid approach in WP1 is still work-in-progress.

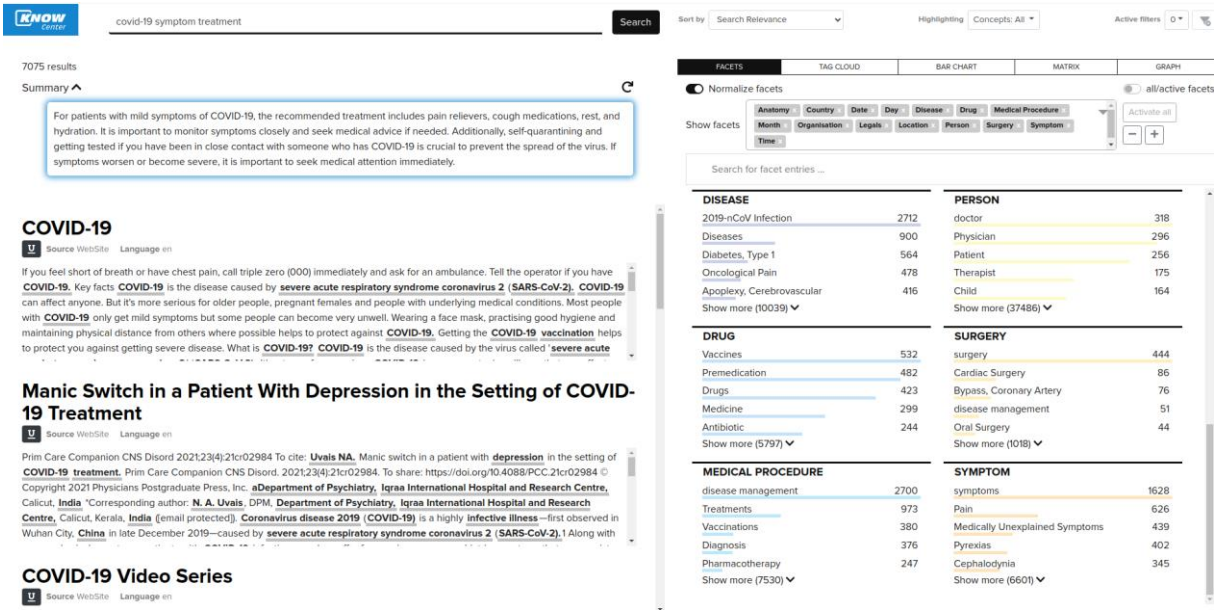

Figure 8: Visual Web Interface for retrieving and analyzing health-related Websites.

Question Answer

Below the query input field, a user can optionally activate the answer generation by clicking on “Summary”. An example for the query “covid-19 symptom treatment” is shown in Figure 9.

For answer generation the integration with WP1 has already been completed by using the results of the hybrid search and the corresponding component described there.

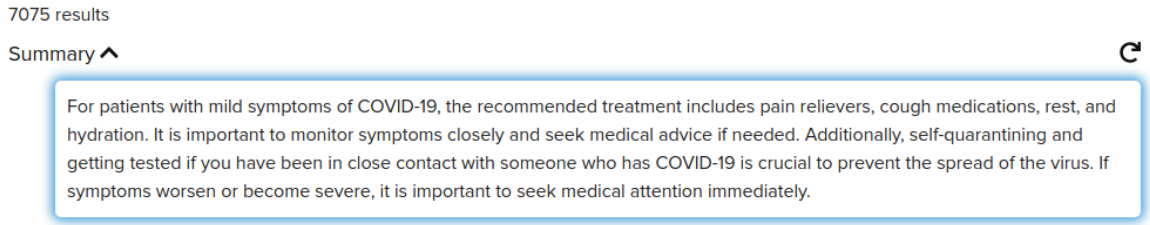

Figure 9 Answer generated for the query "covid-19 symptom treatment".

Search Result List

The search results are displayed below the answer in the lower left area. Users can either investigate the website content (Content View, see Figure 10) where extracted concepts are highlighted or show the list of extracted concepts (Concept View, see Figure 11). The icon below the title enables users to easily switch between the two views. Users can narrow down the result set by clicking on one of the extracted concepts in the “Concept View”.

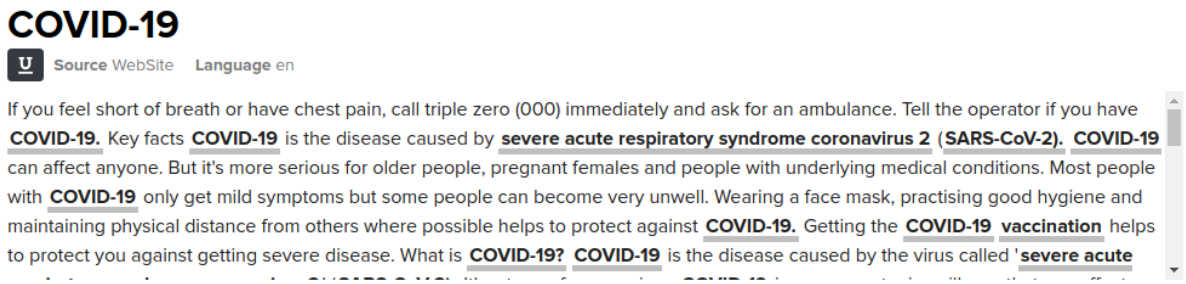

Figure 10 “Content View” showing Website content by highlighting extracted concepts.

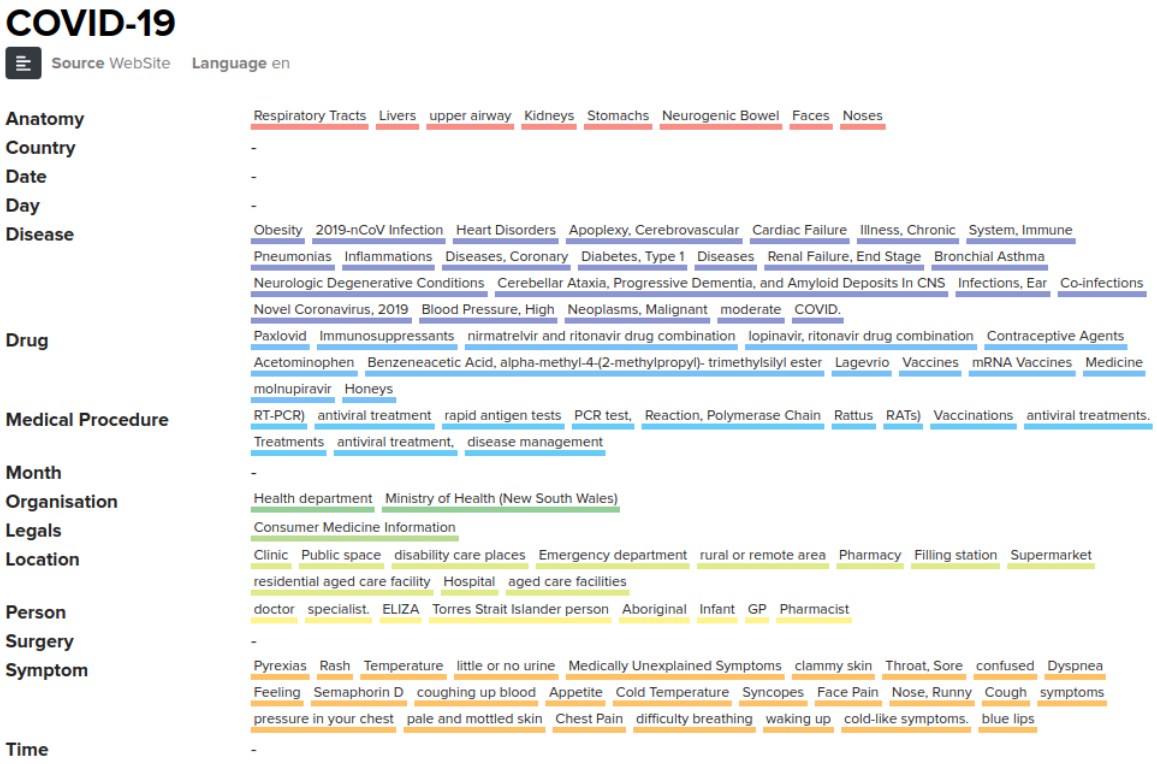

Figure 11 “Concept View” showing list of extracted concepts from the Website.

Visualization Methods

The visual web interface provides several visualization methods to analyze and narrow down the result set using the extracted concepts.

The facet view, see Figure 12, enables users to identify most frequent concepts and easily narrow down the result set.

The tag cloud, see Figure 13, and the bar chart, see Figure 14, show the most frequent concepts for one type at a time and enables users to filter the results.

The matrix visualization enables users to analyze and filter most frequent co-occurrences of two selected concepts. Figure 15 provides an example for the concepts “drug” and “anatomy” for the query “covid-19 symptom treatment”.

Figure 16 shows the knowledge graph of the most relevant Websites for the query “covid-19 symptom treatment”. Initially, only a maximum of six extracted concepts are shown. However, intelligent exploration methods, see Figure 17, enable users to explore the graph.

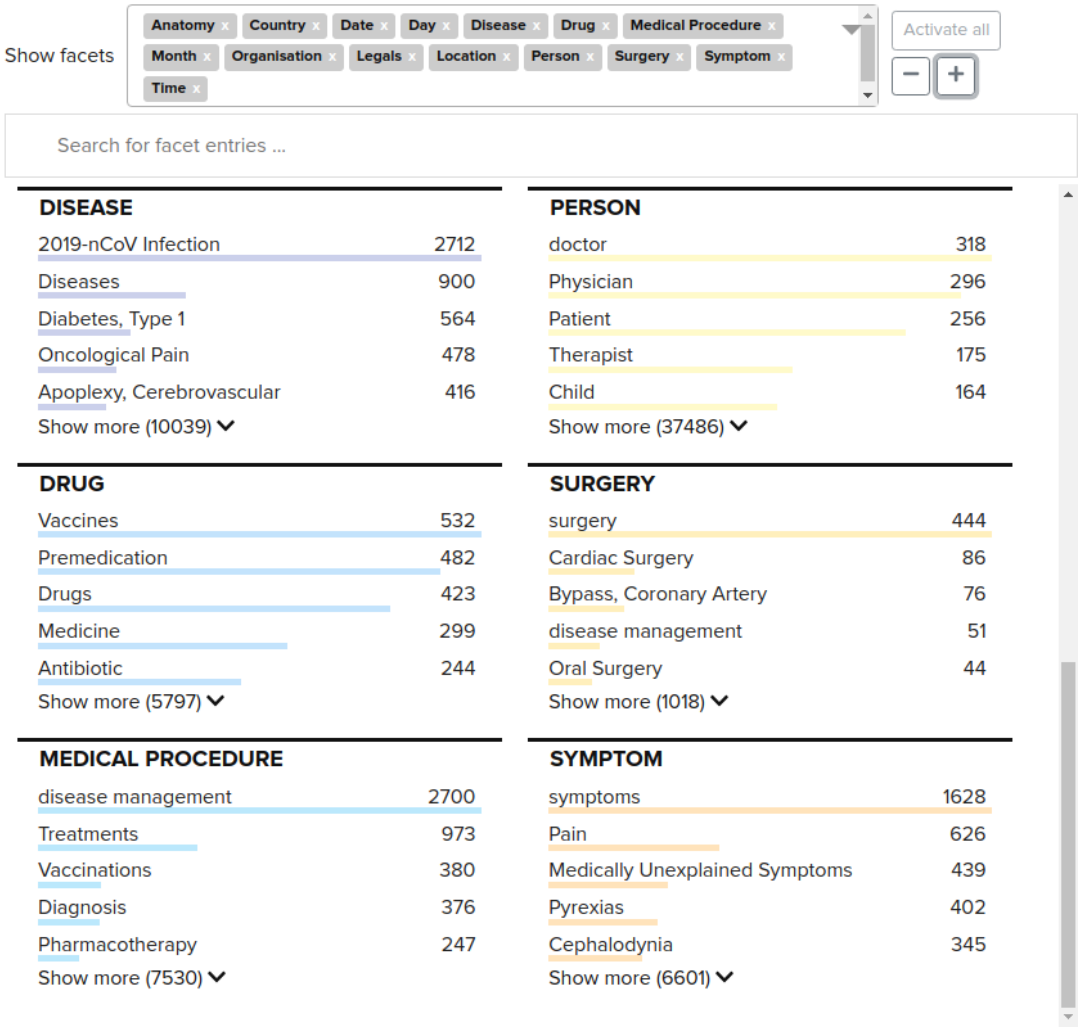

Figure 12 Facet View for query “covid-19 symptom treatment”.

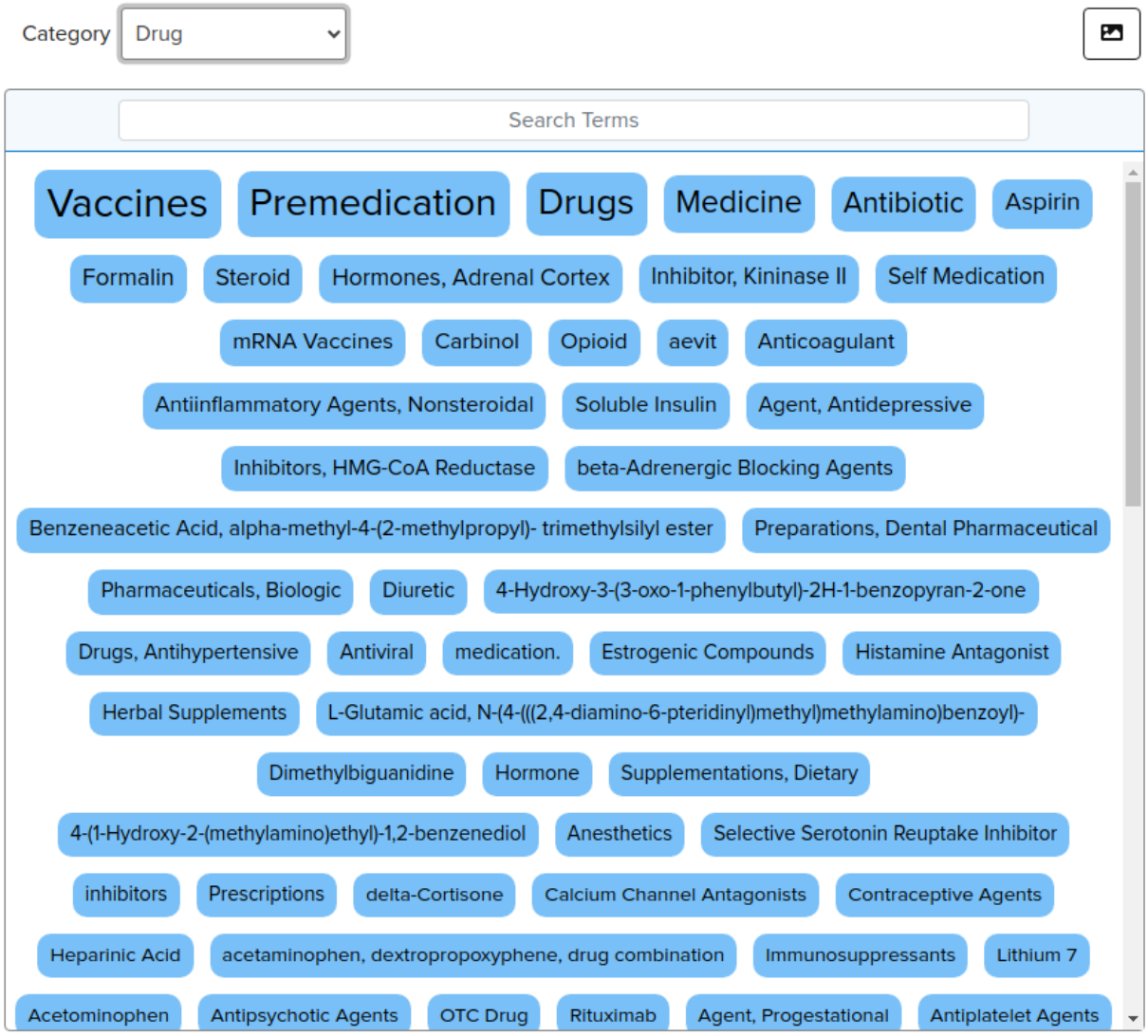

Figure 13 Tag cloud of most frequent drugs for query "covid-19 symptom treatment".

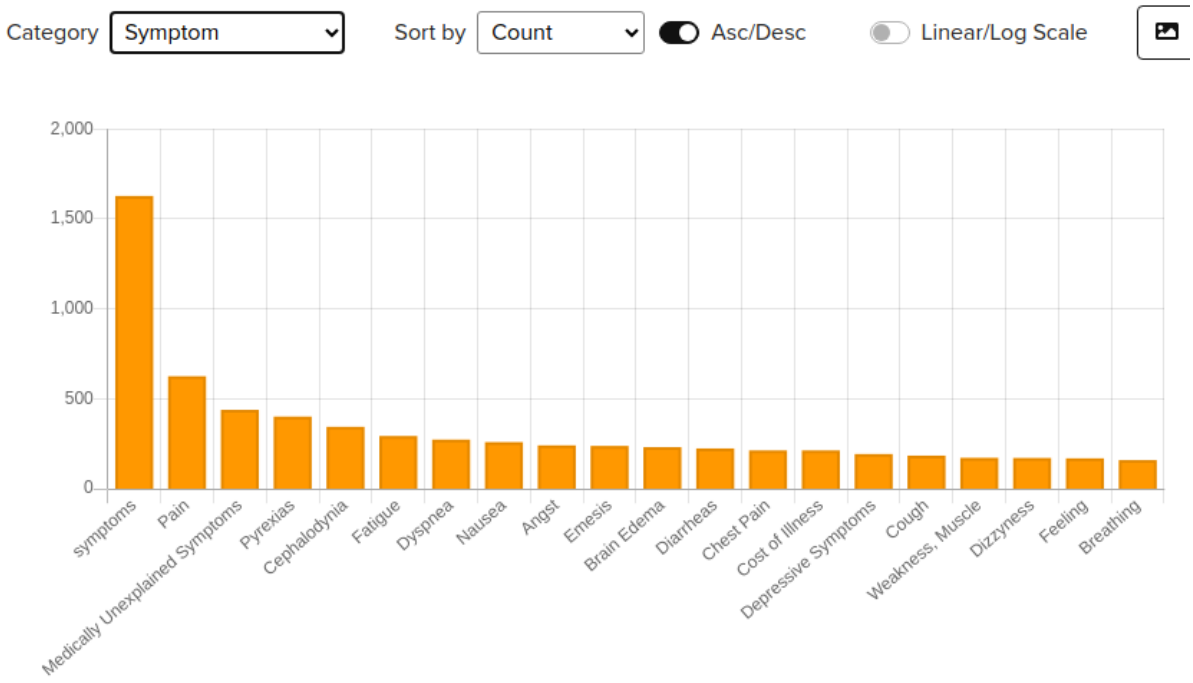

Figure 14 Bar chart of most frequent symptoms for the query "covid-19 symptom treatment".

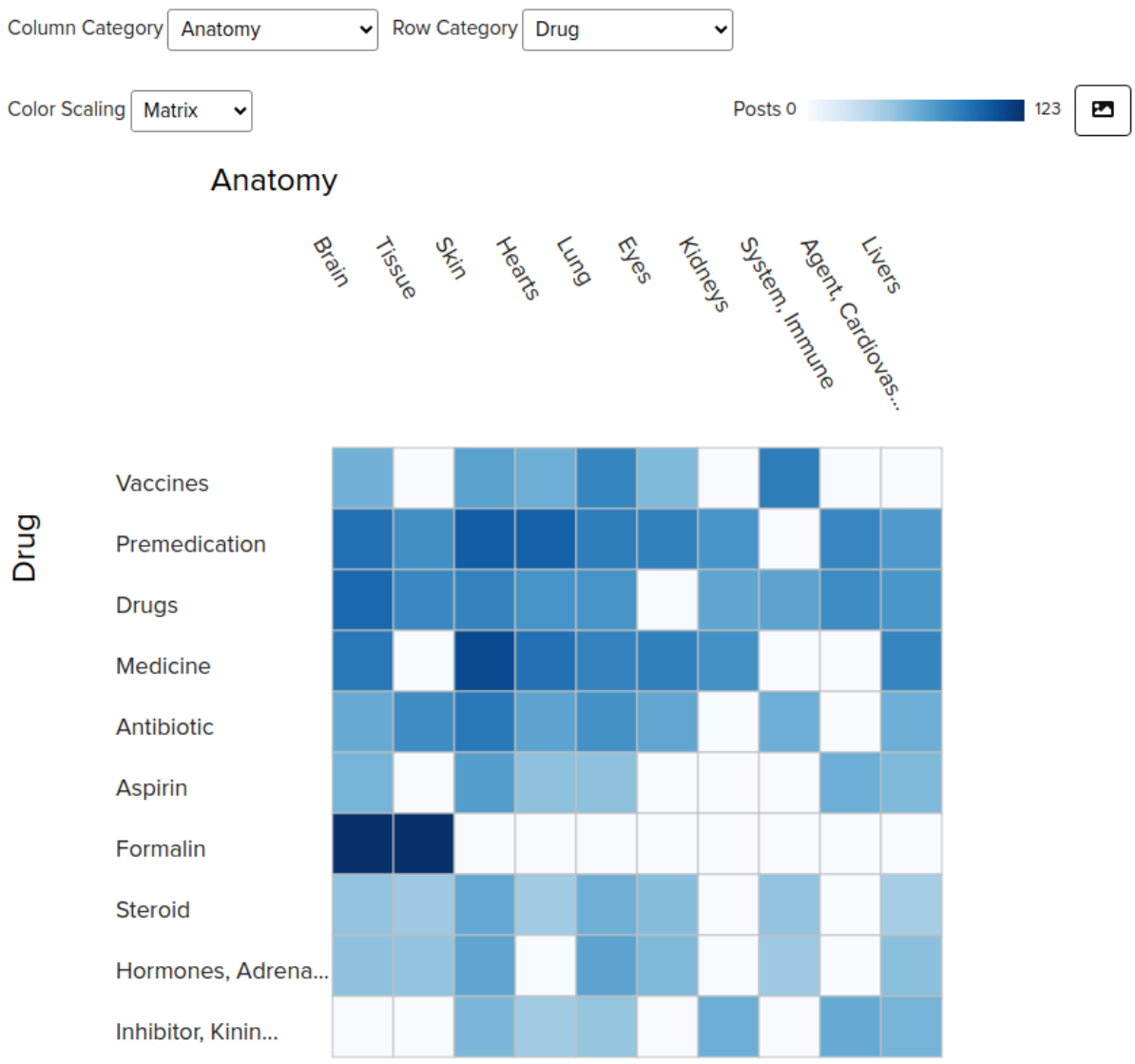

Figure 15 Most frequent co-occurrences of concepts “drug” and “anatomy” for query "covid-19 symptom treatment”.

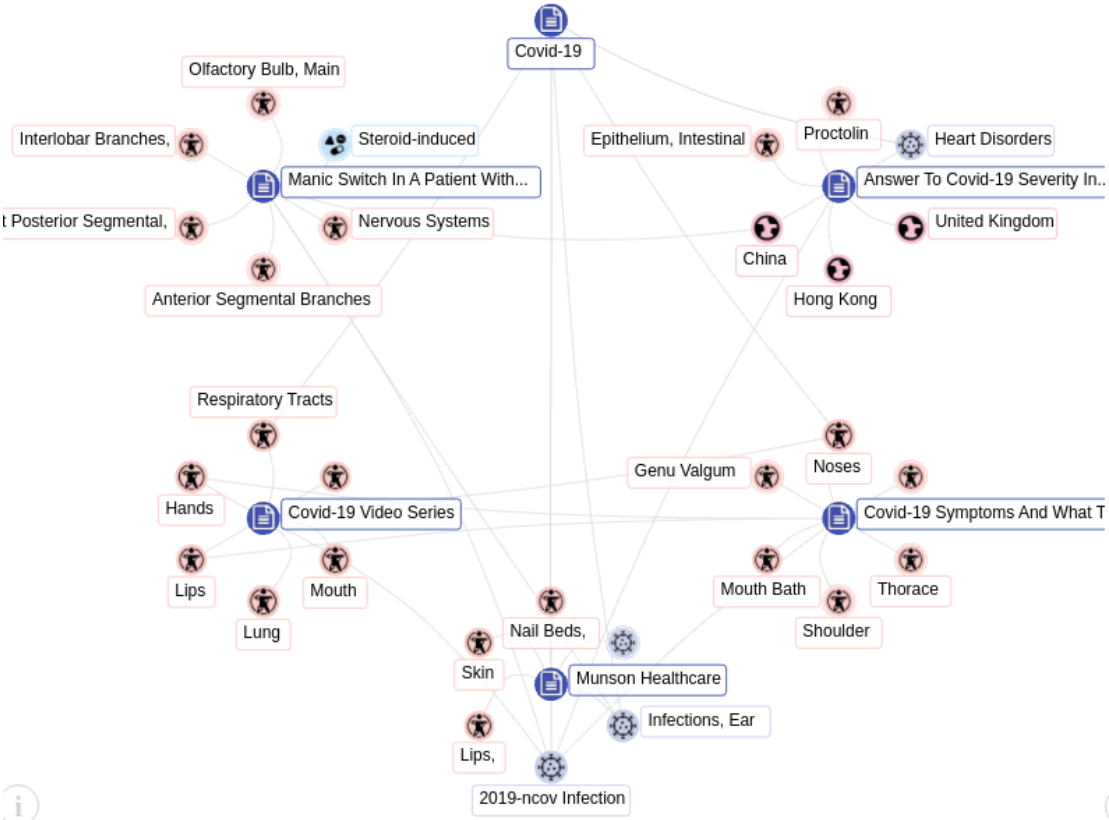

Figure 16: Knowledge graph showing most relevant documents and extracted concepts for query "covid-19 symptom treatment".

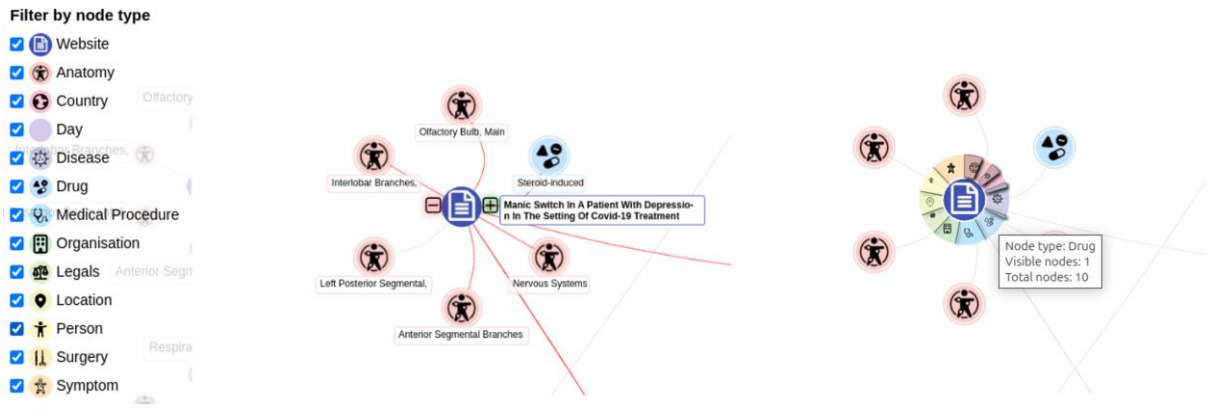

Figure 17 Knowledge graph node types (left) and graph exploration methods (center and right).

3 Data Availability

The processed website data for the Apache Solr import is available at: https://zenodo.org/records/17512328. The source code for the RAG system is available at: https://github.com/mijantscher/tilde-rag/

4 Table of Figures

Figure 1 Explorative statistics of COVID-19 related websites from the OWI 4

Figure 2 Health Knowledge Graph generated from a websites (structured) metadata and extracted and normalized entities from the website body. 5

Figure 3 Illustration of RAG architecture and overview of all included components. 6

Figure 4: Mock-up highlighting facts extracted from a document. 7

Figure 5: Application UI-Design showing highlighted text passages, extracted concepts as well as an aggregated summary to a sample question on the left side and a heat map visualization between instances of the two concepts “Diseases” and “Symptoms” on the right side. 7

Figure 6 Technology stack for a vertical, hybrid search engine in the healthcare domain. 8

Figure 7 Hybrid RAG Architecture with (i) the database indexing phase and (ii) the retrieval step 9

Figure 8: Visual Web Interface for retrieving and analyzing health-related Websites. 14

Figure 9 Answer generated for the query "covid-19 symptom treatment". 14

Figure 10 “Content View” showing Website content by highlighting extracted concepts. 15

Figure 11 “Concept View” showing list of extracted concepts from the Website. 15

Figure 12 Facet View for query “covid-19 symptom treatment”. 16

Figure 13 Tag cloud of most frequent drugs for query "covid-19 symptom treatment". 17

Figure 14 Bar chart of most frequent symptoms for the query "covid-19 symptom treatment". 18

Figure 15 Most frequent co-occurrences of concepts “drug” and “anatomy” for query "covid-19 symptom treatment”. 19

Figure 16: Knowledge graph showing most relevant documents and extracted concepts for query "covid-19 symptom treatment". 20

Figure 17 Knowledge graph node types (left) and graph exploration methods (center and right). 20

5 Bibliography

Chase, H. (2025). *LangChain (Version 0.3.66) [Computer software].* Retrieved from LangChain: https://github.com/langchain-ai/langchain

Cormack, G. V. (2009). Reciprocal rank fusion outperforms condorcet and individual rank learning methods. *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, 758-789.

Khattab, O. a. (2022). Demonstrate-Search-Predict: Composing retrieval and language models for knowledge-intensive NLP. *arXiv preprint arXiv:2212.14024*.

Medicine, N. L. (2025). *Unified Medical Language System (UMLS). U.S. Department of Health and Human Services*. Retrieved from https://www.nlm.nih.gov/research/umls

Melchiorre, A. B.-C. (2021). Investigating gender fairness of recommendation algorithms in the music domain. *Information Processing & Management*.

Nadeem, M. B. (2020). StereoSet: Measuring stereotypical bias in pretrained language models. *arXiv preprint arXiv:2004.09456.*

OpenAI. (2025). *GPT-4o Mini [Large-language-model].* Retrieved from https://platform.openai.com/docs/models/gpt-4o-mini

Parrish, A. C. (2021). BBQ: A hand-built bias benchmark for question answering. *arXiv preprint arXiv:2110.08193*.

Reimers, N. &. (2024). *all-MiniLM-L6-v2 [Machine learning model].* Retrieved from https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Solr, A. (2025). *Apache Solr (Version 9.8) [Computer software].* Retrieved from https://solr.apache.org/

Zaratiana, U. T. (2023). Gliner: Generalist model for named entity recognition using bidirectional transformer. *arXiv preprint arXiv:2311.08526.*