

Article

Search Engine for Open Geospatial Consortium Web Services Improving Discoverability through Natural Language Processing-Based Processing and Ranking

Elia Ferrari ^{1,*}, Friedrich Striewski ¹, Fiona Tiefenbacher ¹, Pia Bereuter ¹ , David Oesch ² and Pasquale Di Donato ²

¹ Institute of Geomatics, FHNW University of Applied Sciences and Arts Northwestern Switzerland, 4132 Muttenz, Switzerland; friedrich.striewski@fhnw.ch (F.S.); fiona.tiefenbacher@fhnw.ch (F.T.); pia.bereuter@fhnw.ch (P.B.)

² Federal Office of Topography Swisstopo, 3084 Wabern, Switzerland; david.oesch@swisstopo.ch (D.O.); pasquale.didonato@swisstopo.ch (P.D.D.)

* Correspondence: elia.ferrari@fhnw.ch; Tel.: +41-61-228-62-97

Abstract: The improvement of search engines for geospatial data on the World Wide Web has been a subject of research, particularly concerning the challenges in discovering and utilizing geospatial web services. Despite the establishment of standards by the Open Geospatial Consortium (OGC), the implementation of these services varies significantly among providers, leading to issues in dataset discoverability and usability. This paper presents a proof of concept for a search engine tailored to geospatial services in Switzerland. It addresses challenges such as scraping data from various OGC web service providers, enhancing metadata quality through Natural Language Processing, and optimizing search functionality and ranking methods. Semantic augmentation techniques are applied to enhance metadata completeness and quality, which are stored in a high-performance NoSQL database for efficient data retrieval. The results show improvements in dataset discoverability and search relevance, with NLP-extracted information contributing significantly to ranking accuracy. Overall, the GeoHarvester proof of concept demonstrates the feasibility of improving the discoverability and usability of geospatial web services through advanced search engine techniques.

Keywords: Natural Language Processing; result ranking; data discovery; geospatial search engine; web services; Open Geospatial Consortium; language model



Citation: Ferrari, E.; Striewski, F.; Tiefenbacher, F.; Bereuter, P.; Oesch, D.; Di Donato, P. Search Engine for Open Geospatial Consortium Web Services Improving Discoverability through Natural Language Processing-Based Processing and Ranking. *ISPRS Int. J. Geo-Inf.* **2024**, *13*, 128. <https://doi.org/10.3390/ijgi13040128>

Academic Editors: Wolfgang Kainz, Mara Nikolaidou, Christos Chalkias, Marinos Kavouras and Margarita Kokla

Received: 15 February 2024

Revised: 6 April 2024

Accepted: 10 April 2024

Published: 12 April 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The improvement of search engines on the World Wide Web has been researched in recent decades, starting from the introduction of the Semantic Web, which enriches the data with further descriptors of the content, leading to better search results for the users. A domain that has proven particularly challenging is the one involving geospatial data, specifically, in the case of geospatial web services. Despite the clear XML structure, they lack a centralized index, which thus makes it difficult for search engines to crawl and index them comprehensively [1].

Despite the introduction of standards by the Open Geospatial Consortium (OGC) for the definition of Web Map Services (WMSs), Web Map Tile Services (WMTSs), and Web Feature Services (WFSs) [2–4], as well as the widespread adoption of them, their implementation can vary significantly among different providers and, unlike other types of web services, there is no centralized or unified platform with all WMS/WMTS or WFS services listed, even within the same country. In addition, depending on the provider, some services may not have comprehensive metadata and indexes, which drastically worsens the dataset discoverability and usability [5].

All mentioned OGC web services (OWSs) are implemented to include a self-describing Service Endpoint, known as *GetCapabilities*. This operation facilitates the retrieval of an

XML document detailing the relevant information of the service. This core function can be leveraged to automatically discover and collect information about the services and elaborate the listed metadata to gain additional insight. Existing works [1,5] integrate a crawler module to collect OWSs published on the internet and validate the retrieved information and services with the *GetCapabilities* operation. As a result, further operations can be applied to the data, aiming to check the metadata quality, implement ad hoc ontology, and geocode the data [1,5].

This study focuses on the development of a proof of concept (PoC) of a search engine tailored to OGC geospatial services in Switzerland. Within the range of OWSs provided by the federal government, cantons, and municipalities, finding geospatial web services is not as efficient as an online web search. It requires expert knowledge to search for datasets within an OWS, understanding the organization of the National Geodata Infrastructure, or manually comparing data from different providers. Therefore, at the national GeoUnconference [6] workshop series in Switzerland, it was established that, today, there is a lack of a service that combines the public geodata services into a catalogue as a single point of entry and presents the Swiss geodata lake of OWSs to the users. In addition, the analysis of the web services offered by various providers in Switzerland in Section 3 reveals strong variations in metadata quality. As defined by OGC standards, some essential fields for indexing, such as keywords, are optional. As a result, such fields are frequently missing, while unstructured data, such as the description or the title, contains relevant information for the dataset discoverability but cannot be used directly for indexing.

The retrieval of information from the web has already been thoroughly explored. Dissimilar to other works [7,8] that focus on the development of a web crawler to discover data on the web, our paper focuses mainly on the semantic augmentation through Natural Language Processing and ranking methods to improve the search results presented to the user. This thematic has been marginally explored in other works, which focus on the generation of web ontologies [9–11] or the exploitation of such ontologies to describe and categorize web services [12]. However, all of these approaches rely on models to parse information from well-structured fields retrieved with *GetCapabilities*. In Section 3.3, we present that OWSs do not always contain such data and, therefore, unstructured data must be used.

The implemented PoC addresses the challenges of searching these types of web services as part of *GeoHarvester*, a research project in collaboration with the Swiss Federal Office of Topography (*Swisstopo*). The project's aim is to develop a centralized and easy-to-use search engine with an open Application Programming Interface (API) for searching OWSs in Switzerland. For the implementation of such a PoC, different aspects must be considered. First, the raw data must be scraped from the servers of different providers. Then, the collected data need to be processed, enhanced, and stored in a capable and responsive database and, finally, a proper architecture and infrastructure are needed to host the system and to enable its frictionless functioning.

With this paper, we show that the extraction of additional information from unstructured data using Natural Language Processing (NLP) and language models can contribute to improving the discoverability of the scraped OWS datasets. In 30% of the cases, the discoverability of search results is enhanced, leading to an increased presence of relevant OWS datasets in the search results. Furthermore, with our approach, the NLP-extracted information contributes to enhancing the relevance ranking of the search results, delivering, in almost all the cases, a better ranking than standard methods without extracted information.

The remainder of this paper is organized as follows. Section 2 discusses existing approaches for dataset discovery, augmentation, and visualization. In Section 3, we present our implementation of the *GeoHarvester* PoC. Section 4 covers the obtained results with the implemented solution. Finally, the results discussion, further works and conclusions are presented in Section 5.

2. Background

2.1. OGC Web Services

The Open Geospatial Consortium (OGC) has been at the forefront of developing standards that facilitate the exchange, integration, and utilization of geospatial data across diverse systems. Among its pivotal standards are the Web Feature Service (WFS), the Web Map Service (WMS), and the Web Map Tile Service (WMTS), all three integral components of the geospatial web service landscape. While the WFS plays a key role as a protocol for sharing geospatial feature data, the WMS and WMTS offer the ability to provide dynamically generated map images over the internet, with or without pre-defined map tiles. All three services share the *GetCapabilities* core functionality, which serves as a crucial operation to retrieve comprehensive metadata detailing the capabilities, configurations, and available functionalities of these services. By issuing a *GetCapabilities* request to a server, users can access essential information in XML format, such as supported operations, available layers, available data formats, coordinate reference systems, and service-specific metadata [2–4]. Understanding the structure of *GetCapabilities* within WFS, WMS, and WMTS services is therefore crucial to leverage these standards effectively as well as integrate the services. Even though the OGC has developed these standards to provide a uniform way of implementing web services, service providers structure them heterogeneously in terms of provided layers, metadata, and operations. This variation can pose challenges for users and search engines [13].

2.2. Geospatial Search Engine

A search engine for geodata is designed for the efficient discovery and retrieval of spatially referenced information. An integral part of more complex Geographic Information Retrieval (GIR) systems involves identifying and indexing geographic references in unstructured text, along with associated thematic information, allowing for targeted searches and explorations of content based on location and theme. The implementation of GIR systems differs, providing unique features, although they all aim to identify and index geographic references in unstructured text, primarily using web documents as their data source and providing the users with spatially and thematically ranked search results. The biggest challenges for such systems include the ambiguity in geographic references, the multilingual variability, the evaluation metrics and benchmarking, and the completeness of the metadata [14–16].

The information interpreted by a GIR system are then submitted in the form of a query to the geospatial search engine, which ranks the results according to relevance and prepares them for a visual representation, usually combined with a map. An index structure is required to resolve search queries efficiently. The purpose of the index is to provide efficient access to all the relevant items and, consequently, considerably reduce the amount of data that need to be processed by the ranking functions [1]. With a vast amount of data, an index can be divided into parts to allow for distributed processing, thus improving response times. In this context, the type of database and query language significantly impact the response times of search queries [17].

2.3. Ranking Methods

An essential component of each search engine is the ranking function. Ranking optimizes the user experience and handles the search ambiguity, prioritizing the most relevant parts of the query. In a spatial search engine, it is essential to handle both thematic and spatial relevance during the ranking. The spatial relationship can be calculated using a measure of spatial similarity between the document and the query, for example, by comparing the footprints of documents and queries [18] or using distances based on information about ontological relations [19]. To rank data based on their thematic correlation methods, consider occurrence and frequency. For instance, the combination between term frequency in a document and the inverse document frequency (TF-IDF) developed in [20] can be used to calculate weights for every term in each document, generating a scoring vector for each

document, which can be efficiently used to place the document in the ranking list [17]. Term-frequency-based methods can suffer from term frequency saturation. In this case, the inverse document frequency comes into play, reducing the impact of words that are common across many documents. The weighting for the ranking can be managed with the BM25 (Best Match) function, which builds upon TF-IDF and introduces the document length as an additional source of weighting [21].

In order to provide a unified ranking score, many approaches combine textual and spatial similarity using a linear combination or an additional vector space model with BM25 term weighting [14,15,17]. Once each document is represented by a vector that combines spatial and thematic components, advanced machine learning methods can be applied to rank and classify the data. Various implementations of machine learning algorithms have exploited benchmarks for the training, as well as users' preferences and behavior data, to obtain a more intelligent ranking method or to categorize the data according to specific themes [18,19].

The evaluation metrics for information retrieval can be subdivided into unranked and ranked methods. Unranked metrics focus solely on the relevance of the results, not considering the results' order, making regression metrics such as Mean Absolute Error (MAE) and Mean Squared Error (MSE) ideal. On the other hand, ranked evaluation metrics consider the results' order, introducing an additional term to only evaluate the relevant results using metrics such as precision and recall [17]. A specific metric tailored for this task is the Kendal tau distance (KTD), which is based on the number of rank inversions for each pair of documents needed to have the same order of results as the ground truth [22].

2.4. Semantic Augmentation

Document metadata are fundamental for result ranking as they include relevant information that cannot be deduced from the data. By leveraging metadata, search engines can employ ranking algorithms that consider crucial attributes, which optimize the relevance, context awareness, and efficiency of the search. Thus, the completeness and quality of the metadata are fundamental for this purpose. Unfortunately, OGC web service providers often neglect the importance of this aspect and, as introduced in Section 1, the metadata show incompleteness, low quality, or missing optional fields. In those cases, the search is limited to a few fields containing partial and structured information about the data, and other fields with unstructured information as plain text are omitted [5]. Some approaches address the issue by integrating a semantic search capability in the OGC Catalogue Service for Web (CSW) or extracting and validating new dependencies, and, in this way, allowing for a more semantically enriched geospatial data discovery through Web ontology Language (OWL) [5,23,24]. With the introduction of deep learning and language models, some works have applied these in Natural Language Processing to extract valuable insight from unstructured text [25,26]. By leveraging NLP, keywords, textual annotations, and geospatial information within the text can be identified and semantically enriched. This process includes Named Entity Recognition (NER) to identify place names and entity linking to connect textual references to geospatial ontologies. This type of semantic augmentation enhances the capabilities of geospatial search engines by making the information more accessible, discoverable, and understandable to both machines and users [27].

Another simple form of semantic augmentation that uses NLP is the translation of text or keywords, enhancing the semantic understanding of contents in different languages and enabling a better user experience.

If the search results can be improved by applying NLP to the data on the internet or in a database, the text of the query can be improved through these methods as well. The goal of query expansion is to improve the relevance of the search by considering a broader set of geospatial concepts, synonyms, or related terms. This method is particularly effective for geographical query expansion, including subsystems in the search, and extending the spatial relevance of the query [17].

2.5. Data Visualization

The user interface (UI) serves as the crucial intermediary between the user and the system, facilitating the transfer of information. Users engage with the interface to define queries, review query results, and modify their queries or filter the results as needed. Reference [28] identified the four phases where the users' key activities in information searching can be summarized: (i) In query formulation, users must be supported by the interface in formulating a query based on the information need. Once the query is established, the (ii) action phase commences, easily initiating the search through the UI. Subsequently, in the (iii) review of results, the search results must be presented in a clear way and order, facilitating users to determine their relevance. Finally, the (iv) refinement phase allows users to adjust and improve their queries based on insights gained from previous results. Based on these four core components, the UI can be extended with various elements that facilitate user interaction and query formulation [29]. Modern search interfaces offer components such as pagination controls, auto-complete functions, dynamic term suggestions, and related searches. With the onset of mobile devices, voice-driven interfaces have gained popularity, allowing users to directly voice their queries [30]. The presentation of search results can vary with different layouts depending on the content. For instance, grid-based layouts are well suited for images and graphic content, providing a more comprehensive overview. In contrast, for list results, an additional dropdown window can be employed to showcase additional entry details, ensuring a balance between an overview and the completeness of information [31].

3. Methods

In this section, we present the implemented proof of concept, explaining in detail all of its components and functionalities. For the implementation, we consider three main challenges. Firstly, by scraping OWSs, different metadata qualities and completeness of fields can be found. This can lead to bad discoverability of the data and, therefore, a preprocessing step is necessary to augment the metadata and improve the search results' quality and relevance by exploiting NLP (Section 3.3). Secondly, looking at the system scalability for the prototype phase, an extensive number of service providers leads to large databases, which can severely increase the response times to the user interface. Consequently, a performant database in combination with custom search functions is required (Section 3.4 and Section 3.5). Thirdly, users expect that a search engine delivers all the relevant search results on the first page; thus, the implementation requires a custom ranking function and fuzzy matching methods that assure that the most relevant results are presented first (Section 3.6).

3.1. Architecture

The user-facing frontend is implemented in *TypeScript* and *React.js*, facilitating user interaction, query formulation, and the presentation of search results. The frontend interacts with the backend through a REST API implemented in *Python* using *FastAPI*. It enables the functionalities shown on the user interface and interacts with the database, searching and ranking the affected data. The scraper, a backend process that automatically collects the OWS, is triggered daily. It also checks the validity of the services and stores the new and updated data entries in a temporary CSV file within the GitHub repository. Subsequently, the temporary data are ingested and preprocessed, enhancing the metadata through NLP methods, calculating the metadata quality, and storing all the information in an in-memory NoSQL database ready to be queried.

The whole system architecture and software stack are summarized in Figure 1.

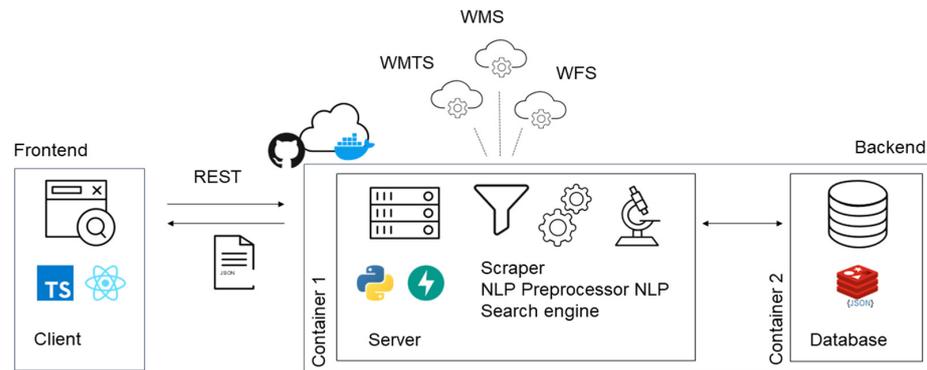


Figure 1. Frontend and backend conceptualization of the architecture used for the GeoHarvester PoC, including Scraper for OWS retrieval, NLP preprocessing, search engine logic in a first Docker container, and the Redis database in a second Docker container.

3.2. Scraper

The scraper is based on a first version developed in [32] and works as separate tier of the system, updating the index of the metadata daily by checking for modifications or newly published data layers. The automation of the process is implemented through the GitHub *Action* workflow, allowing the scraper to automatically update the metadata index overnight. In a first step, it searches for OWSs, drawing from a curated list of more than 1400 Swiss servers hosting Geoportals, and compares the services with the existing ones to discover changes in the metadata or add new entries. Among the fields retrieved (Table 1), only a few that can be used for a semantic search are mandatory for OWSs. Therefore, only the *Title*, *Name*, and *Provider* fields are compared to merge possible identical layers and perform post-processing to remove duplicates in the keywords. In addition, it validates the *GetCapabilities* service links, checking if the XML file with all information can be retrieved. Finally, the scraper stores the metadata in a temporary CSV file, structured as shown in Table 1.

Table 1. Fields extracted from OWSs (WMS/WMTS/WFS), with distinctions between OGC mandatory fields and optional fields. Fields that can be used for semantic searches are in bold.

Field Name	Description	Format	Mandatory
Provider	Manger of the data	Text	✓
Title	Short title	Text	✓
Name	Name or identifier of the layer	Text	✓
Tree	Layer tree	Tree structure	✓
Group	Category of the data	Text	
Abstract	A brief summary	Text	
Keywords	Relevant keywords	List of string	
Legend	Link to legend	URL	
Contact	Contact information	Text	
Service Link	<i>GetCapabilities</i> link	URL	✓
Publication date	Publication Date	Date	
Service type	OGC Service type	WMS/WMTS/WFS	✓
Zoom level	Max zoom level	Int	✓
Center	Lat/Lon in WGS84	Tuple of float	✓
Bounding box	Extent of data layer WSEN	List of float	✓

3.3. Semantic Augmentation and Preprocessing

As shown in Figure 2, the analysis of the collected metadata shows that among 42,000 OWS datasets (WFS, WFS, or WMTS layers) in four different languages (German, French, Italian, and English), keywords are often missing or limited to just one, while unstructured data like descriptions (abstract) or titles hold valuable information that needs to be extracted. Consequently, the metadata are augmented and integrated with additional

information coming from other fields prior to storage in the database. In a first step, the abstract is analyzed with Rapid Automatic Keyword Extraction (RAKE), a simple NLP graph-based method that does not depend on deep learning techniques, but nevertheless outperforms common term frequency (TF) methods. Its simplicity and efficiency favors its integration in applications that need to process large datasets [20].

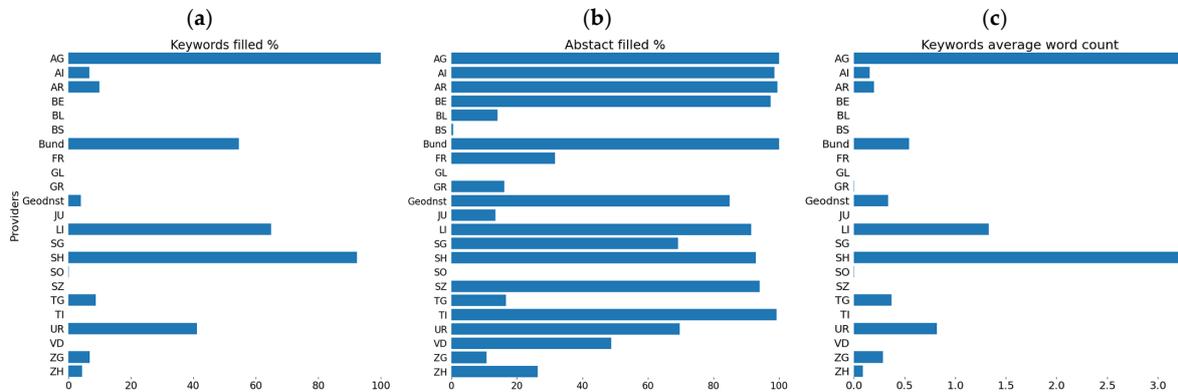


Figure 2. Use of OWS metadata in Switzerland of the investigated service providers. (a) Percentage of keyword fields filled. (b) Percentage of abstract fields filled. (c) Average number of words in filled keyword fields.

Despite its processing speed and domain independence, when compared to more sophisticated language models, RAKE has certain limitations. Its primary disadvantage lies in its reliance on statistical measures, without considering semantic relationships or contextual understanding [25]. Consequently, the method has been reinforced by applying additional keyword refinement through pretrained neural-network-based language models based on Sentence-BERT (SBERT) [33,34]. This not only allows for a deeper understanding of language semantics and syntax, but also offers numerous models for different languages. This last feature is beneficial, since the data collected span four distinct languages, allowing us to employ a dedicated model trained and optimized for each language individually [35–38]. To face the challenge of selecting the right language model for each dataset, a language detection tool has been integrated, which is based on [39] and allows us to include the detected language in the metadata as additional information. Another aspect of NLP preprocessing concerns summarization. Approximately 15% of the data have a description (abstract) longer than 20 words, with a maximum of 294 words, making the integration of the field in the search index computationally expensive. Thus, the key information of longer descriptions (abstract fields) are extracted and the text is summarized in about 20 words, keeping the key information in a couple of sentences. To this end, the SBERT [34] the NER methods are applied, exploiting their capability of capturing the semantic meaning and interconnection among sentences. Their state-of-the-art Siamese network architecture stands out among other sentence-embedding methods, showing better results and computation efficiency [34].

The last preprocessing step covers the quality of the metadata. Aiming to present transparent results to the user, a quality score is calculated on the OWS original fields, which is then also considered for the ranking, showing in which portion the ranking and relevance stem from the original data fields. All NLP-generated fields described above are summarized in Table 2.

Table 2. Fields extracted from the fields in Table 1 with the semantic augmentation and preprocessing, including the calculated field describing the metadata quality.

Field Name	Derived from Original Columns	Format
NLP keywords	Abstract	List of string
NLP summary	Abstract	Text
Metadata quality	Abstract, Keywords	Integer

3.4. High-Performance Database

Once the data are preprocessed, they need to be stored in a database. Due to the nature of the search engine type, which requires a low storage capacity but rapid and frequent access to the data, for the solution, a NoSQL database is adopted. Redis is an open-source in-memory key-value storage system that has been improved in scalability and data safety [40]. As Redis offers limited options for query functions, the latter have been divided into two phases in order to guarantee both rapid response time and optimal sorting of the search results. Firstly, as many matches as possible for each word in the query string are retrieved. These include exact matches and similar words across all relevant fields within the database. Secondly, the results of the former are ranked, scoring the matches with a custom function, described in Section 3.6, which weights the different columns and match types. In addition, pagination is applied to the search results, improving the server response times.

The response times should be as fast as possible, but some comparative measures for such a system are described in [41]. In order to keep the user's flow of thought uninterrupted, the system response times should be less than a second, while a delay longer than ten seconds is the limit to maintain the user's attention [41].

3.5. Query Expansion

To facilitate the search in the database, some core NLP functions are applied to the query string. This involves expanding the query and optimizing matches. After word tokenization and stop word removal, the resulting search tokens are stemmed using a stemming algorithm that supports various European languages, namely the Snowball Stemmer algorithm [42].

Then, both stemmed and non-stemmed tokens are used to execute the query in the database, resulting in exact matches and matches from the stemmed query, as shown in the workflow in Figure 3.

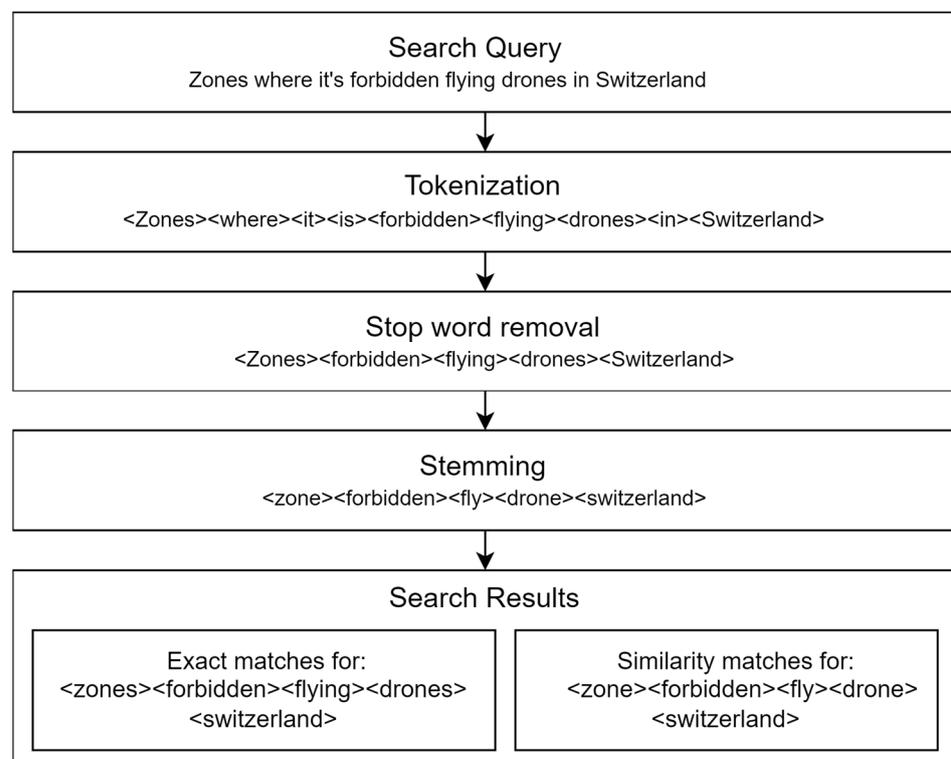


Figure 3. Steps of the query expansion process and resulting tokens for the search in the database for exact and similarity matches.

3.6. Results Ranking

The relevance of the search results is evaluated in the second phase of the search function, aiming to rank them according to the user expectation. In this stage, the search results from the initial phase (Section 3.2) are assigned weights based on two criteria: the match type, such as whether it is an exact word match or if the query word is merely contained, and the match column, assuming that the information contained in manually entered fields, like the *Title* and *Keywords*, are more relevant than the others. Additionally, the scores are been weighted, considering the length of the text, assuming that an exact match in a short text has more relevance than one in a longer text, as an extensive text could contain additional side information, which may not be the focus of the OWS dataset. These weights are then utilized to compute the ranking score, which in turn allows the search results to be sorted.

As it is important for the user to have more relevant results first, instead of using unranked metrics as in other works [9,43], we adopted a ranked metric. For each query, a ground truth order of the results is manually established; then, for the evaluation of the ranking method, the Kendal tau distance [22] is applied (Equations (1)–(3)).

$$K_n = 1 - \frac{2K(\tau_1, \tau_2)}{n - (n - 1)} \quad (1)$$

where

$$K(\tau_1, \tau_2) = \sum_{(j,i)} K_{ji}(\tau_1, \tau_2) \quad (2)$$

where

$$K_{ji}(\tau_1, \tau_2) = \begin{cases} 0 & \text{if } x_j, x_i \text{ are in the same order in } \tau_1 \text{ and } \tau_2 \\ 1 & \text{if } x_j, x_i \text{ are in the inverse order in } \tau_1 \text{ and } \tau_2 \end{cases} \quad (3)$$

The Kendall tau distance counts the pairwise disagreements between items from two rankings: τ_1 (ground truth ranking) and τ_2 (resulting ranking). A penalty point is added for each necessary pairwise swap to bring the elements x_j and x_i in the same order as in τ_1 . Finally, the resulting sum is normalized by the number of elements in the ranking list, n .

4. Results

In this section, we present the results of the implemented PoC, analyzing the implemented system in Section 4.1, the OWS discoverability in Section 4.2, and the search results' ranking in Section 4.3. Finally, the user interface is briefly presented in Section 4.4.

4.1. System Response Times

As the ranking method strongly influences the response times, the search function is divided into two phases (Section 3.4). This approach improved the response time to return the search results in less than a second to the front end. As shown in Figure 4, the ranking function processing time in the second phase increases exponentially with the number of words contained in the query. Thus, the ranking function can process a maximum of three words in order to meet an acceptable response time and keep the users' flow of thought uninterrupted.

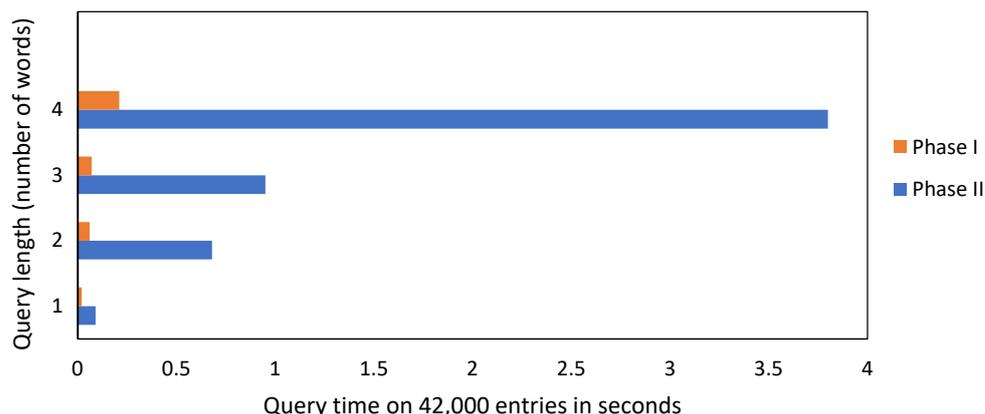


Figure 4. Two-phase query times on Redis database.

4.2. OWS Dataset Discoverability

The evaluation of dataset discoverability compared search results with and without NLP-extracted information to evaluate the improvement of the search result with the enrichment of the metadata with NLP. The datasets of the service provider cover their administrative area, and the metadata of these datasets refer to the extent and set theme. A search for a particular municipality would only return a result if its name was mentioned in the metadata of a dataset. Table 3 shows the evaluation of the number of search results of a search for Swiss municipalities in the generated database, with the objective of retrieving OWS datasets that contain information about that municipality. The selected municipalities are towns with more than 15,000 inhabitants [44], and municipalities that are contained in the name of the canton are excluded to avoid ambiguity in the results.

Table 3. Additional OWS datasets (WMS/WFS/WMTS) discovered with NLP-extracted information in comparison to title and keyword search. The municipalities are ordered by number of inhabitants, excluding Zurich, Geneve, Basel, Bern, Sankt Gallen, Lucerne, Fribourg, Schaffhausen, Zug, Aarau, and Schwyz. Highlighted in grey are the municipalities that do not match any dataset in the database independent of the search method.

Municipality	OWS Datasets Discovered		Municipality	OWS Datasets Discovered		Municipality	OWS Datasets Discovered	
	Without Extracted Information	With Extracted Information		Without Extracted Information	With Extracted Information		Without Extracted Information	With Extracted Information
Lausanne	0	116	Meyrin	0	0	Schlieren	0	0
Winterthur	10	11	Carouge	0	0	Adliswil	0	0
Biel	34	52	Kreuzlingen	0	0	Volketswil	0	0
Thun	12	16	Wädenswil	0	0	Thalwil	0	0
Bellinzona	162	162	Riehen	158	158	Olten	0	3
Uster	4	8	Allschwil	0	0	Pully	0	0
Vernier	0	0	Renens	6	6	Regensdorf	0	0
Chur	0	1	Wettingen	0	3	Ostermundiger	0	0
Sion	0	0	Nyon	0	2	Littau	0	0
Yverdon	0	0	Bülach	0	0	Pratteln	0	0
Emmen	6	10	Vevey	0	0	Freienbach	84	84
Dübendorf	0	0	Opfikon	0	0	Wallisellen	0	0
Rapperswil	0	0	Reinach	0	1	Wohlen	0	1
Dietikon	0	2	Baden	4	7	Morges	0	0
Wetzikon	0	0	Onex	0	0	Steffisburg	0	0

The results in Table 3 show how many additional relevant OWS datasets could be discovered for each municipality by exploiting NLP-extracted information.

4.3. Search Results' Ranking

The second evaluation focuses on assessing the quality of the ranking. Several user queries were analyzed, aiming to compare the quality of the GeoHarvester ranking function,

which incorporates NLP-extracted information, against conventional ranking methods based on similarity matching and applied to different fields of the database. Initially, the GeoHarvester system refines the user-typed queries, aiming to minimize the number of words searched in the database while retaining all relevant information. Before the comparison, the first 15 searched OWS datasets were selected and sorted manually by relevance as ground truth. Subsequently, the ranking quality scores were calculated using the Kendall tau distance [22] and comparing the first 15 search results with the ground truth. Given that the ground truth queries may not consistently contain the same number of OWS datasets, the Kendall tau distance was normalized with the number of entries in the ground truth and inverted to yield an ascending KTD score, as explained in Section 3.6.

The sorting methods involved various column combinations, utilizing the title column, the keywords column, and the NLP-extracted information. As shown in Table 4, in almost all cases, the use of NLP-extracted information delivered better ranking results (the higher the better). In addition, the document store related to each query was analyzed, comparing the potential exact OWS dataset matches in the store and the number of potential thematic similar OWS dataset matches contained in the database. As the database has limited entries in comparison to a web search, these values explain how successfully the desired datasets could be found among other thematically similar datasets within the database.

Table 4. Ranking results of the test queries executed with different column combinations as well as the document store with potential exact matches and potential thematic matches. The KTD column denotes the score calculated using the column title, using the column keywords, using a combination of title and keywords columns, and using a combination of all three columns (title, keywords, and NLP-extracted information). * Field used to find thematic similar matches in the database (in bold).

User Query	NLP-Refined Query * Search Topic Term in Bold	KTD Score Columns Used for Ranking				Document Store	
		Title	Keywords	Title + Keywords	Title + Keywords + NLP Extraction	Potential Exact Matches	Potential Thematic Matches *
Eignung der Solarenergie in der Schweiz	<eignung>< solarenergie > <schweiz>	0.89	0	0.92	1	4	21
Eignung der Solarenergie in Kanton Aargau	<eignung>< solarenergie > <kanton aargau>	0.82	0	0.8	0.88	2	21
Rohstoffe in der Schweiz	< rohstoff ><schweiz>	0.1	0	0.27	0.84	8	20
Rohstoffe in Kanton Schaffhausen	< rohstoff ><kanton schaffhausen>	0	0	0	0.87	1	20
Wildtierkorridore in der Schweiz	< wildtierkorridor > <schweiz>	0.2	0	0.08	0.68	4	43
Wildtierkorridore in Kanton Solothurn	< wildtierkorridor > <kanton solothurn>	0.18	0	0.18	0.84	1	43
Radwege in der Schweiz	< radweg ><schweiz>	0.74	0	0.1	0.6	4	74
Velowege in der Schweiz	< veloweg ><schweiz>	0.66	0	0.14	0.98	4	74
Radwege in Zürich	< radweg ><zürich>	0.8	0.11	0.8	1	1	74
Radwege in Kanton Schwyz	< radweg ><kanton schwyz>	0.53	0	0.53	1	2	74
Bewilligungen von der Wasserbauabteilung in Zürich	<bewilligung> < wasserbauabteilung > <zürich>	0.93	0.71	0.93	0.96	1	10
Bezirke der Kanton Zürich	< bezirk ><kanton zürich>	0.76	0.16	0.75	0.96	3	19
Römische Pfosten Augusta Raurica	< römisch >< pfosten > <augusta raurica>	1	0.89	1	1	1	37
Berufsinformationszentren in Kanton Bern	< berufsinformationszentre > <kanton bern>	0.87	0	0.87	1	4	10
Fotopunkte der Amphibienzugstelle	<fotopunkt> < amphibienzugstelle >	0.72	0.66	0.72	0.88	11	21
Einschränkungen für Drohne in der Schweiz	< einschränkung > < drohne ><schweiz>	0.91	0.44	0.93	0.94	3	10

4.4. GeoHarvester PoC Prototype

To enable users to interact with all the functionalities but at the same time not be overwhelmed, the user interface adopts a minimalist design, focusing solely on core features. Illustrated in Figure 5, it provides a concise overview of the search results while also allowing users to access additional information about the services. Users can sort and filter search results by provider, service type, and metadata quality. Furthermore, to facilitate integration in GIS Software, the corresponding layer definition file can be downloaded or directly visualized on the *Swisstopo* geoportal.

(a)

Title	Abstract	Provider	Service	Metaquality
Bienenstandorte	Der Datensatz umfasst alle Bienenstände mit Imker im Kanton Schaffhausen sowie angrenzende vom Kanton Zürich und der deutschen Enklave Bisingen. Zur Bekämpfung von Bienenkrankheiten ist das Wissen über den genauen Standort der Bienenstände von zentraler Bedeutung.	swisstopo	WMS	75
Bienenstandorte	Standorte von Bienenhäusern, welche von den Aargauischen Bienenzüchtervereinen gemeldet wurden.	swisstopo	WMS	50
Bienenstandorte	Standorte von Bienenhäusern, welche von den Aargauischen Bienenzüchtervereinen gemeldet wurden.	swisstopo	WMS	50
Bienenstandorte KT	Darstellung der Bienenstandorte und Bienensterrgebiete.	swisstopo	WMS	50
Bienenstandorte KT	Darstellung der Bienenstandorte und Bienensterrgebiete.	swisstopo	WMS	50
Bieneninspektionskreise	Der vorliegende Geobasisdatensatz enthält die Bieneninspektionskreise. Diese werden vom Departement Gesundheit und Soziales eingeteilt. Pro Inspektionskreis ist ein Bieneninspektor unter der Leitung der Kantonstierärztin zuständig für den Vollzug der Tierseuchenverordnung. Die Aussengrenzen der Inspektionskreise folgen den Gemeindegrenzen.	swisstopo	WMS	50
Bieneninspektionskreise	Der vorliegende Geobasisdatensatz enthält die Bieneninspektionskreise. Diese werden vom Departement Gesundheit und Soziales eingeteilt. Pro Inspektionskreis ist ein Bieneninspektor unter der Leitung der Kantonstierärztin zuständig für den Vollzug der Tierseuchenverordnung. Die Aussengrenzen der Inspektionskreise folgen den Gemeindegrenzen.	swisstopo	WMS	50
Bienenstände	Standorte, an denen Bienenvölker gehalten werden. Die Standorte sind klassiert (Bienenhaus, Wanderwagen, Magazine, andere).	swisstopo	WFS	50
Bienenstände	Standorte, an denen Bienenvölker gehalten werden. Die Standorte sind klassiert (Bienenhaus, Wanderwagen, Magazine, andere).	swisstopo	WMS	50
Bienenstände	Bienenstände als Punktdaten (Kanton Bern)	swisstopo	WMS	50

(b)

© 2023 GeoHarvester | Ein Projekt in Zusammenarbeit mit dem Institut Geomatik, FHNW und swisstopo

Figure 5. GeoHarvester user interface: (a) presentation of search results for the query <bees> in German, (b) drop-down menu with export and visualizations options of the same query.

5. Conclusions and Future Work

The implemented proof of concept demonstrates the feasibility of collecting OGC web services from different providers, unifying them in a single portal. The proposed solution is performant and improves OWS dataset discovery by leveraging unstructured data through NLP extraction methods. Nevertheless, the complexity of the ranking function resulted in significantly slower response times for queries longer than three words, thereby impacting overall performance. This issue could be addressed by preprocessing queries and extracting the most pertinent information restraining the search tokens.

The adopted system's architecture, with a separate tier for the scraper, facilitates the exploitation of NLP techniques on the collected data before the presentation to the user. In addition, the combination of a simple NLP graph-based method and language models delivered the best refined results, which could be used to improve the OWS datasets' discoverability and the search results' ranking.

Findings emerging from the analysis of the OWS datasets' discoverability concerning the spatial relevance indicated that in just 30% of cases, additional relevant OWS datasets could be discovered with NLP-extracted information, while in 9% of the cases (Bellinzona, Riehen, Renens, Freienbach), the same number of OWS datasets could be found without extracted information. Conversely, in the remaining cases, no OWS datasets could be found with both methods. This can be attributed to providers potentially not offering OWSs related to those specific municipalities or the absence of evidence within the OWSs, indicating their affiliation with said municipalities.

Moreover, enhancing OWS datasets' discoverability could be achieved by leveraging the NLP-extracted information to rank search results and implementing a customized ranking function, which outperformed similarity methods adopted in other works [7,8] based on the Kendal tau distance as an evaluation metric.

These findings also suggest that optional fields, such as keywords, are often missing, and therefore significantly diminish ranking performance when solely relied upon. It can be supposed that in cases where all optional fields are missing and insufficient information is present in the title, even with existing methods, discovering such OWS datasets would prove challenging. An alternative solution could involve additional enhancement of the information starting from other mandatory fields, such as *Title* and *Name*, and using a combination of ontologies and language models to extract implicit information contained.

Future studies should investigate the spatial relevance of the results in more depth, exploiting the related bounding box extensions of the data entries and providing additional qualitative results, as demonstrated in prior studies [14,16]. Moreover, the NLP-based extraction methods can be improved, including the generation of an ontology for the data domain with the assistance of language models, with the aim of searching for related words as illustrated in previous works [43,45].

Although queries are currently presented in German due to the concentration of data and users in the German-speaking part of Switzerland, future work will focus on implementing multi-language support for the frontend and backend, thus enabling a language-independent search returning search results from additional languages.

To improve the user experience, a language model can be trained for topic modelling and applied to the data to classify them into related categories, such as INSPIRE categories, enabling users to further filter the search results.

Finally, the PoC was implemented using Swiss servers and Swiss national languages; however, the system can be easily adapted to scrape other servers outside Switzerland by extending the server list. In addition, SBERT-based models for NLP information extraction can be trained on publicly available datasets (Section 3.1) to support further languages and augment the corresponding metadata.

Author Contributions: Conceptualization, Elia Ferrari, David Oesch, Friedrich Striewski and Pia Bereuter; methodology, Elia Ferrari, David Oesch, Friedrich Striewski and Pia Bereuter; software, Friedrich Striewski, Elia Ferrari, Fiona Tiefenbacher, Pasquale Di Donato and David Oesch; validation, Friedrich Striewski, Pia Bereuter and Elia Ferrari; formal analysis, Elia Ferrari; investigation, Elia Ferrari and Pia Bereuter; resources, Friedrich Striewski; data curation, Elia Ferrari, Pia Bereuter and Friedrich Striewski; writing—original draft preparation, Elia Ferrari; writing—review and editing, Elia Ferrari, Friedrich Striewski and Pia Bereuter; visualization, Fiona Tiefenbacher and Friedrich Striewski; supervision, Pia Bereuter; project administration, Pasquale Di Donato; funding acquisition, Pia Bereuter. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in the context of the Swiss Geoinformation Strategy by the Federal Coordination Body for Geoinformation (GKG) and the Swiss Conference of Directors of Construction, Planning and Environment (BPUK): <https://www.geo.admin.ch/en/strategy-and-implementation> (accessed on 1 January 2024).

Data Availability Statement: Example data and code associated with this PoC are available on GitHub (<https://github.com/FHNW-IVGI/Geoharvester> (accessed on 1 January 2024)). The GeoHarvester prototype is online and freely accessible (<https://geoharvester.ch/> (accessed on 1 January 2024)).

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Ma, J.; Co, J.E.; Quintanilla, A. A Semantic Index Structure for Integrating OGC Services in a Spatial Search Engine. In Proceedings of the 2010 IEEE Conference on Open Systems (ICOS 2010), Kuala Lumpur, Malaysia, 5–7 December 2010; IEEE: Piscataway, NJ, USA, 2010; pp. 103–108.
2. De la Beaujardiere, J. OpenGIS®Web Map Server Implementation Specification 2006. Available online: https://portal.ogc.org/files/?artifact_id=14416 (accessed on 11 November 2023).
3. Maso, J.; Pomakis, K.; Julià, N. OpenGIS®Web Map Tile Service Implementation Standard 2010. Available online: https://portal.ogc.org/files/?artifact_id=35326 (accessed on 11 November 2023).
4. Vretanos, P.A. Web Feature Service Implementation Specification 2005. Available online: https://portal.ogc.org/files/?artifact_id=8339 (accessed on 11 November 2023).
5. Yue, P.; Di, L.; Zhao, P.; Yang, W.; Yu, G.; Wei, Y. Semantic Augmentations for Geospatial Catalogue Service. In Proceedings of the 2006 IEEE International Symposium on Geoscience and Remote Sensing, Denver, CO, USA, 31 July–4 August 2006; IEEE: Piscataway, NJ, USA, 2006; pp. 3486–3489.
6. Oesch, D. Resultate Der GeoUnconference—Thema 16—Service-Verzeichnis 2022. Available online: <https://github.com/GeoUnconference/discussions/discussions/38> (accessed on 29 November 2023).
7. Bone, C.; Ager, A.; Bunzel, K.; Tierney, L. A Geospatial Search Engine for Discovering Multi-Format Geospatial Data across the Web. *Int. J. Digit. Earth* **2016**, *9*, 47–62. [[CrossRef](#)]
8. Huang, C.-Y.; Chang, H. GeoWeb Crawler: An Extensible and Scalable Web Crawling Framework for Discovering Geospatial Web Resources. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 136. [[CrossRef](#)]
9. Miao, L.; Guo, J.; Cheng, W.; Zhou, Y. A Novel Model to Support OGC Web Services Semantic Search Using OWL-S. In Proceedings of the 2016 24th International Conference on Geoinformatics, Galway, Ireland, 14–20 August 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1–4.
10. Saquicela, V.; Vilches-Blázquez, L.M.; Freire, R.; Corcho, O. Annotating OGC Web Feature Services Automatically for Generating Geospatial Knowledge Graphs. *Trans. GIS* **2022**, *26*, 505–541. [[CrossRef](#)]
11. Miao, L.; Liu, C.; Fan, L.; Kwan, M.-P. An OGC Web Service Geospatial Data Semantic Similarity Model for Improving Geospatial Service Discovery. *Open Geosci.* **2021**, *13*, 245–261. [[CrossRef](#)]
12. Halilali, M.S.; Gouardères, E.; Gaio, M.; Devin, F. Geospatial Web Services Discovery through Semantic Annotation of WPS. *ISPRS Int. J. Geo-Inf.* **2022**, *11*, 254. [[CrossRef](#)]
13. Shen, S.; Liu, W.; Wu, H.; Chen, Y. A Multi-Level Comprehensive Evaluation Method for Quality of WMS Based on Fuzzy Mathematics. In Proceedings of the 2009 17th International Conference on Geoinformatics, Fairfax, VA, USA, 12–14 August 2009; IEEE: Piscataway, NJ, USA, 2009; pp. 1–5.
14. Woodruff, A.G.; Plaunt, C. GIPSY: Automated Geographic Indexing of Text Documents. *J. Am. Soc. Inf. Sci.* **1994**, *45*, 645–655. [[CrossRef](#)]
15. Amitay, E.; Har’El, N.; Sivan, R.; Soffer, A. Web-a-Where: Geotagging Web Content. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Sheffield, UK, 25 July 2004; ACM: New York, NY, USA, 2004; pp. 273–280.

16. Purves, R.S.; Clough, P.; Jones, C.B.; Arampatzis, A.; Bucher, B.; Finch, D.; Fu, G.; Joho, H.; Syed, A.K.; Vaid, S.; et al. The Design and Implementation of SPIRIT: A Spatially Aware Search Engine for Information Retrieval on the Internet. *Int. J. Geogr. Inf. Sci.* **2007**, *21*, 717–745. [CrossRef]
17. Manning, C.D.; Raghavan, P.; Schütze, H. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008; ISBN 978-0-521-86571-5.
18. Frontiera, P.; Larson, R.; Radke, J. A Comparison of Geometric Approaches to Assessing Spatial Similarity for GIR. *Int. J. Geogr. Inf. Sci.* **2008**, *22*, 337–360. [CrossRef]
19. Andrade, L.; Silva, M. Relevance Ranking for Geographic IR. In Proceedings of the 3rd ACM Workshop on Geographic Information Retrieval, Seattle, WA, USA, 10 August 2006.
20. Rose, S.; Engel, D.; Cramer, N.; Cowley, W. Automatic Keyword Extraction from Individual Documents. In *Text Mining*; Berry, M.W., Kogan, J., Eds.; Wiley: Hoboken, NJ, USA, 2010; pp. 1–20, ISBN 978-0-470-74982-1.
21. Robertson, S.; Walker, S.; Jones, S.; Hancock-Beaulieu, M.; Gatford, M. *Okapi at TREC-3*; National Institute of Standards and Technology (NIST): Gaithersburg, MD, USA, 1994.
22. Kendall, M.G. A New Measure of Rank Correlation. *Biometrika* **1938**, *30*, 81–93. [CrossRef]
23. Larson, R.R. Ranking Approaches for GIR. *SIGSPATIAL Spec.* **2011**, *3*, 37–41. [CrossRef]
24. Chen, L.; Cong, G.; Jensen, C.S.; Wu, D. Spatial Keyword Query Processing: An Experimental Evaluation. *Proc. VLDB Endow.* **2013**, *6*, 217–228. [CrossRef]
25. Ji, X.; Sungu-Eryilmaz, Y.; Momeni, E.; Rawassizadeh, R. Speeding Up Question Answering Task of Language Models via Inverted Index. *arXiv* **2022**. [CrossRef]
26. Park, D.; Ahn, C.W. Self-Supervised Contextual Data Augmentation for Natural Language Processing. *Symmetry* **2019**, *11*, 1393. [CrossRef]
27. Jehangir, B.; Radhakrishnan, S.; Agarwal, R. A Survey on Named Entity Recognition—Datasets, Tools, and Methodologies. *Nat. Lang. Process. J.* **2023**, *3*, 100017. [CrossRef]
28. Shneiderman, B.; Byrd, D.; Croft, W.B. Sorting out Searching: A User-Interface Framework for Text Searches. *Commun. ACM* **1998**, *41*, 95–98. [CrossRef]
29. Purves, R.S.; Clough, P.; Jones, C.B.; Hall, M.H.; Murdock, V. Geographic Information Retrieval: Progress and Challenges in Spatial Search of Text. *FNT Inf. Retr.* **2018**, *12*, 164–318. [CrossRef]
30. Sarhan, S. Smart Voice Search Engine. *J. Comput. Appl.* **2014**, *90*, 40–44. [CrossRef]
31. Roy, N.; Maxwell, D.; Hauff, C. Users and Contemporary SERPs: A (Re-)Investigation Examining User Interactions and Experiences. *arXiv* **2022**. [CrossRef]
32. Oesch, D. Geoservice Harvester POC Open Geo Services Reported by the Swiss Gov Agencies and Third Parties 2023. Available online: https://github.com/davidoesch/geoservice_harvester_poc (accessed on 15 August 2023).
33. Honnibal, M.; Boyd, A.; Van Landeghem, S.; Montani, I. spaCy: Industrial-Strength Natural Language Processing in Python. 2020. Available online: <https://zenodo.org/doi/10.5281/zenodo.1212303> (accessed on 15 August 2023).
34. Reimers, N.; Gurevych, I. Sentence-BERT: Sentence Embeddings Using Siamese BERT-Networks. *arXiv* **2019**, arXiv:1908.10084.
35. Bosco, C.; Lombardo, V.; Vassallo, D.; Lesmo, L. Building a Treebank for Italian: A Data-Driven Annotation Schema. In Proceedings of the Second International Conference on Language Resources and Evaluation (LREC'00), Athens, Greece, 31 May–2 June 2000; Gavrilidou, M., Carayannis, G., Markantonatou, S., Piperidis, S., Stainhauer, G., Eds.; European Language Resources Association (ELRA): Paris, France, 2000.
36. Weischedel, R.; Palmer, M.; Marcus, M.; Hovy, E.; Pradhan, S.; Ramshaw, L.; Xue, N.; Taylor, A.; Kaufman, J.; Franchini, M.; et al. *OntoNotes Release 5.0*; 2806280 KB; Linguistic Data Consortium: Philadelphia, PA, USA, 2013.
37. Brants, S.; Dipper, S.; Eisenberg, P.; Hansen-Schirra, S.; König, E.; Lezius, W.; Rohrer, C.; Smith, G.; Uszkoreit, H. TIGER: Linguistic Interpretation of a German Corpus. *Res. Lang. Comput.* **2004**, *2*, 597–620. [CrossRef]
38. Candito, M.; Seddah, D. Le Corpus Sequoia: Annotation Syntaxique et Exploitation Pour l'adaptation d'analyseur Par Pont Lexical. In Proceedings of the TALN 2012—19e Conférence sur le Traitement Automatique des Langues Naturelles, Grenoble, France, 4–8 June 2012.
39. Shuyo, N. Language Detection Library for Java 2010. Available online: <http://code.google.com/p/language-detection/> (accessed on 10 November 2023).
40. Chen, S.; Tang, X.; Wang, H.; Zhao, H.; Guo, M. Towards Scalable and Reliable In-Memory Storage System: A Case Study with Redis. In Proceedings of the 2016 IEEE Trustcom/BigDataSE/ISPA, Tianjin, China, 23–26 August 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 1660–1667.
41. Card, S.K.; Robertson, G.G.; Mackinlay, J.D. The Information Visualizer, an Information Workspace. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems Reaching through Technology—CHI' 91, New Orleans, LA, USA, 27 April–2 May 1991; ACM Press: New York, NY, USA, 1991; pp. 181–186.
42. Porter, M.F. An Algorithm for Suffix Stripping. *Program* **1980**, *14*, 130–137. [CrossRef]
43. Chen, J.; Jiménez-Ruiz, E.; Horrocks, I.; Antonyrajah, D.; Hadian, A.; Lee, J. Augmenting Ontology Alignment by Semantic Embedding and Distant Supervision. In *The Semantic Web*; Verborgh, R., Hose, K., Paulheim, H., Champin, P.-A., Maleshkova, M., Corcho, O., Ristoski, P., Alam, M., Eds.; Lecture Notes in Computer Science; Springer International Publishing: Cham, Switzerland, 2021; Volume 12731, pp. 392–408, ISBN 978-3-030-77384-7.

44. Federal Statistical Office Permanent Resident Population by Category of Citizenship and Sex by Canton and City, 1999–2022. 2022. Available online: <https://www.bfs.admin.ch/asset/en/26565157> (accessed on 12 November 2023).
45. Elnagar, S.; Yoon, V.; Thomas, M.A. An Automatic Ontology Generation Framework with An Organizational Perspective. In Proceedings of the Hawaii International Conference on System Sciences 2020, Honolulu, HI, USA, 7–10 January 2020; ScholarSpace: Kathmandu, Nepal, 2020.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.