International Journal of Geo-Information

MDPI

*Article*

# Geo-L: Topological Link Discovery for Geospatial Linked Data Made Easy

**Christian Zinke-Wehlmann** † **and Amit Kirschenbaum** *,†

Institut für Angewandte Informatik an der Universität Leipzig (InfAI), Goerdelerring 9, 04109 Leipzig, Germany; zinke@infai.org

* Correspondence: amit@informatik.uni-leipzig.de

† These authors contributed equally to this work.

**Abstract:** Geospatial linked data are an emerging domain, with growing interest in research and the industry. There is an increasing number of publicly available geospatial linked data resources, which can also be interlinked and easily integrated with private and industrial linked data on the web. The present paper introduces Geo-L, a system for the discovery of RDF spatial links based on topological relations. Experiments show that the proposed system improves state-of-the-art spatial linking processes in terms of mapping time and accuracy, as well as concerning resources retrieval efficiency and robustness.

**Keywords:** geospatial analysis; linked data; semantic web; topological relations



**Citation:** Zinke-Wehlmann, C.; Kirschenbaum, A. Geo-L: Topological Link Discovery for Geospatial Linked Data Made Easy. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 712. https://doi.org/10.3390/ijgi10100712

Academic Editors: Wolfgang Kainz, Rob Brennan, Brian Davis, Armin Haller and Beyza Yaman

Received: 11 August 2021
Accepted: 11 October 2021
Published: 19 October 2021



## 1. Introduction

The Web of Data, or the Semantic Web, is a continuously growing global data space [1]. Semantic Web standards, such as the RDF (Resource Description Framework) [2,3], OWL (Web Onthology Language) [4,5], and SPARQL (SPARQL Protocol and RDF Query Language) [6] were developed to express and exchange semantic information on the web, to address the goal of semantic interoperability [7]. In the geospatial context, most prominent is the GeoSPARQL initiative, which offers a necessary vocabulary to develop geo-related data on the Semantic Web [8]. In recent years, geospatial linked data has gained increasing attention [9], also due to advances in the Earth Observation domain [10]. Thus, numerous resources of linked geospatial data have been developed, e.g., LinkedGeoData [11], Smart Point Of Interest [12], Spanish Cases [13], and Ireland's national geospatial data [14]; the domain is constantly growing within the Linked Data Cloud. Notably, the domain of geospatial linked data contains complex datasets, such as NUTS [15], which describe territories using polygons that may be more than 1700 vertices long.

According to the linked data principles, published data should be interlinked with other datasets on the web [16]. In general, the linking (and fusing) of geospatial linked data sources enable large-scale inferences and data integration [17]. Nevertheless, explicit links are often not part of the dataset and should be discovered automatically, even in a distributed cloud environment and huge datasets. These linking activities are one pillar to foster the development of innovative software solutions. In particular, the linking of geospatial data is a challenging task, since the links express relations which depend on complex geometric computations and a naive computation of such relations between two datasets requires the testing of all pairs of objects, one of each dataset, respectively, resulting in a quadratic time complexity.

The present work introduces Geo-L, a system for the discovery of spatial links in RDF datasets according to topological relations. Geo-L was developed considering the following requirements, which we identified by comparing existing approaches, services, and tools for this task:

1. Scalability and efficiency: As mentioned before, the linked data cloud is continually growing, employing new sources and datasets, and the service should be able to handle big datasets. The idea is to provide a service for different linked data environments (open or closed). Therefore, the time required for linking has to be minimized, and the vision is to discover even extensive datasets in near real time.
2. Robustness: The service must retain functionality under unforeseen conditions, such as corrupted data. This is especially true for crowd-sourced or automatically generated datasets, which are likely to include errors as the size of data grows.
3. Interoperability and flexibility: The service has to be handled as easily and transparently as possible. The (SPARQL affine) user should be able to easily formulate queries to retrieve source and target datasets, as well as the linking condition. This includes the ability to handle data of different formats, as datasets are heterogeneous. For example, the computation of topological relations requires that geometries are represented similarly, e.g., in WKT format. However, if the datasets use different formats, then the service has to provide the means to unify these representations. The service has to operate easily as a standalone system, as a module integrated into other applications, or through RESTful API.
4. Quality: Given two sets of RDF resources with geospatial data, $\mathcal{S}$ and $\mathcal{R}$, and a spatial predicate, $P$, the service shall return all the links between the resources $s \in \mathcal{S}$ and $r \in \mathcal{R}$ which satisfy $P$ (see more in Section 2).

## 2. Background

Linked Data is a method which uses RDF format to publish structured and machine-readable data on the Web, and employ RDF links to express explicit interconnections between data items from various data resources. The result is referred to as the Web of Data [16]. The basic idea of link discovery is to find data items within the target dataset which are logically connected to the source dataset. More formally, this means: given $\mathcal{S}$ and $\mathcal{T}$, sets of RDF resources, called source and target resources, respectively, and a relation, $R$, the aim of link discovery methods is to find a mapping, $M$, such that $M = \{(s,t) \in \mathcal{S} \times \mathcal{T} : R(s,t)\}$. A naive computation of $M$ requires a quadratic time complexity to test for every $s \in \mathcal{S}$ and $t \in \mathcal{T}$ whether $R$ holds, which is unfeasible for large datasets.

In a geospatial context, $\mathcal{S}$ and $\mathcal{T}$ are sets of spatial objects, which contain geometries in a two-dimensional space as features; the links may be based on proximity or on topological relations. In the latter case, relations are expressed by the Dimensionally Extended nine-Intersection Model (DE + 9IM) [18,19], which was accepted as an ISO standard [20]. DE + 9IM classifies binary spatial relationships between two geometries, $a$ and $b$, which may be points, lines, or polygons, based on the intersection of the interiors ($I$), boundaries ($B$) and exteriors ($E$) of $a$ with those of $b$. A combination of these six geometric features define topological relations, which are described in a $3 \times 3$ matrix as follows:

$$\text{DE + 9IM}(a,b) = \begin{bmatrix} dim(I(a) \cap I(b)) & dim(I(a) \cap B(b)) & dim(I(a) \cap E(b)) \\ dim(B(a) \cap I(b)) & dim(B(a) \cap B(b)) & dim(B(a) \cap E(b)) \\ dim(E(a) \cap I(b)) & dim(E(a) \cap B(b)) & dim(E(a) \cap E(b)) \end{bmatrix}$$

The intersection $S$ of some feature of $a$ with a feature of $b$ may be either empty or in itself a geometric object, namely: a point, a line, or a polygon. $dim(S)$ returns the dimension of the geometry $S$; if $S$ consists of multiple geometries, then $dim(S)$ is the maximal dimension of intersection if it is of multiple parts.

$$dim(S) = \begin{cases} -1 & \text{if } S = \varnothing \\ 0 & \text{if } S \text{ contains at least one point,} \\ & \text{but no lines or polygons} \\ 1 & \text{if } S \text{ contains at least one line,} \\ & \text{but no polygons} \\ 2 & \text{if } S \text{ contains at least one polygon} \end{cases}$$

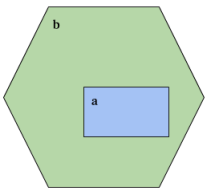
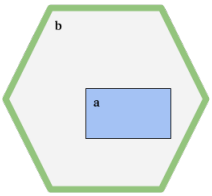
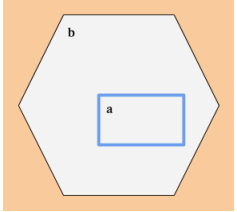
In addition to the dimension values, the matrix may contain the values T ($dim(S) \geq 0$), F ($dim(S) = -1$), and $*$ ("do not-care" value, which means that the value in this matrix cell has no influence on the outcome of a function applied to this matrix): The model defines topological predicates to describe the spatial relations between the two geometries in a compact and human-interpretable manner, which are defined by pattern matrices: *equals*, *disjoint*, *intersects*, *touches*, *crosses*, *overlaps*, *within*, and *contains*. For example, the pattern matrix for the relation *within* is defined by the following pattern matrix (see also Strobl [21]).

$$a.within(b) = \begin{bmatrix} \mathtt{T} & * & \mathtt{F} \\ * & * & \mathtt{F} \\ * & * & * \end{bmatrix}$$

formally described as $(I(a) \cap I(b) \neq \varnothing) \wedge \neg(I(a) \cap E(b) \neq \varnothing) \wedge \neg(B(a) \cap E(b) \neq \varnothing)$.

To illustrate how this matrix and, hence, the formula define the *within* relation consider Figure 1, which shows two geometries $a$ and $b$, such that $a$ is within $b$. Table 1 graphically depicts the respective features $f_1(a), f_2(b)$, such that $f_1, f_2 \in \{I, B, E\}$, used in each component of the *within* formula, for those two geometries, as well as the dimension of their intersection. As can be observed, the conditions of the topological relation *within* are satisfied.

**Table 1.** Geometry features of components of *within* formula and dimension of their intersections.

| $f_1(a), f_2(b)$ | | $dim(f_1(a) \cap f_2(b))$ |
|---|---|---|
| $I(a), I(b)$ |   | 2 |
| $I(a), E(b)$ |   | −1 |
| $B(a), E(b)$ |   | −1 |

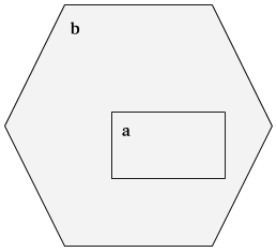



**Figure 1.** a within b.

## 3. Related Work

The link discovery of topological relations among RDF datasets has received growing interest in recent years, and various methods for this problem have been proposed. These methods usually define the topological relations between two geometries based on their relations computed between their minimum bounding boxes. A minimum bounding box (MBB) is the rectangle of the minimum area that encloses all coordinates of geometry and is commonly used as an approximation to the geometry to reduce computational costs that involve this geometry [22].

Smeros and Koubarakis [23] use the *MultiBlocking* technique [24] to discover topological relations. This technique divides the Earth's surface into curved rectangles and assigns each geometry to all blocks in which it intersects, based on the geometry's MBB. Relations discovered within each block are, then, aggregated to construct the links. This method is embedded in the Silk framework [25].

RADON [26] divides the space into hypercubes and uses optimized sparse space tiling to index geometries. This is performed by mapping each geometry to the set of hypercubes over which its minimum bounding box (MBB) spans. The method first indexes geometries $s \in \mathcal{S}$ and, then, only indexes geometries $t \in \mathcal{T}$ that may potentially reside in hypercubes already contained in the index. To minimize the size of the index, the method implements a swapping strategy, that is, prior to the indexing phase, it calculates an estimated total hypervolume (*eth*) for each of the datasets $\mathcal{S}$ and $\mathcal{T}$. If $eth(\mathcal{T}) < eth(\mathcal{S})$, then it swaps the two datasets and computes the reverse relation of the requested relation $R$. The link generation itself is conducted using a method that reduces computations on a subset of DE + 9IM relations. RADON is implemented as part of the LIMES framework [27,28]

Faria et al. [29] adapt the AgreementMakerLight (AML) [30], a framework for automated ontology matching, to tackle the task of topological relations. This is performed by utilizing ESRI Geometry API [31], which uses quadtree as means to index geometries and detect a topological relationship among them.

These methods, as well as OntoIdea [32], were evaluated on several sets of geometries: Achichi et al. [33] apply them to discover topological relations between LineStrings, constructed of trajectories from the TomTom [34] dataset. Saveta et al. [35] apply these methods to find relations between LineStrings to LineStrings and between LineStrings to Polygons, from the TomTom dataset and Spaten dataset [36], respectively. All datasets included at most 2000 instances. Both evaluations report that the methods mentioned above discover links correctly, that is, the *F*-score of most of them is 1.0 (apart from OntoIdea, whose *F*-score lies between 0.91 and 0.99, and did not take part in the tasks for link discovery between linestrings and polygons).

Strabon [37] is an open-source geospatial RDF store. It is based on the RDF4J (previously Sesame) RDF store and adds geospatial capabilities to it by implementing the OGC-standard GeoSPARQL, where, as part of the implementation, the stored geometries in Strabon are indexed with an R-Tree-over-GiST. Implementing GeoSPARQL means that Strabon includes topological functions; thus, queries that use these functions can be viewed as a means to discover topological relations. Sherif et al. [26] compares the performance of Silk, Strabon, and RADON, where they are applied to discover links between different subsets of NUTS and *CORINE Land Cover* [38] datasets, which map land and

land-usage, respectively. The biggest dataset used in their experiments consists of 2,209,538 resources. The evaluations compare the running times of these methods with different dataset sizes. It has already been acknowledged that a significant portion of big data is geospatial data [39,40]; thus, our interest lies in the performance of these systems on large datasets. Table 2 summarizes how well the methods described above perform, regarding the criteria for useful geospatial link discovery systems, discussed in Section 1, as reported in the literature [26,33,35].

As can be observed in Table 2, the LIMES system, that implements RADON, was the one that completed all the link discovery tasks for all topological relations and performed best for most of them. We, therefore, took LIMES as our main reference point. Nevertheless, LIMES as it is (we used version 1.5.5, the latest version available at the time of writing) is not sufficiently flexible to accommodate geospatial linked data of different formats, and requires an external pre-processing of the input. Additionally, LIMES assumes an error-free download and curated datasets, which is not always the case in reality. This motivated us to incorporate advantages of existing techniques in a single solution and test what existing technologies might be used for an efficient, flexible, robust, and interoperable system for on-the-fly semantic linking of geospatial data.

**Table 2.** Comparison of properties of systems for topological link discovery.

| System | Scalability and Efficiency | | Robustness | | Interoperability and Flexibility | |
|---|---|---|---|---|---|---|
| Silk | − | long running time on large datasets | −<br>− | instances limited to size of 64 K<br>not evaluated for relations *cover* and *covered by* | +<br>+<br>−<br>− | standalone framework<br>has REST and programmable APIs<br>linkage definition language is restricting<br>does not support transformation of geospatial linked data |
| AML | +<br>− | achieves best run time for *touches* and *intersects* for LineStrings<br>long running time on large datasets for LineString/ Polygon tasks for *contains*, *within*, and *covers* | −<br>− | reaches time limit for *disjoint* (75 min)<br>no information is given about error handling | +<br>− | uses ESRI, an external module for handling geometries<br>strict linkage definition |
| OntoIdea | −<br>− | long running time on large datasets<br>not evaluated for large datasets | −<br>− | not evaluated for *disjoint*<br>no information about error handling | − | no specification given |
| Strabon | + | run time for *intersects* on smaller datasets is better than that of LIMES | −<br>−<br>− | does not finish any experiment on a large dataset within the time limit (2 h)<br>does not provide feedback about progress of its task<br>no transparent error handling | + | implements GeoSPAQRL; thus, is able to transform geospatial object in retrieval time |
| LIMES | +<br>+ | addresses all tasks regarding topological link discovery<br>achieves the best runtime performance for most of the topological relations (except intersect, and touches) | − | data or server error interrupt whole process | +<br>− | can be applied as part of a framework or as a part of an application via its API<br>strict linkage definition (XML), no direct SPARQL support |
| Geo-L | +<br>+ | addresses all tasks regarding topological link discovery<br>achieves the best runtime performance for all topological relations | +<br>+ | storing chunks of datasets regularly minimizes data loss if connection is interrupted due to, e.g., server error<br>provides feedback about task progress | +<br>+ | can be applied as an independent application or through its API (as well as via REST API)<br>supports dataset definition via SPARQL query |

## 4. Geo-L

We developed a system for topological link discovery for geospatial linked data, which provides the required functionality and shows a high performance and accuracy. Geo-L also offers flexible configuration options for the SPARQL affine user as well as accurate error handling.

### *4.1. Input*

The input for a link discovery task provides the resources to be linked and the conditions upon which the links are generated, in a simple, yet flexible manner. In particular, our method offers a way to retrieve relevant properties from the endpoint, which may be either remote or local, via a SPARQL query; thus, it natively supports the manipulation of data without any need for external pre-processing. This is useful, for example, when geometry values at the endpoint are not represented in a format that directly allows computations of topological relations.

### *4.2. Download*

Downloading from a SPARQL endpoint might occasionally be interrupted before the complete dataset has been delivered. To avoid a total loss of data, our solution does not store all the data in memory while downloading, but, instead, periodically writes smaller chunks to the disk. In addition, a download might take a relatively long time due to the application implementation itself. Our solution seeks to improve this state by reducing the application overhead when querying the remote endpoint.

### *4.3. Caching*

To accelerate access to the source and target resources, we incorporated a caching mechanism. Data retrieved from the SPARQL endpoint are stored in a central data store with an internal index. Further requests for data items from the same endpoint were first served from the cache if the items were already indexed. This ensures a single local resource parallel to the endpoint, which may handle many configurations; thus, saves both time and storage. This differs from the behaviour of LIMES, where data items may be downloaded multiple times, and duplicates of the data may then be stored. Algorithm 1 sketches the caching process. The method essentially compares the required triples range to the triple indices stored in an internal database, based on the offset and limit parameters given in the configuration. It detects the indices of triples which are not already stored, retrieves the respective triples in chunks from the endpoint, and stores them in the database.

### *4.4. Link Discovery*

The task of topological link discovery requires to identify topological relations between geometries, according to the DE + 9IM model, and to efficiently process spatial data. Therefore, we used R-trees [41] as our underlying data structure. An R-tree is a data structure used to store and query multi-dimensional objects, in a way that preserves spatial relations, as vicinity and nesting, among the indexed objects. An R-tree represents each object by its minimum bounding box (MBB) and a leaf node stores the MBB of that object and a pointer to the actual geometry. An R-tree is organized hierarchically; it groups MBBs by proximity and represents them by their MBB in a higher level of the tree. This process proceeds until all the MBBs are nested in a single bounding box—the tree root. R-Trees have shown to be efficient in processing spatial joins, to find topological relations between different datasets [42]. R-Trees support both individual element searches as well as a range search, where all the items within a rectangle are retrieved.

A practical problem occurs when the data contain errors, i.e., invalid geometries. The implications of using such data are wrong results, application performance issues, etc. For this reason, geometries are examined before indexing; invalid geometries are not indexed and, thus, do not participate in the link discovery.

**Algorithm 1:** Dataset Caching

```
Input: resource.endpoint, resource.id resource.geo, config.offset, config.limit
/* create table if not exists for resource */
T ← get-table(resource,DB)
if T ∉ DB then
    T ← create-table(resource.id,resource.geo, server-offset)
    create GIST-index(T.geo)
/* checking cache */
min-offset ← offset
[min-server-offset,max-server-offset] ← get-stored-offsets()
max-offset ← offset + limit - 1
if min-server-offset > 0 and max-offset < min-server-offset then // all queried records are before all stored ones
    /* download triples with from the given offset, save after every chunk */
    retrieve-triples(resource.endpoint, offset,limit,chunksize,T)
else if min-offset > max-server-offset then // all queried records are after all stored ones

    if endpoint-has-more-entries(min-offset + 1) then // are there any more entries at this offset?
        retrieve-triples(resource.endpoint, offset,limit,chunksize,T)
else // queried entries and stored entries overlap
    /* find intervals of triple indices to be downloaded */
    intervals = list() // list of range pairs, i.e., (start-range, end-range)
    if offset < min-server-offset then
        interval.append((offset, min-server-offset - 1))
        min-offset = min-server-offset
    if max-offset > max-server-offset then
        if endpoint-has-more-entries(max-server-offset + 1) then
            intervals.append((max-server-offset + 1,max-offset))
        max-offset ← max-server-offset
    missing-limit = max-offset - min-offset + 1
    /* find intervals of triple indices to be downloaded */
    missing-intervals = find-missing-data(min-offset, missing-limit)
    missing-intervals = concat (intervals, missing-intervals)
    if length(missing-intervals) > 0 then
        foreach interval ∈ missing-intervals do
            interval-offset ← interval[0]
            interval-limit ← interval[1] - interval-offset + 1
            retrieve-triples(resource.endpoint, interval-offset,interval-limit,chunksize,T)
```

### 4.5. Implementation

We used Python as our preferred programming language, since it became the language of choice for data science in general, and provides useful tools for handling geospatial data, in particular. We experimented with the following technologies:

#### 4.5.1. GeoPandas

Our initial implementation involved custom-built caching and mapping mechanisms. We used Python's GeoPandas library [43], which implements data structures for storing geometric types, as well as analysis tools for geospatial data. In particular, GeoPandas provides an interface for spatial joins, which allow combining observations stored in these data structures based on their spatial relations. For this purpose GeoPandas indexes geometries using R*-Tree [44], a variant of R-Tree —both have a similar hierarchical structure, and aim at minimizing the total area covered. R*-Tree, however, provides a better search performance, at the cost of an increased construction time. GeoPandas currently supports finding the following spatial relations: *within*, *intersects*, and *contains*.

We further experimented with Cython [45], a language which is a superset of Python, where code can be compiled directly to C, generating efficient code. GeoPandas has been reimplemented in Cython in a way that optimizes the storage of geometries and *should* improve the performance of spatial operations.

#### 4.5.2. PostgreSQL

Furthermore, we implemented the system using PostgreSQL, an open source object-relational DBMS, withthe PostGIS extension, which provides functionality to manage geospatial data, such as geometry data types, efficient indexing, and spatial joins, and is compliant with the Open Geometry Consortium (OGC) OpenGIS "Simple Features for SQL" specifications [46]. PostGIS implements spatial indexing with an R-Tree-over-GiST [47]. GiST, Generalized Search Tree [48], is a height-balanced tree structure and allows arbitrary indexing schemes. The choice to use this as the backend of our system was multi-fold:

- GiST indexes are "null safe"; therefore, attempting to build an R-Tree on data which contain an empty geometry field will fail.
- GiST uses a compression technique which results in fast indexing.
- The database facilitates the implementation of the resource caching mechanism

The source code of Geo-L is available at https://github.com/DServSys/Geo-L (accessed on 11 October 2021 ).

## 5. Experimental Settings

### 5.1. Datasets

The evaluation was performed by finding different relations between points to polygons, and polygons to polygons in the following datasets.

- SPOI—Smart Points of Interest: A dataset which contains over 30 million Points of Interest important for tourism around the world [49].
- OLU—Open Land-Use: Maps land use on a local and regional level; contains over 11 million geometries—Polygons and MultiPolygons [50].
- NUTS—Nomenclature of Territorial Units for Statistics: A standard for referencing European countries and their regions, for statistical processes [15].

These datasets are stored in the SPARQL endpoint of the FOODIE project [51] (see also Data Availability Statement).

While SPOI and OLU are excellent examples for big (open) linked data, NUTS is a standard schema. NUTS geometries are not represented in WKT format, but use the NeoGeo vocabulary [52], and must be manipulated to conform to the form required by the procedures of topological relations computation. Tools such as LIMES, however, do no support such cases.

We compared the performance of LIMES and Geo-L with respect to both topological relations discovery and data retrieval time from endpoints.

### 5.2. Experiments

The performance of the Geo-L systems was evaluated in terms of runtime by conducting experiments on simulation test sets, as well as real-world scenarios. We also noted differences in linking results if they occurred. In order to compare the performance of our system with that of LIMES, which was implemented in a parallelized framework, the task was viewed as consisting of two stages: download and caching, and linking; we report the performance for each of them. The simulations enabled the evaluation of system performance under realistic conditions, with scenarios which otherwise might not have been explored and, at the same time, providing a reliable way to confirm their results. All experiments were performed on a 64-bit Linux machine with an Intel Core i7-7800X CPU @ 3.50 GHz and a total of 12 threads (six CPU cores $\times$ two threads per core).

#### 5.2.1. Simulation

Our simulations consisted of finding topological relations where the subsets of OLU dataset were used as both source and target datasets. This setting had multiple advantages: First, it allowed to demonstrate the benefits of caching, regarding datasets retrieval. Additionally, the structure of the OLU set, which consisted of separate geometries with non-hierarchical relations, facilitated the link quality evaluation. We used this approach to perform a preliminary comparison of three implementations on a subset of 165,000 entities (as source and target sets) and observed that the implementations which used GeoPandas performed considerably slower than the one which employed PostgreSQL with PostGIS. For example, the mapping time required for calculating the *within* relation was 38 s for the implementation which used GeoPandas, about 20 min for the GeoPandas Cython implementation (apparently, GeoPandas has an open issue about the performance of the Cython implementation; see https://github.com/geopandas/geopandas/issues/563, accessed on 11 October 2021), and less than 4 s for the implementation which used PostgreSQL. Therefore, in the following experiments, the latter served as our reference system.

We tested the systems with two subsets: the one containing the first 165,000 geometries, and the other with the first 400,000 geometries. Figure 2 compares the retrieval times of OLU subsets for both LIMES and Geo-L. The first scenario showed that retrieval time for LIMES was about twice as long compared to Geo-L. The reason was that LIMES does not detect whether data already exist or not, and downloads the same OLU subset twice, both as source and target datasets. The second scenario emphasized this phenomenon: whereas Geo-L retrieved only the data which have not been already downloaded, and performed this only once, LIMES retrieved the subset of 400,000 geometries twice, which took more than six times longer.

Moreover, LIMES stored redundant data, e.g., as we tested with two subsets—the first 165,000 geometries and the first 400,000 geometries—, and used them as both source and target datasets, the subset of the first 165,000 geometries was stored four times, as it was contained in the 400,000 geometries subset.

Experiments were repeated ten times for each topological relation type per subset, and the average mapping times are shown for both LIMES and Geo-L in Figures 3 and 4. As can be observed, Geo-L discovered topological links faster than LIMES, for all relations in these experiments. The coefficients of the variation (CV) of runtimes for the different experiments were found to be low in all cases (CV $< 0.1$), which indicated that these results were consistent.

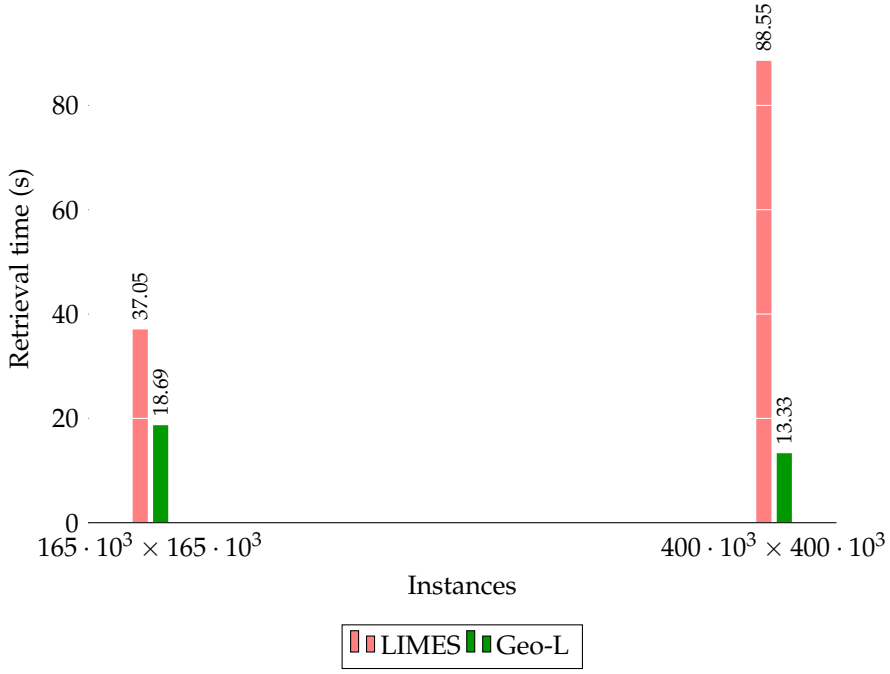



**Figure 2.** Retrieval time OLU–OLU.

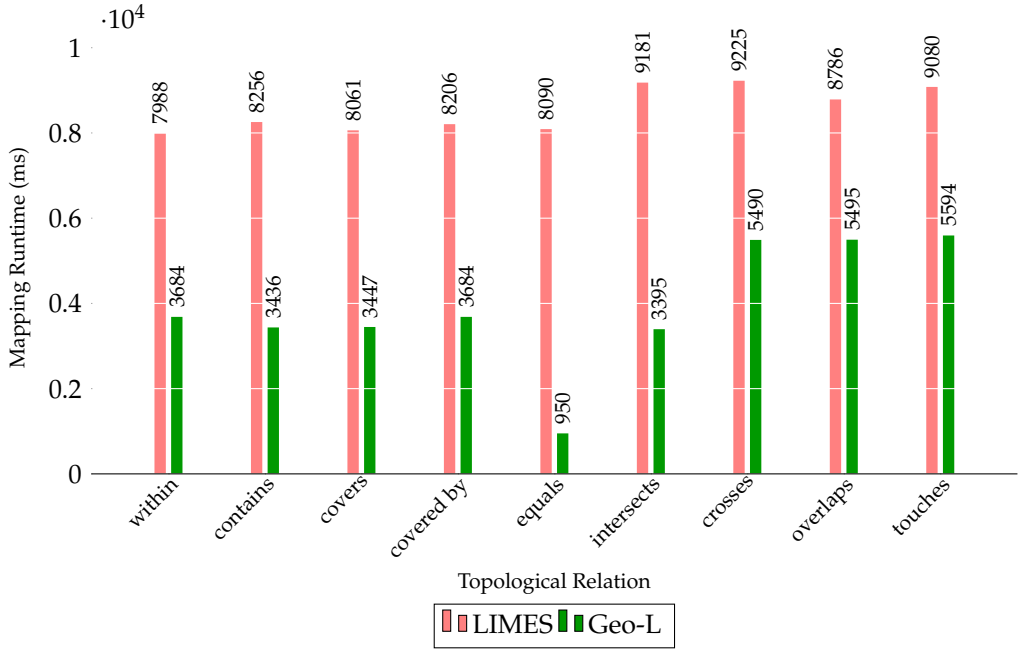



**Figure 3.** Performance OLU–OLU; size: $165 \cdot 10^3 \times 165 \cdot 10^3$.

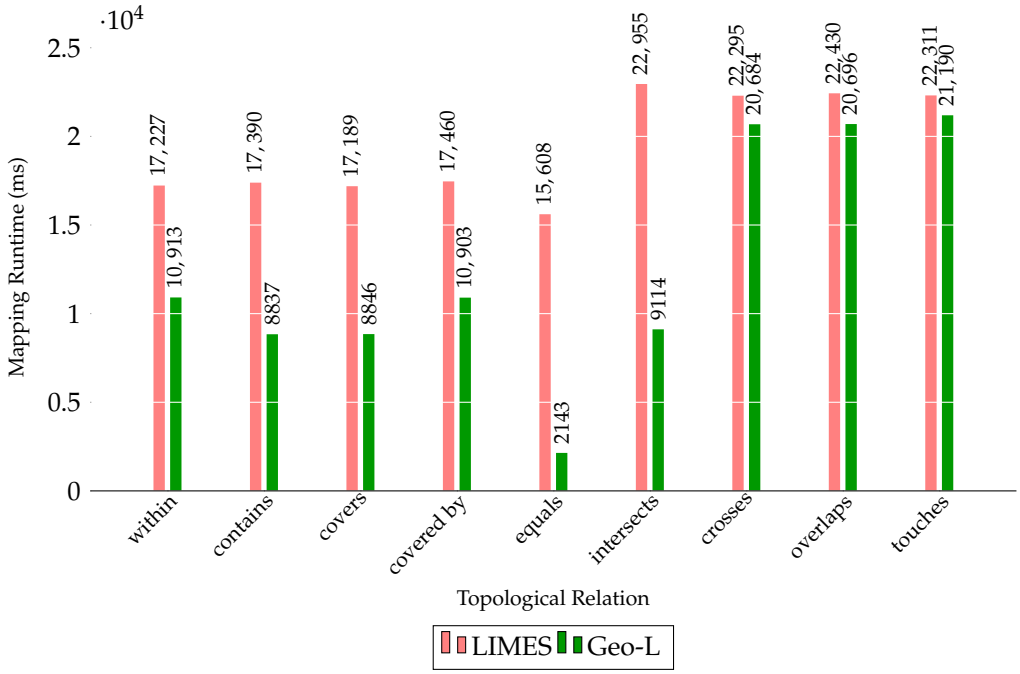



**Figure 4.** Performance OLU–OLU; size: $400 \cdot 10^3 \times 400 \cdot 10^3$.

In addition, we found discrepancies between the links discovered by each system. For example, when looking for links of entities which stood in the *within* relation in two sets with identical entities, the expected result was that each item in the source set would stand in this relation with exactly one entity of the target set, and that the size of the returned set would be equal to the size of each set. However, for the $165 \cdot 10^3$ OLU subset, Geo-L found 164,935 links, whereas LIMES found 155,083. The 65 entities which Geo-L did not include had invalid geometries, which were detected already during construction and were omitted from the search space. We examined the result computed by LIMES and noticed that the difference of 9852 consisted mostly of "false negatives" errors, i.e., valid geometries which were omitted from the result set (9849 links). Furthermore, there were three links that Geo-L did not find and LIMES did. These, however, were "false positives", i.e., the links contained invalid geometries, which were included in the result set by LIMES, whereas Geo-L already omitted them before computing the links. Similar errors also occurred for other topological relations.

#### 5.2.2. Real-World Scenarios

We experimented with the topological relation discovery between pairs of geospatial resources mentioned in Section 5.1, and compared their performance to that of LIMES. Figure 5 shows the performance, in terms of mapping runtime, on different subsets of SPOI and OLU. In this example, the largest subset did not contain the other two: the first $500 \cdot 10^3$ entities of OLU contained geometries which caused the LIMES system to crash and, therefore, we chose a subset of the same size but specified a different offset.

Figure 6 shows the running times for mapping SPOI to NUTS with different subset sizes of SPOI. Since NUTS geometries were not represented in WKT format, we used a configuration feature which defined a resource via a SPARQL query. In this case, the query also transformed the geometries into the required format. This, however, was not possible in LIMES and, therefore, the comparison of the systems was not presented.

Figure 7 shows the mapping runtime for different subsets of OLU to NUTS, for different topological relations.

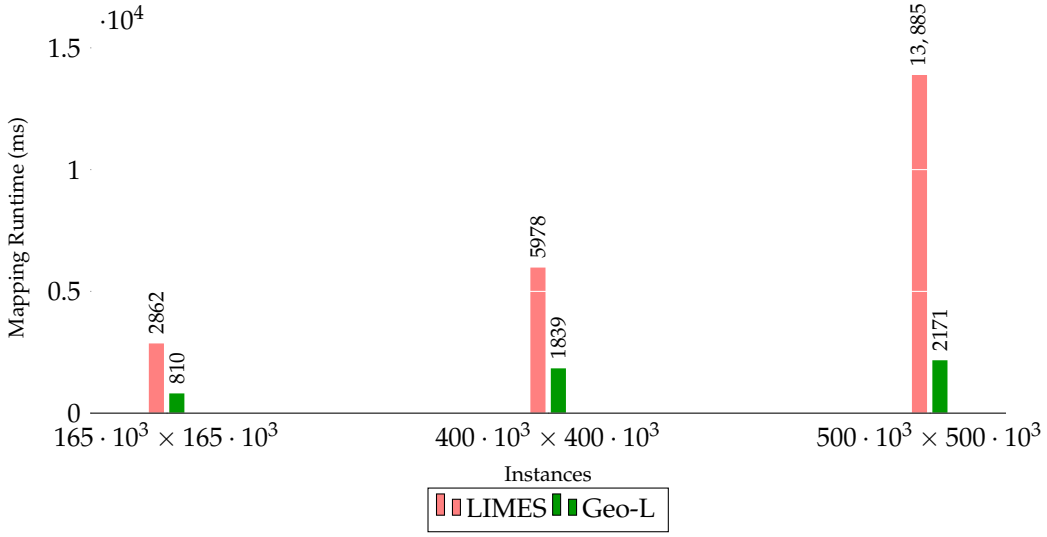



**Figure 5.** Performance SPOI–OLU; topological relation: within.

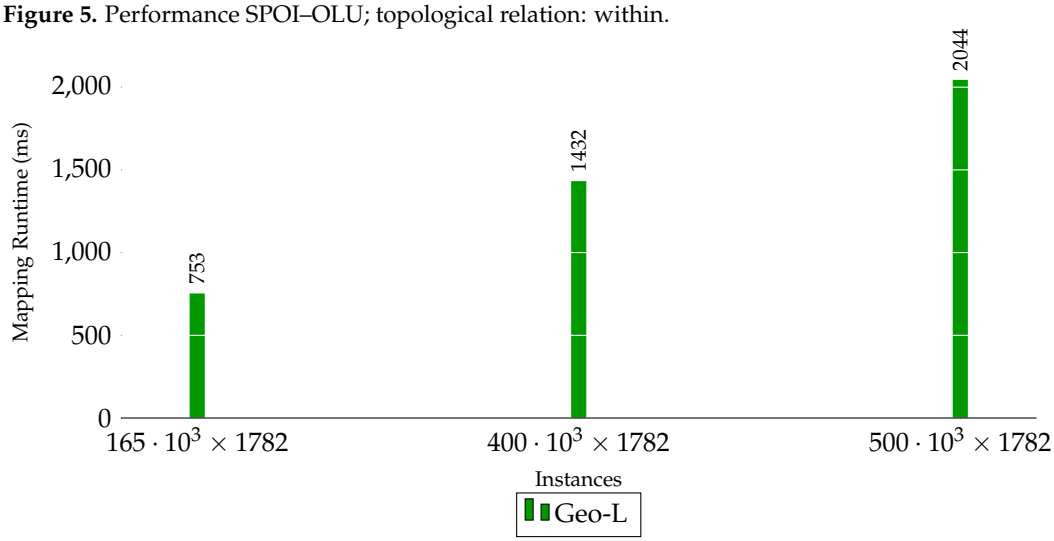



**Figure 6.** Performance SPOI–NUTS; topological relation: within.

#### 5.2.3. Practical Use Cases

The system has been employed as part of DataBio, an EU Horizon 2020 project. A major goal of the project is to show the benefits of Big Data technologies in the raw material production from agriculture for the bioeconomy industry. The project uses linked data as a federated layer to integrate cross-organizational heterogeneous data.

In particular, Geo-L has been successfully applied to various use cases in field management, as is demonstrated through the examples:

*Riparian buffer zones* are vegetated or forested strips around lakes and along water courses. Their purpose, in the context of agricultural management, is to protect water bodies from pollutants such as pesticides, nutrients, and sediment [53]. It is, therefore, crucial to detect cases where field areas and buffer zones intersect. We applied Geo-L to identify plots from the Czech registry of farmland, which intersect with buffer zones around water bodies, and Figure 8 depicts such a case where a buffer zone of a lake intersected with a field. The intersection was marked with orange.

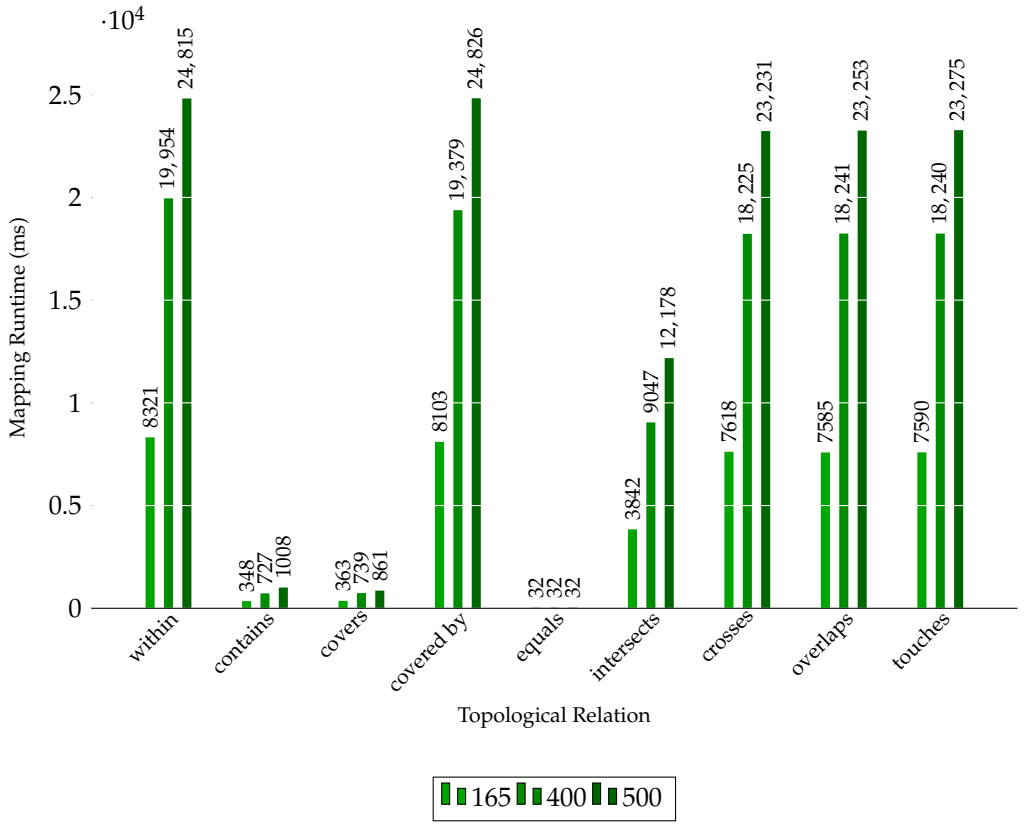



**Figure 7.** Performance OLU–NUTS; size: $X \cdot 10^3 \times 1782$.

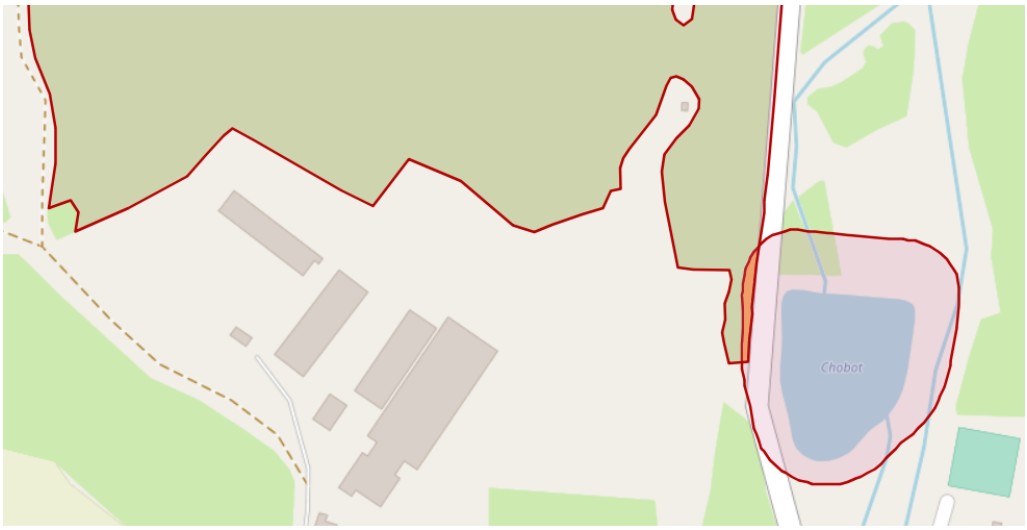


**Figure 8.** Buffer zone of a lake which intersects with a field.

*Soil erosion* is the detachment and deposition of soil particles. It may be caused by natural physical forces, e.g., wind, rainfall, ice, gravity, or due to human-induced land use [54]. As the latter results in much faster erosion rates, it can affect the soil quality dramatically due to the loss of nutrients, as well as the ability to accept and hold them. Soil erosion, therefore, impacts biological productivity and sustainability negatively and it is of high importance to control erosion zones [55,56]. We used Geo-L to identify soil erosion zones in farms; Figure 9 shows the erosion zones overlapping with a plot marked in dark blue.

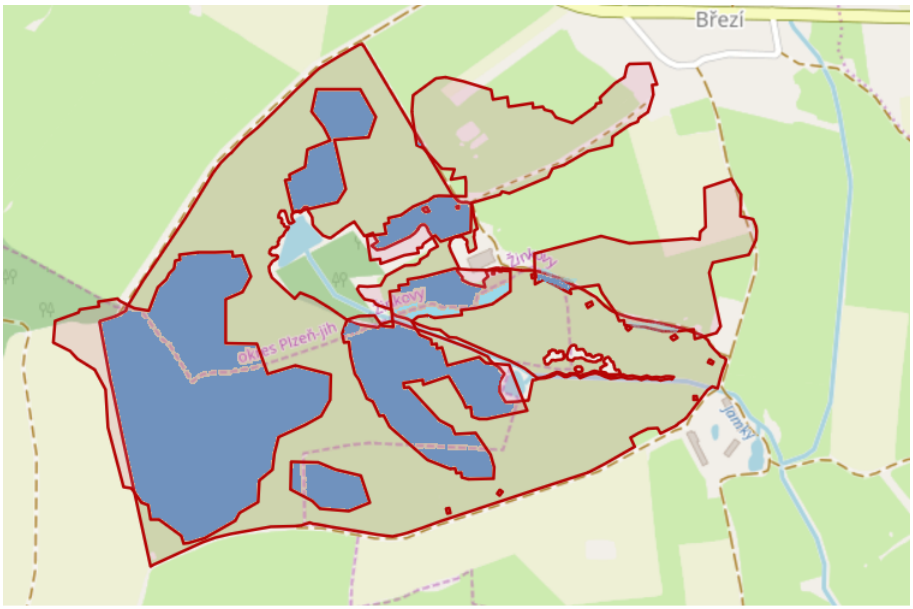



**Figure 9.** Erosion land zones of a field.

Farm management and agricultural landscape planning include, among others, practices of *crop rotation* or diversification to improve soil organic matter, maintain field productivity, and control plant diseases [57]. A method for identifying fields with the same crop type for a specific year can, thus, serve as an assisting tool for policy makers to implement and coordinate such strategies at different territorial levels. In order to support the management of crop diversity, Geo-L was used to locate fields within a specific region which grew the same type of crop as a reference field. A reference field, for example, is presented in Figure 10 marked in brown. Geo-L extracted its crop type for 2019 from the endpoint—in this case, maize for silage—and identified all other fields recorded for this use case in which maize for silage was grown in that year within the South Moravian Region (region borders marked in grey).

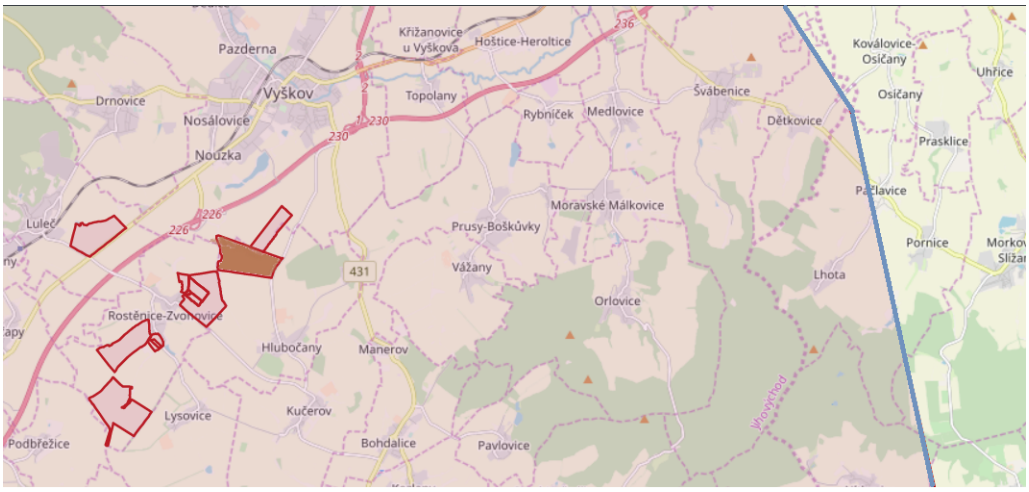



**Figure 10.** Fields which grew maize for silage during 2019 within the South Moravian Region.

## 6. Discussion

This paper presents Geo-L, a system for discovering the RDF links between geospatial entities, based on topological relations. We conducted experiments to detect topological relations between points and polygons, and between polygons and polygons. The experiments showed that Geo-L outperformed LIMES [27], a state-of-the-art link discovery system, for this task in several aspects.

- Scalability and efficiency: Geo-L configuration allowed to form a dataset directly by the SPARQL query that defined it. This feature was, in particular, useful when data at the SPARQL endpoint were stored differently than specified for the linking task, but could be transformed into the required format through SPARQL functions.LIMES, on the other hand, allowed only the detection of relations applied directly to entities of the datasets:
  - Download time: Datasets were cached not for a single task, but were regarded as resources of their own. Thanks to its caching mechanism, Geo-L accessed the SPARQL endpoints only when data required in the dataset were missing, and expanded existing datasets where possible. LIMES, on the other hand, performed a download for each dataset query; previously downloaded datasets were re-downloaded and, as a result, its operation required more time and space.
  - Mapping time: Geo-L utilized PostgreSQL with the PostGIS index for the storing and indexing of the data. This enabled efficient spatial joins between source and target datasets.
- Robustness: Geo-L included multiple features that strengthened the robustness of the application.
  - Caching: Geo-L cached portions of the data as they were downloaded, rather than writing the whole dataset after being downloaded, as LIMES did. This property prevented data loss when, e.g., connection to the remote endpoint was lost.
  - Mapping accuracy: Geo-L detected entities with invalid geometries (compliant to OGC OpenGIS "Simple Features for SQL" specifications) and did not include them in the search space. In addition, in several cases, LIMES did not include valid geometries in the result set, whereas Geo-L correctly did.
- Interoperability and flexibility: Geo-L could be used as a stand-alone application or as a REST service (in a docker), which would allow it to be integrated with other applications. The easy SPARQL-based and slim set-up of the target and source configuration (as JSON) enabled a very free usage of the tool.

Future work will examine relations between other types of geometries as well as explore geospatial relations based on various distance measures. The current implementation recalled the same items for each dataset once they were cached. In the future, we will also address re-caching to reflect the latest data on the SPRAQL endpoint, an issue which is, to the best of our knowledge, not handled by other geospatial linking systems.

**Author Contributions:** Conceptualization, Christian Zinke-Wehlmann; Investigation, Christian Zinke-Wehlmann and Amit Kirschenbaum; methodology, Christian Zinke-Wehlmann and Amit Kirschenbaum; project administration, Christian Zinke-Wehlmann; Software, Amit Kirschenbaum; validation, Christian Zinke-Wehlmann and Amit Kirschenbaum; data curation, Ami Kirschenbaum; writing—original draft, Christian Zinke-Wehlmann and Amit Kirschenbaum; writing—review and editing, Christian Zinke-Wehlmann and Amit Kirschenbaum; funding acquisition, Christian Zinke-Wehlmann. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the DataBio project, funded by the European Union's Horizon 2020 research and innovation programme under grant agreement no. 732064, as well as by the STARGATE project, funded by the European Union's Horizon 2020 research and innovation programme under grant agreement no. 818187.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The reported results are based on datasets available through the SPARQL endpoint of the FOODIE project [51]: https://www.foodie-cloud.org/sparql (accessed on 11 October 2021). These datasets are stored under the following graphs: OLU: http://w3id.org/foodie/olu# (accessed on 11 October 2021); SPOI: http://www.sdi4apps.eu/poi.rdf (accessed on 11 October 2021); NUTS: http://nuts.geovocab.org/ (accessed on 11 October 2021).

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

## References


1. W3C. *W3C Data Web Activity—Building the Web of Data*; W3C: Cambridge, MA, USA, 2014.
2. Klyne, G.; Caroll, J.J. *Resource Description Framework (RDF): Concepts and Abstract Syntax*; W3C: Cambridge, MA, USA, 2004.
3. RDF Working Group. *Resource Description Framework (RDF)*; W3C: Cambridge, MA, USA, 2014.
4. Bechhofer, S.; Van Harmelen, F.; Hendler, J.; Horrocks, I.; McGuinness, D.L.; Patel-Schneider, P.F.; Stein, L.A. OWL web ontology language reference. *W3C Recomm.* **2004**, *10*, 1–53.
5. OWL Working Group. *OWL 2 Web Ontology Language Document Overview*, 2nd ed.; W3C: Cambridge, MA, USA, 2012.
6. Prud'hommeaux, E.; Seaborne, A. SPARQL Query Language for RDF. *W3C Recomm.* **2008**. Available online: http://www.w3.org/TR/rdf-sparql-query/ (accessed on 11 October 2021).
7. Hitzler, P.; Krotzsch, M.; Rudolph, S. *Foundations of Semantic Web Technologies*; Chapman and Hall/CRC: Boca Raton, FL, USA, 2009.
8. Battle, R.; Kolas, D. Geosparql: Enabling a geospatial semantic web. *Semant. Web J.* **2011**, *3*, 355–370. [CrossRef]
9. Nikolaou, C.; Dogani, K.; Bereta, K.; Garbis, G.; Karpathiotakis, M.; Kyzirakos, K.; Koubarakis, M. Sextant: Visualizing time-evolving linked geospatial data. *J. Web Semant.* **2015**, *35*, 35–52. [CrossRef]
10. Koubarakis, M.; Bereta, K.; Papadakis, G.; Savva, D.; Stamoulis, G. Big, Linked Geospatial Data and Its Applications in Earth Observation. *IEEE Internet Comput.* **2017**, *21*, 87–91. [CrossRef]
11. Auer, S.; Lehmann, J.; Hellmann, S. Linkedgeodata: Adding a spatial dimension to the web of data. In Proceedings of the International Semantic Web Conference, Washington, DC, USA, 25–29 October 2009; pp. 731–746.
12. Čerba, O.; Charvát, K.; Mildorf, T.; Bērziņš, R.; Vlach, P.; Musilová, B. SDI4Apps Points of Interest Knowledge Base. In *Progress in Cartography*; Springer: Berlin/Heidelberg, Germany, 2016; pp. 229–237.
13. de León, A.; Saquicela, V.; Vilches, L.M.; Villazón-Terrazas, B.; Priyatna, F.; Corcho, O. Geographical linked data: A Spanish use case. In Proceedings of the 6th International Conference on Semantic Systems, Graz, Austria, 1–3 September 2010; p. 36.
14. Debruyne, C.; Clinton, É.; McNerney, L.; Nautiyal, A.; O'Sullivan, D. Serving Ireland's Geospatial Information as Linked Data. In Proceedings of the International Semantic Web Conference (Posters & Demos), Kobe, Japan, 19 October 2016.
15. Eurostat-European Commission. *Regions in the European Union. Nomenclature of Territorial Units for Statistics—NUTS 2013/EU-28*; European Union: Brussels, Belgium, 2015.
16. Bizer, C.; Heath, T.; Berners-Lee, T. Linked data: The story so far. In *Semantic Services, Interoperability and Web Applications: Emerging Concepts*; IGI Global: Hershey, PA, USA, 2011; pp. 205–227.
17. Wiemann, S.; Bernard, L. Spatial data fusion in Spatial Data Infrastructures using Linked Data. *Int. J. Geogr. Inf. Sci.* **2016**, *30*, 613–636. [CrossRef]
18. Clementini, E.; Di Felice, P.; Van Oosterom, P. A small set of formal topological relationships suitable for end-user interaction. In Proceedings of the International Symposium on Spatial Databases, Singapore, 23–25 June 1993; Springer: Berlin/Heidelberg, Germany, 1993; pp. 277–295.
19. Clementini, E.; Sharma, J.; Egenhofer, M.J. Modelling topological spatial relations: Strategies for query processing. *Comput. Graph.* **1994**, *18*, 815–822. [CrossRef]
20. Standard, International Organization for Standardization. *Geographic Information—Spatial Schema*; Standard, International Organization for Standardization: Geneva, CH, USA, 2003.
21. Strobl, C. Dimensionally Extended Nine-Intersection Model (DE-9IM). In *Encyclopedia of GIS*; Shekhar, S., Xiong, H., Eds.; Springer: Berlin/Heidelberg, Germany, 2008; pp. 240–245.
22. Freeman, H.; Shapira, R. Determining the Minimum-Area Encasing Rectangle for an Arbitrary Closed Curve. *Commun. ACM* **1975**, *18*, 409–413. [CrossRef]
23. Smeros, P.; Koubarakis, M. Discovering Spatial and Temporal Links among RDF Data. In Proceedings of the Workshop on Linked Data on the Web (LDOW 2016), Montreal, QC, Canada, 12 April 2016; Volume 1593.
24. Isele, R.; Jentzsch, A.; Bizer, C. Efficient Multidimensional Blocking for Link Discovery without losing Recall. In Proceedings of the Fourteenth International Workshop on Web and Databases (WebDB 2011), Athens, Greece, 12 June 2011.
25. Volz, J.; Bizer, C.; Gaedke, M.; Kobilarov, G. Silk—A Link Discovery Framework for the Web of Data. In Proceedings of the Workshop on Linked Data on the Web (LDOW 2009), Madrid, Spain, 20 April 2009; Volume 538.
26. Sherif, M.A.; Dreßler, K.; Smeros, P.; Ngomo, A.N. Radon—Rapid Discovery of Topological Relations. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, San Francisco, CA, USA, 4–9 February 2017; pp. 175–181.
27. Ngomo, A.C.N.; Auer, S. LIMES—A Time-Efficient Approach for Large-Scale Link Discovery on the Web of Data. In Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, Barcelona, Spain, 16–22 July 2011; pp. 2312–2317.
28. DICE Group. LIMES—Link Discovery Framework for Metric Spaces: v1.5.5. 2019. Available online: https://github.com/dice-group/LIMES/releases/tag/1.5.5 (accessed on 11 October 2021).
29. Faria, D.; Balasubramani, B.S.; Shivaprabhu, V.R.; Mott, I.; Pesquita, C.; Couto, F.M.; Cruz, I.F. Results of AML in OAEI 2017. In Proceedings of the Twelfth International Workshop on Ontology Matching, Vienna, Austria, 21 October 2017; pp. 122–128.

30. Faria, D.; Pesquita, C.; Santos, E.; Cruz, I.F.; Couto, F.M. AgreementMakerLight: A scalable automated ontology matching system. In Proceedings of the 10th International Conference on Data Integration in the Life Sciences (DILS 2014), Lisbon, Portugal, 17–18 July 2014; pp. 29–32.
31. DICE Group. Geometry API for Java: v2.2.3. 2019. Available online: https://github.com/Esri/geometry-api-java/releases/tag/v2.2.3 (accessed on 11 October 2021)
32. Khiat, A.; Mackeprang, M. I-Match and OntoIdea results for OAEI 2017. In Proceedings of the Twelfth International Workshop on Ontology Matching, Vienna, Austria, 21 October 2017; pp. 135–137.
33. Achichi, M.; Cheatham, M.; Dragisic, Z.; Euzenat, J.; Faria, D.; Ferrara, A.; Flouris, G.; Fundulaki, I.; Harrow, I.; Ivanova, V.; et al. Results of the Ontology Alignment Evaluation Initiative 2017. In Proceedings of the Twelfth International Workshop on Ontology Matching, Vienna, Austria, 21 October 2017; pp. 61–113.
34. TomTom. Available online: https://www.tomtom.com (accessed on 11 October 2021).
35. Saveta, T.; Fundulaki, I.; Flouris, G.; Ngomo, A.N. SPgen: A Benchmark Generator for Spatial Link Discovery Tools. In Proceedings of the 17th International Semantic Web Conference (ISWC2018), Part I, Monterey, CA, USA, 8–12 October 2018; Volume 11136, pp. 408–423.
36. Doudali, T.D.; Konstantinou, I.; Koziris, N. Spaten: A Spatio-temporal and Textual Big Data Generator. In Proceedings of the 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, USA, 11–14 December 2017; pp. 3416–3421.
37. Kyzirakos, K.; Karpathiotakis, M.; Koubarakis, M. Strabon: A Semantic Geospatial DBMS. In Proceedings of the 11th International Semantic Web Conference (ISWC 2012), Boston, MA, USA, 11–15 November 2012; Volume 7649, pp. 295–311.
38. European Commission. *CORINE Land Cover Project—Technical Guide*; Office for Official Publications of the European Communities: Luxembourg, 1994.
39. Lee, J.G.; Kang, M. Geospatial Big Data: Challenges and Opportunities. *Big Data Res.* **2015**, *2*, 74–81. [CrossRef]
40. Li, S.; Dragicevic, S.; Castro, F.A.; Sester, M.; Winter, S.; Coltekin, A.; Pettit, C.; Jiang, B.; Haworth, J.; Stein, A.; et al. Geospatial Big Data Handling Theory and Methods: A Review and Research Challenges. *ISPRS J. Photogramm. Remote. Sens.* **2016**, *115*, 119–133. [CrossRef]
41. Guttman, A. R-Trees: A Dynamic Index Structure for Spatial Searching. In Proceedings of the 1984 ACM SIGMOD Conference, Boston, MA, USA, 18–21 June 1984; pp. 47–57.
42. Brinkhoff, T.; Kriegel, H.; Seeger, B. Efficient Processing of Spatial Joins Using R-Trees. In Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, DC, USA, 25–28 May 1993; pp. 237–246.
43. Jordahl, K.; den Bossche, J.V. Geopandas/Geopandas: v0.4.0. 2018. Available online: https://github.com/geopandas/geopandas/tree/v0.4.0 (accessed on 11 October 2021).
44. Beckmann, N.; Kriegel, H.P.; Schneider, R.; Seeger, B. The R*-tree: An efficient and robust access method for points and rectangles. In Proceedings of the 1990 ACM SIGMOD Internatioanl Conference on Management of Data (SIGMOD'90), Atlantic City, NJ, USA, 23–26 May 1990; pp. 322–331.
45. Behnel, S.; Bradshaw, R.; Citro, C.; Dalcin, L.; Seljebotn, D.S.; Smith, K. Cython: The Best of Both Worlds. *Comput. Sci. Eng.* **2011**, *13*, 31–39. [CrossRef]
46. The PostGIS Development Group. *PostGIS 3.1.5dev Manual*; The PostGIS Development Group: Beaverton, OR, USA, 2021.
47. Refractions Research Inc. *PostGIS 2.5.0 Manual*; Refractions Research Inc.: Victoria, BC, Canada, 2018.
48. Hellerstein, J.M.; Naughton, J.F.; Pfeffer, A. Generalized search trees for database systems. In Proceedings of the 21st International Conference on Very Large Data Bases (VLDB'95), Zurich, Switzerland, 11–15 September 1995; pp. 562–573.
49. Cerba, O.; Mildorf, T. Smart Points of Interest: Big, Linked and Harmonized Spatial Data. In Proceedings of the 19th International Research Symposium on Computer-Based Cartography (AutoCarto 2016), Albuquerque, NM, USA, 14–16 September 2016; pp. 4–13.
50. Mildorf, T.; Charvát, K.; Ježek, J.; Templer, S.; Malewski, C. Open Land Use Map. *AGRIS On-Line Pap. Econ. Inform.* **2014**, *6*, 81–88.
51. Charvát, K.; Esbri, M.A.; Mayer, W.; Charvát, K., Jr.; Campos, A.; Palma, R.; Krivanek, Z. FOODIE—Open data for agriculture. In Proceedings of the IST-Africa 2014 Conference Proceedings, Pointe aux Piments, Mauritius, 7–9 May 2014; pp. 1–9.
52. Norton, B.; Vilches, L.M.; De Léon, A.; Goodwin, J.; Stadler, C.; Anand, S.; Harries, D.; Villazón-Terrazas, B.; Atemezing G.A. *NeoGeo Vocabulary Specification–Madrid Edition*; Martín Salas, J., Harth, A., Eds.; Public Draft. 2012. Available online: http://geovocab.org/doc/neogeo/ (accessed on 11 October 2021).
53. Zhang, X.; Liu, X.; Zhang, M.; Dahlgren, R.A.; Eitzel, M. A Review of Vegetated Buffers and a Meta-analysis of Their Mitigation Efficacy in Reducing Nonpoint Source Pollution. *J. Environ. Qual.* **2010**, *39*, 76–84. [CrossRef]
54. Vanwalleghem, T. Soil Erosion and Conservation. In *International Encyclopedia of Geography: People, the Earth, Environment and Technology*; Wiley Online Library: Hoboken, NJ, USA, 2016; pp. 1–10.
55. Larson, W.E.; Pierce, F.J.; Dowdy, R.H. The threat of soil erosion to long-term crop production. *Science* **1983**, *219*, 458–465. [CrossRef] [PubMed]
56. Blanco-Canqui, H.; Lal, R. Erosion Control and Soil Quality. In *Principles of Soil Conservation and Management*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 477–492.
57. Curl, E.A. Control of plant diseases by crop rotation. *Bot. Rev.* **1963**, *29*, 413–479. [CrossRef]