Discovering Spatial and Temporal Links among RDF Data

Panayiotis Smeros[*]
EPFL
panayiotis.smeros@epfl.ch

Manolis Koubarakis
National and Kapodistrian University of Athens
koubarak@di.uoa.gr

ABSTRACT

Link Discovery is a new research area of the Semantic Web which studies the problem of finding semantically related entities lying in different knowledge bases. This area has become more crucial recently, as the volume of the available Linked Data on the web has been increasing considerably. Although many link discovery tools have been developed, none of them takes into consideration the discovery of spatial or temporal relations, leaving datasets with such characteristics weakly interlinked and therefore disallowing the exploitation of the rich information they provide.

In this paper, we propose new methods for Spatial and Temporal Link Discovery and provide the first implementation of our techniques based on the well-known framework Silk. Silk, enhanced with the new features, allows data publishers to generate a wide variety of spatial, temporal and spatiotemporal relations between their data and other Linked Open Data, dealing effectively with the common heterogeneity issues of such data. Furthermore, we experimentally evaluate our implementation by using it in a real-world scenario and demonstrate that it discovers accurately all the existing links in a time efficient and scalable way.

CCS Concepts

•**Information systems → Information integration; Spatial-temporal systems; Data extraction and integration;**

Keywords

Spatial and Temporal Link Discovery, Semantic Web, Linked Data

1. INTRODUCTION

Linked data is a research area which studies how one can make RDF data available on the Web, and interlink it with other data in order to increase its value for users [4]. The goal of Linked Data is to allow people to share structured data on the web as easily as they can do with documents today. An important step for the evolution of the Web of documents to the Web of data is the transformation of the data from any form that it exists into a common format, the Resource Description Framework (RDF) so that it can be easily integrated with other data already transformed in this format. All the data that is compatible with the Linked Data Principles composes the Linked Open Data (LOD) cloud[1].

Recently, spatial and temporal extensions to RDF have been proposed and implemented. GeoSPARQL [22] is a recent OGC[2] standard that allows representing and querying geospatial data on the Semantic Web. Also, the data model stRDF accompanied by the query language stSPARQL [17] are extensions of the standard RDF and SPARQL for representing and querying geospatial data that changes over time. Both of the above extensions are implemented in the open source spatiotemporal RDF store Strabon [18]. Furthermore, OGC and W3C have established a joined working group which studies the use of spatial data on the web [29].

Link Discovery is the fourth Linked Data Principle. Its main objective is to establish semantic links between entities in order to enhance and enrich the information that is known about them. Whilst the problem of Entity Resolution i.e., the problem of finding entities which are equivalent, has been studied a lot in areas such as Relational Databases and Information Retrieval, Link Discovery defines the more generic problem of finding semantically related entities lying in different knowledge bases [2].

Although a lot of effort has been given in the representation and querying of geospatial and temporal RDF data, there are not many works in the respective area of Link Discovery. In the context of Spatial Link Discovery, state of the art techniques are focusing on finding only spatial equivalences between entities, leaving other kinds of relations e.g., topological relations, undiscovered and the rich geospatial information lying in many datasets unexploited. The situation regarding Temporal Link Discovery is even more premature and thus, to the best of our knowledge, there are almost no datasets with temporal information that are connected with each other with links that denote a temporal relation.

Another common use of Link Discovery is for detecting internal links within a single dataset. For example, by applying an Entity Resolution method on a dataset, we can

[*] Work done while the author was at National and Kapodistrian University of Athens.

[1] http://www.w3.org/DesignIssues/LinkedData.html

[2] http://www.opengeospatial.org

WWW2016 Workshop: Linked Data on the Web (LDOW2016). April 11, 2016 - Montréal, Canada.

discover all the similar entities i.e., all the duplicates of this dataset. Hence, with Spatial and Temporal Link Discovery we can materialize all the spatial, temporal and spatiotemporal relations that hold between the entities of a dataset. This operation is very useful in the areas of Qualitative Spatial and Temporal Reasoning where the large graphs that are created based on the qualitative relations are given as input to corresponding reasoners [7, 10] in order to extract useful information or to verify the consistency of a dataset.

The lack of research in the area of Spatial and Temporal Link Discovery will be made more notable when more datasets with such characteristics will be made available in the LOD Cloud. Currently, a lot of initiatives are moving towards this direction by publishing open geospatial data and metadata coming out of open government directives[3] and open Earth Observation (EO) data and metadata that is currently made available by space and environment agencies (e.g., ESA, NASA and EEA)[4]. This data usually consists of measurements produced by observations with hundreds of gigabytes of geospatial and temporal information. Making all this data available as Linked Data and interlinking it with semantic connections will allow the development of services with great environmental and commercial value.

EU projects such as LEO, MELODIES and TELEIOS, have already started exploiting this kind of data by studying its whole life cycle [15]. Their final goal is to build big knowledge bases with data from various sources and be able to perform queries efficiently on them. Use cases from these projects have shown that combining heterogeneous data, especially in its spatial dimension, can be a bottleneck to this procedure, since they significantly increase the execution time[5]. This happens due to the fact that most of the spatial join operations (e.g., check of containment or non-intersection) have by definition quadratic complexity[6]. This complexity can get even higher when we first have to homogenize the joining data. Thus, computing such relations among very complex or heterogeneous data, on query time, can be extremely time consuming and prohibitory for real-time applications. Therefore, there is a need for a framework that will be able to materialize those relations (links) efficiently, even for highly demanding workloads.

In this paper, we propose new methods for Spatial and Temporal Link Discovery and provide the first implementation of our techniques based on the well-known framework Silk. Silk, enhanced with the new features, allows data publishers to generate a wide variety of spatial, temporal and spatiotemporal relations between their data and other Linked Open Data, dealing effectively with the common heterogeneity issues of such data. Furthermore, we experimentally evaluate our implementation by using it in a real-world scenario, with datasets that comprise rich spatial and temporal information, and demonstrate that it discovers accurately all the existing links in a time efficient and scalable way.

The structure of the paper is organized as follows. In Section 2 we present related work in the area of Link Discovery. In Section 3 we provide the background on which the new methods for Spatial and Temporal Link Discovery that we propose in Section 4 are based. In Section 5 we describe the implementation of our techniques in the framework Silk and in Section 6 we present the experimental evaluation of them. Finally, in Section 7 we conclude the work by discussing future directions.

2. RELATED WORK

Up to now, little effort has been given in the research area of Spatial and Temporal Link Discovery. Most of the approaches on generic Link Discovery do not exploit the rich spatial and temporal information existing in some datasets, whereas domain specific approaches on Spatial Link Discovery are able to discover only spatial similarities (spatial duplicates). Hence, to the best of our knowledge, there are no frameworks, either generic or domain specific, for discovering spatial or temporal relations other than equivalences among RDF datasets. Below we describe the most related to our work state-of-the-art Link Discovery frameworks.

In the area of generic Link Discovery, the authors of [12] propose the declarative link specification language LinQL, which is translated to standard SQL by the framework LinQuer, for discovering semantic links over relational data.

The LIMES framework [20] introduces a generic algorithm for Link Discovery which reduces the number of comparisons that are needed during the interlinking phase by utilizing the triangle inequality in metric spaces. For finding link specifications, LIMES implements supervised and unsupervised machine learning algorithms.

Similarly to LIMES, Silk [13] is also a generic framework for discovering relationships between data items within different Linked Data sources. Silk, which is the only open source generic Link Discovery framework, features a declarative link specification language for specifying which types of RDF links should be discovered between data sources as well as which conditions entities must fulfill in order to be interlinked. These linkage rules may combine various metrics and can take the graph around entities into account, which is addressed using an RDF path language. Silk accesses the data sources that should be interlinked via the SPARQL protocol and can thus be used against local as well as remote SPARQL endpoints.

In the area of Spatial Link Discovery there are some domain specific approaches which are able to discover only spatial equivalences among datasets i.e., they focus on Spatial Entity Resolution. In order to achieve this, they combine the geographic distance of the geometries of the entities with other kinds of distances e.g., with the string distance of their labels. The supported geographic distances can be applied either between any kind of spatial objects (e.g., Hausdorff distance), or only between point objects (e.g., Orthodromic distance). Some of these approaches are presented in [25, 26, 30].

From the generic frameworks, LIMES also addresses the problem of Spatial Entity Resolution in [21]. The computation of the distance between the spatial objects lies on a combination of Hausdorff and Orthodromic metrics. On the other hand, Silk supports the geographic distance only between point objects.

A detailed review on state of the art algorithms and frameworks is performed in the surveys [2, 19] as well as in [28].

[3] http://www.linkedopendata.gr

[4] http://datahub.io/organization/teleios and http://datahub.io/organization/leo

[5] http://www.melodiesproject.eu/content/enhancing-geospatial-sparql-query-times-silk

[6] $O(nm)$ where n and m are the numbers of points of the joining spatial objects.

3. BACKGROUND

In this section we provide the state-of-the-art on the representation of spatial and temporal information in the RDF data model and a formal definition of the problem of Link Discovery. We also present the models and calculi on which we base the new Link Discovery relations that we introduce. The extended background of this paper is given in [28].

3.1 Representation of Spatial and Temporal Information in the RDF data model

Spatial information in the RDF data model is usually represented as serializations of geometries accompanied with a Coordinate Reference System (CRS) which defines how to relate these serializations to real geometries on the surface of Earth. For encoding this information Well-Known Text (WKT) and Geography Markup Language (GML) are used. WKT is an OGC standard for the representation of vector geometry objects, CRSs, and transformation rules between different CRSs. On the other hand, GML, developed by OGC as well, is the most common XML-based encoding standard for the representation of geospatial data, that provides XML schemas for defining a variety of concepts that are of use in Geography: geographic features, geometries, CRSs and topologies.

The main RDF vocabularies for representing spatial as well as temporal information are described below.

W3C GEO. W3C GEO is an RDF vocabulary for representing simple location information in RDF. It provides the basic terminology for serializing point geometries using a namespace for representing latitude, longitude and other information about spatially-located things. The CRS of this vocabulary is encoded in the namespace and it is the WGS 84[7]. Below, we give an example of the W3C GEO vocabulary:

```
 _:1 rdf:type wgs84geo:Point .
 _:1 wgs84geo:lat "10"^^xsd:double.
 _:1 wgs84geo:long "20"^^xsd:double.
```

GeoSPARQL. GeoSPARQL [22] is a recent OGC standard that allows representing and querying geospatial data on the Semantic Web. It defines a vocabulary for representing geospatial data in RDF, and an extension to the SPARQL query language for processing geospatial data. It uses literal values to encode geometries and introduces two RDF datatypes, the `geo:wktLiteral` and `geo:gmlLiteral`, for the WKT and GML literals. An example of a geometry in GeoSPARQL is given below:

```
_:1 rdf:type geo:Geometry .
_:1 geo:hasGeometry
"<epsg:4326> POINT(10 20)"^^geo:wktLiteral .
```

stRDF: The Spatial Dimension. The data model stRDF [17] is an extension of the standard RDF for representing geospatial data. Similarly to GeoSPARQL, stRDF uses the OGC standards WKT and GML for the representation of geospatial data and introduces two new literal datatypes, the `stdf:WKT` and `strdf:GML`. An example of a geometry in stRDF is shown below:

```
_:1 rdf:type strdf:Geometry .
_:1 strdf:hasGeometry
"POINT(10 20);<epsg:4326>"^^strdf:WKT .
```

stRDF: The Temporal Dimension. An approach for the representation of temporal information in RDF was introduced with the temporal dimension of stRDF [3]. This approach assumes a discrete time line and uses the value space of the datatype `xsd:dateTime` of XML-Schema[8] to model time. Two kinds of time primitives are supported: time instants and time periods. Time instants are represented by literals of the `xsd:dateTime` datatype and time periods by literals of the datatype `strdf:period`. These literals are used as objects of triples to represent *user-defined time* and as *valid time* of *temporal triples*. A *temporal triple* is an expression of the form `(s, p, o, t)` where `(s, p, o)` is an RDF triple and `t` is the *valid time* of this triple. An example of a *temporal triple* in stRDF is shown below:

```
_:1 strdf:hasGeometry
"POINT(10 20);<epsg:4326>"^^strdf:WKT
"2000-01-01T00:00:00"^^xsd:dateTime .
```

3.2 Definition of Link Discovery

The definition of Link Discovery based on which we define our new methods is described as follows:

Definition 1. *Let S and T be two sets of entities and R the set of relations that can be discovered between entities. For a relation $r \in R$, w.l.o.g., a distance function d_r and a distance threshold θ_{d_r} are defined as follows:*

$$d_r : S \times T \to [0, 1] \ , \ \theta_{d_r} \in [0, 1]$$

The domain of d_r is the Cartesian product of S and T and the range is the computed distance normalized to the interval $[0, 1]$. θ_{d_r} is also normalized to $[0, 1]$.

The set of discovered links for relation r (DL_r) is defined as follows:

$$DL_r = \{(s, r, t) \mid s \in S \ \wedge \ t \in T \ \wedge \ d_r(s, t) \leq \theta_{d_r}\}$$

*DL_r contains triples which have as **subject** an entity from dataset S, as **object** an entity from dataset T and as **predicate** the relation r. A triple belongs to DL_r **iff** the function d_r returns a distance that does not exceed the threshold θ_{d_r}.*

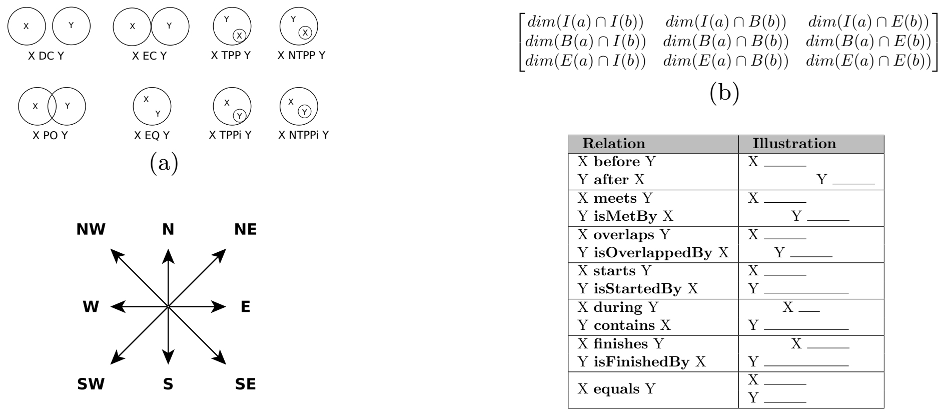
3.3 Relation Models and Calculi

Region Connection Calculus. The Region Connection Calculus (RCC) [24] is a formalization that provides a sound and complete set of topological relations between two spatial regions. RCC-8, which is a well-known subset of RCC, is based on eight topological relations (Figure 1(a)) where DC stands for DisConnected, EC for Externally Connected, TPP for Tangential Proper Part, NTPP for Non Tangential Proper Part, and TPPi and NTPPi are the inverse relations of TPP and NTPP.

Dimensionally Extended 9-Intersection Model. The Dimensionally Extended 9-Intersection Model (DE-9IM) [5] is a well-known model for representing topological relations between geometries. More specifically, this model captures topological relations in $\mathbb{R}^2$, by considering the dimension (*dim*) of the intersections involving the interior (I), the boundary (B) and the exterior (E) of the two geometries. Given the intersection matrix (Figure 1(b)), for any two spatial objects that can be points, lines and/or polygonal areas, we can define relations derived from DE-9IM such as:

[7] http://spatialreference.org/ref/epsg/wgs-84/

[8] http://www.w3.org/TR/xmlschema-2#dateTime



(a)

$$\begin{bmatrix} dim(I(a)\cap I(b)) & dim(I(a)\cap B(b)) & dim(I(a)\cap E(b)) \\ dim(B(a)\cap I(b)) & dim(B(a)\cap B(b)) & dim(B(a)\cap E(b)) \\ dim(E(a)\cap I(b)) & dim(E(a)\cap B(b)) & dim(E(a)\cap E(b)) \end{bmatrix}$$

(b)

Relation	Illustration
X **before** Y Y **after** X	X ____ Y ____
X **meets** Y Y **isMetBy** X	X ____ Y ____
X **overlaps** Y Y **isOverlappedBy** X	X ____ Y ____
X **starts** Y Y **isStartedBy** X	X ____ Y ________
X **during** Y Y **contains** X	X __ Y ________
X **finishes** Y Y **isFinishedBy** X	X ____ Y ________
X **equals** Y	X ____ Y ____

(c)

(d)

Figure 1: (a) RCC-8 Relations, (b) DE-9IM Intersection Matrix, (c) Cardinal Direction Relations and (d) Allen's Relations

Intersects, *Overlaps*, *Equals*, *Touches*, *Disjoint*, *Contains*, *Crosses*, *Covers*, *CoveredBy* and *Within*.

Egenhofer's and OGC Simple Features Model. The Egenhofer's Model [6] and the Simple Features Model proposed by OGC[9] contain different subsets of the topological relations that derive from the DE-9IM mentioned above.

Cardinal Direction Calculus. This calculus concentrates on cardinal direction relations [11, 27] which are used to describe how regions of space are placed relative to one another e.g., *region a* `is north of` *region b* (Figure 1(c)).

Allen's Interval Calculus. A widely used algebra for temporal reasoning is Allen's Interval Calculus [1], which provides the definition of possible relations between time periods. It is based on the thirteen jointly exclusive and pairwise disjoint qualitative relations given in Figure 1(d). From these basic relations, one can build new ones by taking disjunctions of them.

Spatiotemporal Constraint Calculus. By pairing a spatial and a temporal relation model or calculus we can create a spatiotemporal one. For example, the Spatiotemporal Constraint Calculus (STCC) [9] combines RCC-8 with Allen's Interval Calculus. This calculus is useful when we want to discover relations between spatial objects that change over time (as we will see in Section 6) or between moving objects.

4. METHODS FOR SPATIAL AND TEMPORAL LINK DISCOVERY

In this section we describe the new methods that we introduce for Spatial and Temporal Link Discovery. Specifically, we present the new sets of relations and transformations that we provide and an optimization technique called *Blocking*. Finally, we prove theoretically the soundness and completeness of our methods and describe how the latter has as result their accuracy to be 100%.

4.1 Spatial and Temporal Relations

According to the definition of Link Discovery, presented above, set R contains all the relations that can be discovered between entities. In this paper we introduce the sets of spatial (R_s), temporal (R_t) and spatiotemporal (R_{st}) relations. These sets contain all the relations that are included in the models and calculi described in Section 3. In more detail, R_s contains the relations that are included in the DE-9IM, Egenhofer's and OGC Simple Features Models and the Region Connection and Cardinal Direction Calculi, R_t contains the relations included in the Allen's Interval Calculus and R_{st} contains all the per two combinations of the above models and calculi.

We consider these relations as Boolean relations (R_B) i.e., either they hold or they do not hold ($R_s, R_t, R_{st} \subset R_B \subset R$). These relations have also been studied in the context of fuzzy logics but this is out of the scope of this paper.

R_B constitutes a special subset of R. The distance function d_r and the distance threshold θ_{d_r} for a relation $r \in R_B$ are defined as follows:

$$d_r(s,t) = \begin{cases} 0 & \text{if r holds} \\ 1 & \text{elsewhere} \end{cases}, \; \theta_{d_r} = 0$$

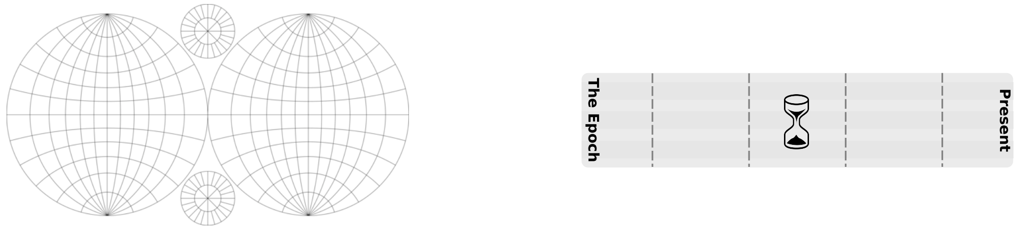
d_r returns 0 if r holds between two entities (e.g., relation *intersects* holds between two geometries or time periods) and 1 elsewhere. Hence, since d_r is a Boolean function, θ_{d_r} as a parameter is needless, thus, in order to be compliant with the definition, we can set it constantly to 0.

4.2 Spatial and Temporal Transformations

A common issue that occurs when one tries to discover links between datasets created by different data providers is *heterogeneity* (e.g., we can have datasets expressed in different time-zones). A preprocessing technique that addresses this problem is the application of transformations on the attributes of the entities before checking the existence of a link. The transformations that we introduce are generic and not tightly coupled to specific kind of datasets.

Spatial Transformations. The spatial transformations that we introduce are the following:

[9] http://www.opengeospatial.org/standards/sfs



(a) (b)

Figure 2: Blocking technique for (a) Spatial and (b) Temporal Relations

- *Vocabulary Transformation.* As mentioned in Section 3, the geometries of a dataset can be expressed in different vocabularies (e.g., W3C GEO, GeoSPARQL or stRDF). This transformation converts the geometry literals of a dataset in order to be expressed in a common vocabulary (GeoSPARQL).

- *Serialization Transformation.* The geometries of a dataset can be also serialized in different ways (e.g., using WKT or GML). This transformation converts the geometries of a dataset to follow a common serialization (WKT).

- *CRS Transformation.* Although a CRS such as WGS 84 is a comprehensive way to describe locations on Earth, some applications work on a projection of the Earth. In these cases, a projected CRS is used that transforms the 3-dimensional ellipsoid approximation of the Earth into a 2-dimensional surface. This transformation reverts the CRS of a geometry from a projected one (e.g., the GGRS87 for Greece) to the World Geodetic System (WGS 84) for better precision and homogeneity.

- *Validation Transformation.* This transformation converts not valid geometries (e.g., self-intersecting polygons) to valid ones.

- *Simplification Transformation.* Some datasets have very complex geometries, which makes the computation of spatial relations inefficient. This transformation simplifies a geometry according to a given distance tolerance, ensuring that the result is a valid geometry having the same dimension and number of components as the input.

- *Envelope Transformation.* This transformation computes the envelope (i.e., the minimum bounding rectangle) that contains a geometry and it is useful in cases that we want to compute approximate spatial relations between two datasets.

- *Area Transformation.* In some cases we may want to compare the areas of two geometries to verify the existence of a relation. This transformation computes the area of a given geometry in square metres.

- *Points-To-Centroid Transformation.* In crowdsourcing datasets like OpenStreetMap[10], multiple users can define the position of the same placemark. As a better approximation of the real position of this placemark we can compute the centroid of these positions. This transformation computes the centroid of a cluster of points.

Temporal Transformations. The respective temporal transformations that we introduce are the following:

- *Period Transformation.* The stRDF data model supports both time instants and periods. This transformation converts a time instant to a time period with the same starting and ending point.

- *Time-Zone Transformation.* Time elements (i.e., time instants and periods) can be expressed in different time-zones. This transformation converts the time-zone of a given time element to the Coordinated Universal Time (UTC).

4.3 Blocking Technique

Since the size of datasets with spatial or temporal information can be very big, approaches that perform exhaustive checks between datasets are considered inefficient. Thus, there is need for a scalable, yet sound and complete technique for decreasing the number of checks by dismissing definitive non-matches prior to the actual check. The most well-known technique to achieve this is known as *Blocking* [14, 23]. *Blocking* is an optimization technique that effectively reduces (as we will see in Section 6) the execution time of our methods without affecting (as we prove below) the accuracy of the discovered links.

Spatial Relations. The *Blocking* technique for spatial relations that we propose builds blocks that divide the earth into curved rectangles as depicted in Figure 2(a). The reference system of our technique is WGS 84 which considers a spheroid approximation of the Earth. This approximation requires complicated mathematics for calculations such as areas, distances, etc. (e.g., the shortest path between two points in the spheroid is a circle arc whereas, in a planar approximation it would be just a straight line). Nevertheless, we chose WGS 84 because we prefer having highly accurate calculations rather than fast ones.

The area of the created blocks is measured in square degrees and can be adjusted by a spatial blocking factor sbf. The formula for the computation of the area is the following:

$$blockArea = \frac{1}{sbf^2}\ {}^{\circ 2}$$

[10] http://www.openstreetmap.org

The bigger the value sbf gets, the more and smaller blocks will be created. For example, if we assign to sbf the value 10, our *Blocking* technique will create $6,480,000$ non-overlapping blocks with area 0.01°^2} that cover the whole surface of the earth[11].

After the division of the space into blocks, we compute the set of blocks into which each geometry must be inserted. In order to achieve this, we first compute the minimum bounding box (MBB) that contains each geometry, and then we find the blocks that this MBB intersects with. Thus, each geometry is assigned to all the blocks with which its MBB intersects.

Temporal Relations. In the *Blocking* technique for temporal relations we follow a similar approach to the one for spatial relations. Time is one-dimensional and thus the blocks are one-dimensional as well. Let us suppose that all the time elements of our data are included in the interval from *the Epoch*[12] until the *Present* (Figure 2(b)). This interval can be manually configured according to the time range of our data. Following the same strategy as before, we divide the time in blocks whose length can be adjusted with a temporal blocking factor tbf. The formula for the computation of the length of the blocks is the following:

$$blockLength = \frac{1}{tbf} \text{ time units}$$

After the division of the time into blocks, we insert in each block all the time periods or instants that temporally intersect with it.

Spatiotemporal Relations. The *Blocking* technique for spatiotemporal relations is a combination of the techniques mentioned above. In this case the blocks are three-dimensional (two dimensions for space and one for time). The formula for the computation of the volume of the blocks is the following:

$$blockVolume = \frac{1}{sbf^2 \times tbf}^{\circ^2} \times \text{ time units}$$

4.4 Link Discovery

Given that all the entities are inserted in blocks using the aforementioned *Blocking* technique, we check a relation $r \in R_s \cup R_t \cup R_{st}$ only within the scope of each block. In order to achieve this, we use the actual spatial and/or temporal information, computing explicitly the relation.

Since the blocks are built to be completely independent of each other, this check can be performed in parallel with respect to the blocks. Then we construct the set of discovered links (DL_r) by aggregating the respective links that have been discovered within each block.

Spatial relation *Disjoint* is treated in a slightly different way from the other relations. If two entities belong to the same block, then we follow the previous approach and we check explicitly the relation. If they don't, the *Disjoint* relation holds by definition and thus we add a link without actually checking the relation. A similar approach is followed for the temporal relations *Before* and *After* as well as for the cardinal direction relations.

4.5 Soundness and Completeness

The soundness and completeness of the proposed methods for Spatial and Temporal Link Discovery is proved in two steps. In the first step we prove that the methods are sound and complete when performing an exhaustive check of all the possible pairs of entities (Cartesian product) and in the second that the *Blocking* technique that we propose does not affect the accuracy of the discovered links.

Cartesian Product Technique. The proposed algorithms that check if a spatial or temporal relation holds between two geometries or time instants have been proven sound in [1, 5, 24]. Also, by definition, Cartesian product denotes that we perform an exhaustive (complete) check of all the pairs of entities from the datasets. Hence, we can state that, our methods are sound and complete i.e., for each relation they discover the exact set of pairs of entities for which this relation holds.

Blocking Technique. We prove that the application of the *Blocking* technique does not affect the completeness of our methods for Spatial Link Discovery with reduction to absurdity.

PROOF. Let two geometries lying in different blocks with a spatial relation $r \in R_s \setminus \{Disjoint\}$ holding between them.

If r holds between two geometries then they intersect at least at one point (from the definition of the spatial relations).

If two geometries intersect at one point, so do their $MBBs$ (from the definition of MBB).

If the $MBBs$ of two geometries intersect, then there is at least one block in which they will be both inserted (as described above).

This results in a contradiction because we assumed that the two geometries are lying in different blocks. Therefore the initial assumption must be false. □

The above proves that if a relation other than *Disjoint* holds between two geometries, then they will be placed in at least one common block and consequently the relation between them will be checked and discovered.

With a similar proof for the *Disjoint*, the temporal and the cardinal direction relations we can state that our methods remain complete, even after the application of the *Blocking* technique.

4.6 Precision and Recall

Since we proved theoretically that our methods are sound and complete, their accuracy is guaranteed. Metrics such as *Precision* and *Recall* are by definition equal to 100%:

$$\begin{aligned} Precision &= \frac{TDL}{TDL + FDL} = \frac{TDL}{TDL} = 100\% \\ Recall &= \frac{TDL}{TDL + FNDL} = \frac{TDL}{TDL} = 100\% \end{aligned}$$

TDL stands for True Discovered Links, FDL for False Discovered Links and $FNDL$ for False Not Discovered Links. FDL and $FNDL$ are both equal to zero as a consequence of the soundness and the completeness of our methods.

[11] The longitude range of WGS 84 is $[-180^\circ, 180^\circ]$ and the latitude range is $[-90^\circ, 90^\circ]$.

[12] The Epoch has been set to January 1, 1970, 00:00:00 GMT.

5. EXTENDING THE LINK DISCOVERY FRAMEWORK SILK

All the proposed methods for Spatial and Temporal Link Discovery have been implemented as extensions to the Silk framework[13]. As mentioned in Section 2, Silk is the only, to the best of our knowledge, open-source generic framework for discovering relationships between data items within different Linked Data sources.

For our implementation we extended transparently the core components of the Link Discovery Engine of Silk in the following phases:

- *Transformation Phase.* In this phase, Silk reads the incoming entities from the data sources. As an optional step, a transformation operator can be applied to normalize the attributes of these entities. For this phase we implemented the transformation operators that we presented in Section 4.

- *Blocking Phase.* Silk employs a blocking technique which maps entities to a multidimensional index. After the mapping, the entities are divided into multidimensional and optionally overlapping blocks. Blocking works on arbitrary link specifications and no separate configuration is required. For our implementation we adapted the blocking technique that we introduced in Section 4 to the one that Silk uses, by utilizing a two-dimensional index for the spatial blocking and a one-dimensional for the temporal and a three-dimensional for the spatiotemporal blocking.

- *Link Generation Phase.* Finally, a distance operator computes the distance for each pair of entities that have been inserted in the same block and if this does not exceed a given threshold, it writes the pair to the output. For this phase we implemented distance operators for all the spatial[14] and temporal relations that were introduced in Section 4.

On top of the Link Discovery Engine, Silk implements two main applications. The first is used for generating RDF links on a single machine and is called *Silk Single Machine.* The datasets that must be interlinked can either reside on the same machine or on remote machines which are accessed via the SPARQL protocol.

The second application is *Silk MapReduce* which is used for generating RDF links between datasets using a cluster of multiple machines. *Silk MapReduce* is based on Hadoop[15] and it can scale out to very big datasets by distributing the link generation to multiple machines.

In both of these applications our *Blocking* technique divides the source datasets into blocks and then the distance operators run in parallel with respect to the blocks. In *Silk Single Machine* we have multi-thread parallelization and in *Silk MapReduce* multi-machine parallelization.

6. EXPERIMENTAL EVALUATION

In this section we experimentally evaluate the spatial and temporal extensions of Silk by using it in a real-world scenario. The datasets, detailed instructions and other useful information for reproducing the experiments are publicly available[16].

6.1 Compared Frameworks

As we discussed in Section 2, to the best of our knowledge, there is no related framework with which we can discover spatial or temporal relations other than equivalences among RDF datasets. Hence, in the experiments that we conducted, we compared against variants of Silk and the state-of-the-art spatiotemporal RDF store Strabon [18].

Strabon is not considered as a Link Discovery framework but since it supports the *GeoSPARQL* and *stSPARQL* query languages, *NAMED GRAPHS* and *CONSTRUCT* queries it can be employed for discovering spatiotemporal relations e.g., the `intersects` relation, by posing Link Discovery queries like the one depicted in Figure 3(a). The only restriction that we face with Strabon is that both the source and the target datasets must be stored locally, in different named graphs. On the other hand, with Silk, we can interlink a local dataset with a remote one, that is published by another data publisher. The only access that we need to it, is via a SPARQL endpoint.

6.2 Environment of Experiments

We conducted our experiments both in a single machine and a distributed environment. For the single machine environment, we used a machine with two Intel Xeon E5620 processors (12MB L3 cache, 2.4 GHz), 32 GB of RAM and a RAID-5 disk array that consists of four disks (32 MB cache, 7200 rpm). For the distributed environment we used a cluster provided by the European Public Cloud Provider Interoute[17], in which we reserved 1 Master and 20 Slave Nodes with 2 CPUs, 4GB RAM and 10GB disk each.

We ran our experiments using the latest version of Silk with the spatial and temporal enhancements (v2.6.1) and the latest version of Strabon (v3.2.10) with accordingly tuned PostgreSQL (v9.1.13) and PostGIS (v2.0) as proposed by the developers.

6.3 Scenario

In [16] the authors present a real-time wildfire monitoring service that exploits satellite images and linked geospatial data to detect and monitor the evolution of fire fronts. This service is now operational at the National Observatory of Athens and is being used during the summer season by emergency managers monitoring wildfires in Greece[18].

A part of the processing chain of the service is to improve the thematic accuracy of the detected fires (hotspots) by correlating them with auxiliary geospatial data. More specifically, the service finds the land cover of each area which is threatened at a specific time by a hotspot, in order to avoid false alarms from fires started e.g., by farmers as part of their agricultural practices. Also, it finds the municipalities that a hotspot threatens and thus, competent authorities are made aware about the existence of a fire in their area of

[13] The source code of the spatial and temporal extensions of Silk is publicly available here: https://github.com/silk-framework/silk.

[14] For the cardinal direction relations, the developed operators support only points and not complex geometries as happens with all the other relations.

[15] http://hadoop.apache.org

[16] http://silk.di.uoa.gr

[17] http://www.interoute.com

[18] http://ocean.space.noa.gr/fires

```
CONSTRUCT {?s strdf:intersects ?t .} WHERE{
GRAPH ex:source{?s geo:hasGeometry/geo:asWKT ?sg.
                ?s strdf:hasValidTime ?st.}
GRAPH ex:target{?t geo:hasGeometry/geo:asWKT ?tg.
                ?t strdf:hasValidTime ?tt.}
FILTER(geof:sfIntersects(?sg, ?tg) &&
      strdf:intersects(?st, ?tt))}
```

(a)

		Geometries		Time Elements	
Dataset	**#Entities**	**Type**	**#Points**	**Type**	**#Instants**
GAG	325	Polygons	979,929	Periods	650
CLCG	4,868	Polygons	8,004,058	Periods	9,736
HG	37,048	Polygons	148,192	Instants	37,048

(b)

Figure 3: (a) Example of a Link Discovery Query and (b) Characteristics of the Datasets

responsibility.

Below, we provide a short description of the datasets of the scenario, whilst in Figure 3(b) we present some quantitative characteristics of them. These datasets also constitute a subset of the datasets used in the state-of-the-art benchmark for Geospatial RDF Stores, Geographica [8]:

- *Hotspots of Greece (HG).* The HG dataset contains the location and the acquisition time of detected fires for each fire season as produced by the National Observatory of Athens[19] after processing appropriate satellite images.

- *CORINE Land Cover of Greece (CLCG).* The Corine Land Cover project[20] is an activity of the European Environment Agency that provides data regarding the land cover of European countries. The CLCG is a subset of the whole dataset that contains all the available information about Greece.

- *Greek Administrative Geography (GAG).* The GAG dataset contains an ontology that describes the administrative divisions of Greece (prefectures, municipalities, districts, etc.) which has been populated with relevant data that is publicly available in the Greek open government data portal[21].

With Silk, the above scenario can be translated into two interlinking tasks between the HG and the CLCG and the GAG datasets respectively. In these tasks, Silk discovers the spatiotemporal relation *intersects* between the datasets. For this, it first applies a *CRS transformation* to normalize the heterogeneous geometries and a *Period* and a *TimeZone transformation* to normalize the heterogeneous time elements of the datasets. Finally, it populates the three-dimensional spatiotemporal blocks and discovers the relation *intersects* within each one of them. Thus, if a hotspot threatens a municipality or a land cover area, then the output of this procedure will contain the respective entities from the HG, GAG and CLCG datasets, interlinked with the predicate `strdf:intersects`.

6.4 Parameters sbf and tbf

As we have discussed in Section 4, sbf and tbf adjust the area and the length of the blocks in which we divide the space and the time respectively. The bigger they get, the more and smaller blocks are created.

The optimal value for sbf depends on the distribution and the size of the geometries which are not known in advance in the case of our scenario. On the other hand, given the metadata of the datasets, we can easily compute the optimal value for tbf. The temporal resolution of HG dataset is 15 minutes[22], while the one of GAG and CLCG is a multiple of a year (e.g., 5 years). By constructing blocks with temporal dimension equal to 1 year, we can guarantee that, for all the pairs of entities of the datasets HG-GAG and HG-CLCG respectively which belong to the same block, the temporal relation *intersects* holds by design (i.e., we do not have to check it explicitly between each pair). Hence, the optimal value for tbf is 1 year.

6.5 Experiment 1: Adjusting the Spatial Blocking Factor

Since we know the optimal value for tbf, in this experiment we try to approximate the optimal value for sbf. In order to achieve this, we analyze the performance of the single machine implementation of Silk with respect to different values of sbf. Particularly for this experiment, we use the full datasets of the scenario and we measure separately the computation times of HG-GAG and HG-CLCG. Furthermore, we count the total number of discovered links in each of these executions.

The graph of Figure 4(a) summarizes the results of this experiment. When sbf takes values close to 0, the blocks are spanning big surfaces of the earth. Hence, most of the geometries of the datasets are inserted in the same block, making the link discovery procedure almost as time consuming as the computation of the Cartesian product.

As sbf gets bigger, Silk seems to perform better. However, this improvement continues until a certain value (value 10) and then, as the sbf increases, the computation time deteriorates. This is due to the fact that a big value for sbf causes the division of space into very small blocks and thus each geometry is inserted into a big number of them. If two geometries are inserted into multiple blocks, then the check of the spatial relation is performed independently in each block that they appear. Hence, in this case we have redundant checks of the same relation that decrease the performance of Silk.

Another useful outcome from this experiment is the comparison of the time consumed for the link discovery task between HG-GAG and HG-CLCG. Figure 4(a) shows that the HG-CLCG interlinking takes orders of magnitude more than the respective HG-GAG. In Figure 3(b) we observe that the CLCG dataset has more geometries than GAG, whereas GAG has more complex ones (with respect to the number of points and not necessarily to their shape). Hence, in cases of spatial relations like `intersects`, the bottleneck is the number and not the complexity of the geometries.

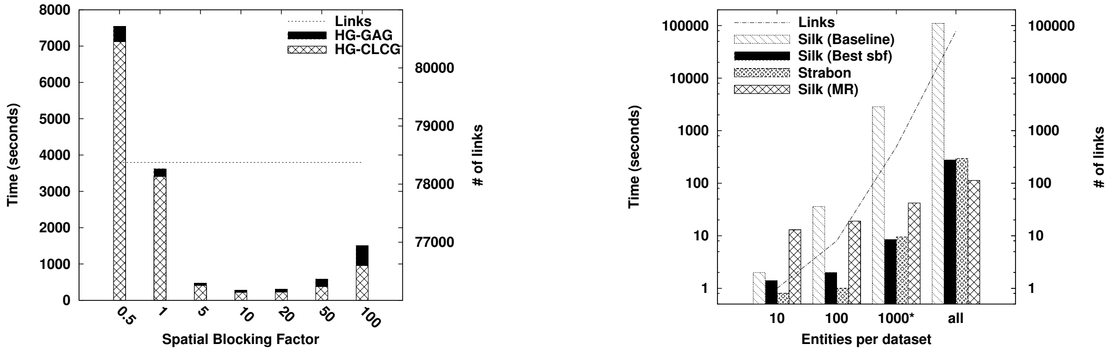
One final observation from this experiment is the number

[19] http://www.noa.gr

[20] http://www.eea.europa.eu/publications/COR0-landcover

[21] http://geodata.gov.gr

[22] METEOSAT Second Generation (MSG) satellites return images every 15 minutes.



(a)

(b)

Figure 4: Experiments of adjusting (a) the Spatial Blocking Factor and (b) the Entities per Dataset

of discovered links. This number remains the same independently from the value of *sbf*. This is expected, since we have already proven in Section 4 that the *Blocking* technique that we propose does not affect the accuracy of the discovered links which remains 100%.

6.6 Experiment 2: Adjusting the Entities per Dataset

In the second experiment we analyze the performance of three variants of Silk and Strabon with respect to different number of entities per dataset. For that reason, we use four different subsets of the datasets of the scenario[23] and we measure the total execution time for performing the interlinking tasks i.e., we don't distinguish between HG-GAG and HG-CLCG.

The first variant (Silk (Baseline)), which we consider as baseline, computes the full Cartesian product of the entities and then it checks if the spatiotemporal relation *intersects* holds between them. The second variant (Silk (Best sbf)), utilizes the *Blocking* technique with the best *sbf*, as the latter occurred from the previous experiment. The third one (Silk (MR)), is the distributed variant of Silk, which also utilizes the *Blocking* technique with the best *sbf*. In the case of Strabon, the datasets are stored locally and a `CONSTRUCT` query like the one we described in Figure 3(a) is performed.

The results of this experiment can be seen in Figure 4(b). As we can see from the graph, Silk (Baseline) is the most inefficient implementation, since even for the 1000 entities per dataset it is more time consuming that all the other implementations are for the full datasets.

On the other hand, Strabon seems to be faster for small number of entities per dataset whereas Silk (Best sbf) is faster when interlinking the full datasets. This happens because Silk fully utilizes the cores of the running machine by assigning the workload of each block it creates into a new thread. For big datasets, where the total workload is big enough, the *Blocking* approach of Silk seems to be the most efficient.

The effect of the massive parallelization is more remarkable with the distributed variant of Silk (Silk (MR)). With Silk (MR) the total workload is divided into different machines and in each machine it is divided into different cores. Also, we observe that, the computation time of *Hadoop* for small datasets is negligible with respect to the initialization time and the time consumed to copy the data from the local file system to the *Hadoop Distributed File System* (HDFS) and vice versa. Hence, Silk (MR) outperforms the other Silk variants and Strabon only for the measurement with the full datasets. Also, we can observe from the graph that it has the best scaling factor. Hence, we can claim that, for large datasets it is the only viable solution.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed new methods for Spatial and Temporal Link Discovery and provided the first implementation of our techniques based on the well-known framework Silk. Silk, enhanced with the new features, allows data publishers to generate a wide variety of spatial, temporal and spatiotemporal relations between their data and other Linked Open Data, dealing effectively with the common heterogeneity issues of such data. Furthermore, we experimentally evaluated our implementation by using it in a real-world scenario, with datasets that comprise rich spatial and temporal information, and demonstrate that it discovers accurately all the existing links in a time efficient and scalable way.

Future work concentrates on extending Silk with more spatial and temporal relations. These relations will be based on algebras and calculi that appear frequently in the relevant bibliography and are useful for specific use cases. We will also examine how we can estimate the optimal value of the blocking factor, for both the spatial and the temporal dimension, by posing preprocessing queries on the datasets that we want to interlink. Finally, we will try approximate blocking techniques which are more efficient than the one we propose but not 100% accurate.

8. ACKNOWLEDGEMENTS

This work has been partially funded by the FP7 projects LEO (611141) (http://linkedeodata.eu and MELODIES (603525) (http://www.melodiesproject.eu). We also want to thank Hervé Caumont and Cesare Rossi from Terradue (http://www.terradue.com) for helping us conduct our experiments to a cloud environment.

[23] The GAG dataset contains less than 1000 entities (Figure 3(b)) and thus for the third measurement of the experiment (Figure 4(b)) we used the full dataset.

9. REFERENCES

[1] J. F. Allen. Maintaining knowledge about temporal intervals. *Commun. ACM*, 26(11):832–843, Nov. 1983.

[2] S. Auer, J. Lehmann, A.-C. N. Ngomo, and A. Zaveri. Introduction to Linked Data and Its Lifecycle on the Web. In S. Rudolph, G. Gottlob, I. Horrocks, and F. van Harmelen, editors, *Reasoning Web*, volume 8067 of *Lecture Notes in Computer Science*, pages 1–90. Springer, 2013.

[3] K. Bereta, P. Smeros, and M. Koubarakis. Representation and querying of valid time of triples in linked geospatial data. In P. Cimiano, O. Corcho, V. Presutti, L. Hollink, and S. Rudolph, editors, *The Semantic Web: Semantics and Big Data*, volume 7882 of *Lecture Notes in Computer Science*, pages 259–274. Springer Berlin Heidelberg, 2013.

[4] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data - The Story So Far. *International Journal on Semantic Web and Information Systems*, 5(3):1–22, 2009.

[5] E. Clementini, P. Di Felice, and P. van Oosterom. A small set of formal topological relationships suitable for end-user interaction. In D. Abel and B. Chin Ooi, editors, *Advances in Spatial Databases*, volume 692 of *Lecture Notes in Computer Science*, pages 277–295. Springer Berlin Heidelberg, 1993.

[6] M. Egenhofer. A formal definition of binary topological relationships. In W. Litwin and H.-J. Schek, editors, *Foundations of Data Organization and Algorithms*, volume 367 of *Lecture Notes in Computer Science*, pages 457–472. Springer Berlin Heidelberg, 1989.

[7] Z. Gantner, M. Westphal, and S. Woelfl. GQR - A Fast Reasoner for Binary Qualitative Constraint Calculi. In *AAAI Workshop on Spatial and Temporal Reasoning*, 2008.

[8] G. Garbis, K. Kyzirakos, and M. Koubarakis. Geographica: A benchmark for geospatial rdf stores (long version). In *The Semantic Web–ISWC 2013*, pages 343–359. Springer, 2013.

[9] A. Gerevini and B. Nebel. Qualitative spatio-temporal reasoning with rcc-8 and allen's interval calculus: Computational complexity. In *ECAI*, volume 2, pages 312–316, 2002.

[10] S. Giannakopoulou, C. Nikolaou, and M. Koubarakis. A reasoner for the RCC-5 and RCC-8 calculi extended with constants. In C. E. Brodley and P. Stone, editors, *Proceedings of the 28th AAAI Conference, Québec, Canada.*, pages 2659–2665. AAAI Press, 2014.

[11] R. Goyal and M. Egenhofer. Cardinal Directions Between Extended Spatial Objects. *IEEE Trans. on Data and Knowledge Engineering*, 2000.

[12] O. Hassanzadeh, A. Kementsietsidis, L. Lim, R. J. Miller, and M. Wang. A framework for semantic link discovery over relational data. In *Proceedings of the 18th ACM conference on Information and knowledge management*, pages 1027–1036. ACM, 2009.

[13] R. Isele and C. Bizer. Active learning of expressive linkage rules using genetic programming. *Web Semantics: Science, Services and Agents on the World Wide Web*, 23:2–15, 2013.

[14] R. Isele, A. Jentzsch, and C. Bizer. Efficient multidimensional blocking for link discovery without losing recall. In *WebDB*, 2011.

[15] M. Koubarakis. Linked Open Earth Observation Data: The LEO Project. In *Image Information Mining Conference: The Sentinels Era*. ESA-EUSC-JRC, 2014.

[16] M. Koubarakis, C. Kontoes, and S. Manegold. Real-time wildfire monitoring using scientific database and linked data technologies. In *Proceedings of the 16th International Conference on Extending Database Technology*, pages 649–660. ACM, 2013.

[17] M. Koubarakis and K. Kyzirakos. Modeling and querying metadata in the semantic sensor web: The model stRDF and the query language stSPARQL. In *ESWC*, 2010.

[18] K. Kyzirakos, M. Karpathiotakis, and M. Koubarakis. Strabon: a semantic geospatial dbms. In *The Semantic Web–ISWC 2012*, pages 295–311. Springer, 2012.

[19] M. Nentwig, M. Hartung, A.-C. N. Ngomo, and E. Rahm. A survey of current link discovery frameworks. *Semantic Web Journal*, 2015.

[20] A.-C. N. Ngomo and S. Auer. Limes: A time-efficient approach for large-scale link discovery on the web of data. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence - Volume 3*, IJCAI'11, pages 2312–2317. AAAI Press, 2011.

[21] A.-C. Ngonga Ngomo. Orchid - reduction-ratio-optimal computation of geo-spatial distances for link discovery. In *Proceedings of ISWC 2013*, 2013.

[22] OGC. GeoSPARQL - A geographic query language for RDF data, November 2010.

[23] G. Papadakis, E. Ioannou, T. Palpanas, C. Niederée, and W. Nejdl. A blocking framework for entity resolution in highly heterogeneous information spaces. *Knowledge and Data Engineering, IEEE Transactions on*, 25(12):2665–2682, 2013.

[24] D. A. Randell, Z. Cui, and A. G. Cohn. A spatial logic based on regions and connection. In *KR*, pages 165–176, 1992.

[25] J. Salas and A. Harth. Finding spatial equivalences accross multiple RDF datasets. In *Proceedings of the Terra Cognita Workshop on Foundations, Technologies and Applications of the Geospatial Web*, pages 114–126. Citeseer, 2011.

[26] V. Sehgal, L. Getoor, and P. D. Viechnicki. Entity resolution in geospatial data integration. In *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, pages 83–90. ACM, 2006.

[27] S. Skiadopoulos and M. Koubarakis. Composing cardinal direction relations. *Artificial Intelligence*, 152(2):143 – 171, 2004.

[28] P. Smeros. Discovering Spatial and Temporal Links among RDF Data. In M. Koubarakis, editor, *Master Thesis*. National and Kapodistrian University of Athens, 2014. Available from: http://openarchives.gr/view/2547590.

[29] K. Taylor and E. Parsons. Where is everywhere: Bringing location to the web. *Internet Computing, IEEE*, 19(2):83–87, Mar 2015.

[30] L. M. Vilches-Blázquez, V. Saquicela, and O. Corcho. Interlinking geospatial information in the web of data. In *Bridging the Geographic Information Sciences*, pages 119–139. Springer, 2012.