Deeply Integrating Linked Data with Geographic Information Systems

Gengchen Mai, Krzysztof Janowicz, Bo Yan

STKO Lab, University of California, Santa Barbara, USA

Simon Scheider

Utrecht University, Netherlands

Abstract

The realization that knowledge often forms a densely interconnected graph has fueled the development of graph databases, Web-scale knowledge graphs and query languages for them, novel visualization and query paradigms, as well as new machine learning methods tailored to graphs as data structures. One such example is the densely connected and global Linked Data cloud that contains billions of statements about numerous domains including life science and geography. While Linked Data has found its way into everyday applications such as search engines and question answering systems, there is a growing disconnect between the classical ways in which GIS are still used today and the open-ended, exploratory approaches used to retrieve and consume data from knowledge graphs such as Linked Data. In this work, we conceptualize and prototypically implement a Linked Data connector framework as a set of toolboxes for Esri's ArcGIS to close this gap and enable the retrieval, integration, and analysis of Linked Data from within geographic information systems. We discuss how to connect to Linked Data endpoints, how to use ontologies to probe data and derive appropriate GIS representations on-the-fly, how to make use of reasoning, how to derive data that is ready for spatial analysis out of RDF triples, and, most importantly, how to utilize the link structure of Linked Data to enable analysis. The proposed Linked Data connector framework can also be regarded as the first step towards a guided geographic question answering system over geographic knowledge graphs.

Keywords: Linked Data, Knowledge Graphs, Path Queries, GeoEnrichment, Ontology, SPARQL, Geographic Information Systems, Geographic Question Answering

1 Introduction and Motivation

Linked Data, and more generally the idea of making semantically-annotated raw data available on the Web, has taken information technologies by storm. Today, knowledge graphs[1] power intelligent assistance systems such as Apple's Siri and search engines such as Google. Millions of Webpages contain semantic markup, and the publicly available part of the Linked Data cloud contains approximately 150 billion triples distributed over 10000 datasets and connected to each other by millions of links. The Linked Data paradigm offers a radically new perspective on structuring, publishing, discovering, accessing, reusing, and integrating data, thereby addressing many key challenges of GIScience and cyber-data infrastructures (Kuhn et al., 2014). Geographic data play a prominent role in the Linked Data cloud as places act as central nexuses that interconnect events, people, and objects. Thus, unsurprisingly, geo-data sources are among the most densely interlinked and central hubs. In fact, the rapidly increasing amount of geo-data published on the Web led to the first joint working group[2] of the Open Geospatial Consortium (OGC) and the World Wide Web Consortium (W3C). The group was tasked to provide best practice and bridge between specifications for OGC Web Services and the Semantic Web technology stack (van den Brink et al., 2019; Haller et al., 2019).

Despite all these success stories, from a Geographic Information Systems perspective, Linked Data seems almost like a one-way street. Given a continuously growing stack of open-source tools, it is now easier than ever to publish and consume geo-data on the (Semantic) Web, e.g., by converting shapefiles to RDF, fusing geometries from different sources (Giannopoulos et al., 2014), discovering links (Ngomo and Auer, 2011; Mai et al., 2016), querying remote endpoints (Battle and Kolas, 2012), or computing geospatial properties on-demand (Regalia et al., 2016). Nonetheless, all this work focuses merely on how to get geo-data out of *data silos*. The question of how to actually make *use* of this plethora of data remains largely unanswered. Typically, the retrieved data are either used directly, e.g., when querying for the location and construction year of the Empire State Building, or *flattened* into tabular form and used in environments such as R, e.g., when computing clusters from Point Of Interest (POI) data accessed using the LinkedGeoData hub (Stadler et al., 2012).

While it is possible to convert RDF-based Linked Data into a format that can be handled by modern GIS, e.g., via JSON or CSV import, nothing is gained in the process that would not have been available from bulk downloads or OGC Web Services such as the Web Feature Service (WFS) as the data are flattened and the link structure is largely lost. The transparent encapsulation of Linked Data and Semantic Web services within OGC Web Services has been proposed to seamlessly bridge between the Geo Web and the Semantic Web (Janowicz et al., 2010). Following this approach, Diekhof (2010) implemented a Web Reasoning Service (WRS) encapsulated in a OGC Web Processing Service (WPS) hull and Jones et al. (2014) implemented an adapter which converts WFS requests into SPARQL queries to be executed over Linked Data. Realizing that using Linked Data within a GIS has potential beyond simply being yet another data source, Zhu et al. (2016) and Iwaniak et al. (2016) demonstrated how spatial analysis can improve the quality of Linked Data, improve data fusion, ontology alignment, and so forth. In short, while we can semantically enrich geo-data and publish them as Linked Data, consuming these data in a GIS, and, thereby, applying

[1] It is worth noting that the term Data Graph or Statement Graph may more truthfully represent the relation between data, information, and knowledge. However, we will use the established terminology throughout this paper.

[2] https://www.w3.org/2015/spatial/wiki/Main_Page

the vast toolboxes of modern spatial analysis is more difficult, especially if we aim at maintaining the link structure while doing so.

At first glance, the aforementioned situation appears as nothing more than a software engineering task, namely an improved import functionality for GIS that would add Linked Data to the already large set of data formats that can be consumed. However, Linked Data is not a data format. It is a paradigm that follows strategies required for Web-scale, distributed data infrastructure and does not harmonize well with how we conceptualize data (exchange) in GIS. This requires new ways of thinking about how GIS and its users should interact with Linked Data, which concrete benefits Linked Data brings to the table with respect to spatial analysis, how these key benefits of Linked Data can be maintained during conversion into GIS data formats and analysis without having to flatten the data back to a tabular format, and, finally, how to utilize the ontologies used to semantically lift Linked Data instead of merely relying on strings, e.g., to represent place types. To give but one example of the arising conceptual challenges, it is worth noting that in contrast to relational (geo-)databases Linked Data follows the so-called Open World Assumption (OWA) by which the truth value of statements (here RDF triples) is irrespective of whether they are *known* to be true or not.

In this work, we will address some of the aforementioned questions by conceptualizing and prototypically implementing a Linked Data connector framework for the *deep* integration of Linked Data and GIS. In contrast to prior work, deep integration refers to the ability to actually make use of the Linked Data *paradigm* from within a GIS instead of following a typical Extract, Transform, Load (ETL) process in which Linked Data becomes just another data source. As GIS does not have a distinct graph data model, we generate geodatabase tables to store the results of SPARQL queries on-the-fly. This is partially related to work on mapping SPARQL to SQL or relational algebra more broadly (Cyganiak, 2005; Prudhommeaux and Bertails, 2008). We will discuss how to connect to distant Linked Data endpoints from within a GIS, how to use ontologies to probe data and derive appropriate GIS representations at query time, how to make use of reasoning from within a GIS, how to derive data that is ready for spatial analysis out of RDF triples, and, most importantly, how to utilize the link structure of Linked Data to enable analysis that would not be possible by downloading data from OpenStreetMap or a Web Feature Service. More concretely, we will address the meaningful selection of properties, the casting of so-called data-type and object-type properties, the usage of partonomical relations and the handling of non-functional properties, and, finally, the exploration of links to other spatial and non-spatial entities.

To give an intuitive example for the power of deeply integrating Linked Data with GIS, consider Alexander von Humboldt and his famous expeditions. A simple SPARQL query to a Linked Data endpoint can return the regions he traveled, the researchers and explorers he influenced/advised during his life, and the regions they studied in turn. Analyzing these regions, however, e.g., to discover patterns and clusters, would not be possible. The other way around, a GIS can provide the analytical capabilities for said regions but does not allow for their retrieval using the query illustrated above, namely *return those regions studied by students of von Humboldt*.

The contributions of this work are as follows:

- We demonstrate a deep integration of Linked Data into GIS, not merely data interchange by bulk download, import, and export, or by the transparent encapsulation of Linked Data using OGC Web Services. We will motivate the need for such deep integration by giving examples such as retrieving n-degree sister cities.

- We show how to utilize ontologies to address some of the key challenges involved in deeply integrating Linked Data on-demand. More specifically, we discuss how to guide the selection and pre-processing of property-object pairs and here in particular the transformation of non-functional properties into functional properties.

- We demonstrate how to utilize Semantic Web reasoning and ontologies to extract additional properties by using subsumption reasoning and (inverse) partonomical relations as examples. By doing so, we include triples that would not be available through a flat import and discuss use cases that showcase the need for reasoning support.

- Our deep integration supports exploratory search via n-degree property path queries, a feature that is not typically found in a GIS environment. More generally, we demonstrate how the queried Linked Data can be seamlessly used to perform GIS analysis.

Our evaluation is threefold, **(I)** we will provide an implementation of the Linked Data connector as a set of toolboxes for Esri's ArcGIS, **(II)** we will demonstrate the seamless use of the resulting data and compare these examples to GeoEnrichment, and **(III)** to show generalizability, we will use Wikidata as well as DBpedia as data sources.

Finally, the proposed Linked Data connector framework for GIS can be treated as the first step towards a guided geographic question answering system over geographic knowledge graphs (Scheider et al., 2018). Instead of using a natural language question as the input of the QA system, this framework takes a set of actions of users as the input (guides) such as clicks on the maps, selections of spatial relations. After converting the user input to SPARQL queries, the returned Linked Data is retrieved and converted into a GIS format. Some spatial analysis can be applied to it in order to answer geographic questions which require multiple spatial processing steps. The answer to the question will be visualized in the GIS itself. Put differently, because of the uniqueness of geographic questions, the users of a geographic QA system may benefit from expressing their questions by interacting with maps rather than phrasing their question in natural language.

The remainder of this paper is structured as follows. Section 2 briefly reviews related work and background readings. Next, section 3 introduces the general Linked Data connector framework and then discusses the individual challenges outlined above by discussing the conceptual aspects, providing an implementation as proof-of-concept, and then closing with an illustrative example. We provide an evaluation by comparison to Esri's Geo-Enrichment in section 4 and summarize our work and discuss directions for further work in section 5.

2 Related Work

Following our discussion of integration challenges in the introduction, in this section we review the state-of-the-art of Linked Data integration into GIS. We start with *shallow* forms of integration, which are mostly straightforward technical interfaces that do not involve interoperability on a conceptual level, and then proceed towards deeper forms of integration. Note that the way we proceed also reflects temporal order of these approaches, since deeper forms of integration have only very recently been addressed.

One approach to importing Linked Data into a GIS is by simply transforming it into a simple relational structure, following an Extract-Transform-Load (ETL) approach. Whereas tools like

D2RQ (Bizer and Seaborne, 2004) or Fluid Workbench (Fluid Operations, 2016) offer access to relational and file data sources from within the Web of data, there are also tools available for importing RDF into relational databases, such as Oracle (Oracle, 2016). Within the Web of data, appropriate geospatial data sources can be queried as SPARQL endpoints (Battle and Kolas, 2012), and geospatial RDF sources can be fused (Giannopoulos et al., 2014). An ETL approach is followed e.g., by the LinkedGeoData hub (Stadler et al., 2012), which started as a transformation of OpenStreetMap data to Linked Data. More generally speaking, there are various approaches for integrating relational databases and SQL with Linked Data and SPARQL (Prudhommeaux and Bertails, 2008; Cyganiak, 2005).

Other approaches aim at a deeper integration by enabling spatial query processing within SPARQL (Brodt et al., 2010) or by developing new standards or extensions for qualitative and quantitative spatial and temporal reasoning and querying over Linked Data (Battle and Kolas, 2012; Koubarakis and Kyzirakos, 2010). These approaches, however, largely focus on adding some basic spatial capabilities to Linked Data and not the other way around. To give a concrete example, they enable buffer queries for nearby places such as parks within 3km of the Washington Monument (Battle and Kolas, 2012) but cannot be used to compute the density of such parks, an isochrone map of travel distances to these parks, point pattern analysis to determine whether parks are clustered or regularly distributed, and so forth.

However, it has been recognized early that the Semantic Web has a big potential for enhancing open geospatial services and GIS operations themselves. These enhancements include reasoning services (Roman and Klien, 2009; Diekhof, 2010), Web Processing Services (WPS) (Janowicz et al., 2010) and Web Feature Services (WFS) (Donaubauer et al., 2007; Staub, 2007; Roth, 2011; Jones et al., 2014). The former two allow the use of ontologies in geo-computation and service chaining (Yue et al., 2007). The latter enables interactive access to Linked Data from within standard GIS data interfaces as defined by the Open Geospatial Consortium. While these approaches are already interactive in the sense that they allow Linked Data queries to be materialized as Web service requests, they lack a possibility to directly make use of RDF and graph based-queries from *within* a GIS.

For this reason, the Web of data and GIS still remain largely separated as the data and service pipelines established between them only act as temporal connections during data selection and import. What is needed to overcome this gap is a way to integrate the different *paradigms and workflows* behind GIS and Linked Data (Kuhn et al., 2014). There are particular advantages to both for handling and processing information that ideally should be preserved from one world to the other. For example, a particular strength of Linked Data is that meta data is on the same syntactical level as data (Kuhn et al., 2014), enabling exploratory querying (Scheider et al., 2017), and, thus, meaningful exploration of what geospatial data are available (Olieman et al., 2015) without relying on a separate catalog service. In this way, available RDF properties that link resources to their attributes or to other objects can be explored by users in visual graphs and maps using SPARQL (Mai et al., 2016; Scheider et al., 2017).

Another question is also how Linked Data should be systematically turned into GIS tables, attributes, and data types. GIS data formats have been translated to RDF before and ordinary data types (integer, double, string) reappear in both GIS and RDF. However, it remains unclear how GIS feature tables (corresponding to GIS layers, i.e., collections of spatial objects of similar type with attributes) can be systematically built from RDF and how to most efficiently handle complex graph structures from within a GIS without having to immediately flatten them. More generally, it has not yet been investigated how such deep integration approaches can be incorporated into an existing GIS architecture such as ArcGIS or QGIS.

A successful deep integration would have to enable querying for Linked Data from within a GIS, exploring the results to retrieve more data by following the graph, ingesting these data in a way that makes them ready for spatial analysis, and, finally, performing such analysis seamlessly, i.e., by following established GIS workflows as defined by layers and toolboxes. We will demonstrate our work towards achieving these steps one at a time in the following section.

Successfully integrating Linked Data into GIS is also a first step to approaching a GIS from a knowledge graph-based (geographic) question answering (QA) perspective (Yih et al., 2016; Liang et al., 2017; Berant et al., 2013; Liang et al., 2018; Scheider et al., 2018). Given a natural language (geographic) question, e.g., *what is the most densely populated city west of the Mississippi*, a knowledge graph-based (geographic) QA system *translates* this question into programs (e.g. λ-calculus (Yih et al., 2015; Liang et al., 2017), SPARQL queries) which will be executed on its underlining knowledge graph. The result of executing programs (in some sequence) will be the answer to the (geographic) question. This geographic question answering functionality is not supported by current GIS for two reasons. First, GIS cannot directly operate on geographic knowledge graphs but layer-based geospatial data. Second, the *translation* or *semantic parsing* ability of QA systems from natural language questions to programs is not supported by any GIS.

Our proposed Linked Data framework is an attempt in this direction in the sense that 1) we let the users directly interact with a geographic knowledge graph within a GIS and 2) each implemented toolbox *translates* a set of user input into SPARQL queries which will be executed on the connected knowledge graph and return answers. Neither our work nor existing GIS can handle natural language questions directly. In fact, due to the uniqueness of geographic questions (Mai et al., 2019), only allowing users to express geographic questions in natural language may put some restriction on the system. For example, instead of expressing a location as a set of coordinates, it might be easier for the user to select (or draw) the region of interest on the map. Our work starts at the stage where a user's intent can be approximated by a SPARQL query. This idea is similar to the question-based spatial computing approach proposed by Vahedi et al. (2016). Instead of finding and building geoprocessing workflows on map layers, Vahedi et al. (2016) propose an alternative approach for spatial analysis which can let users ask geographic questions directly to a GIS. The other way around, Scheider et al. (2018) use questions to guide the selection of appropriate GIS methods.

3 Methods and their Prototypical Implementation

The presented work focuses on conceptual challenges arising from the different paradigms underlying GIS workflows and Linked Data, the required on-demand data models, and conceptual as well as technological solutions to these challenges. To do so, we discuss them step-by-step and implement *individual* toolboxes for Esri's ArcGIS 10.4 as proof-of-concept by using ArcPy. As depicted in Figure 1, our workflow starts with the retrieval of linked geographic data, e.g. *find all places within 10 miles of the selected location(s)*, which retrieves geographic entities (e.g. cities, POIs) from geographic knowledge graphs such as DBpedia and saves the resulting data in a geodatabase. Next, by using the retrieved geographic entities as the start point, the Linked Data Connector allows users to extract additional attribute information about these entities from knowledge graphs, e.g. *find all people who were born in these places*. Due to differences in data models used by geodatabases and geographic knowledge graphs, three major issues are considered by our toolbox: property selection, datatype casting, and relation normalization. They will be discussed in detail in Section 3.2. In the third step, our toolbox enables users to compute basic statistical information about the retrieved properties of the geographic entities. The last and the most important step of our framework is to enable users to explore a knowledge graph from within a GIS and help to answer n-degree relationship questions which are otherwise impossible to answer by a state-of-the-art GIS, e.g. *retrieve the affiliations of people influenced by Alexandar von Humboldt*. Jointly, these toolboxes form a complete workflow for Geographic Linked Data retrieval, extraction, enrichment, and exploration. We will provide concrete examples for each of those steps to demonstrate the seamless usage of the retrieved data. Put differently, every toolbox (on-the-fly) returns data in the form of records in a geodatabase that are ready for further analysis in a GIS.

Out of scope, however, are issues of (end user) usability such as proving convenient access to and search of Linked Data endpoints, handing timeouts, sorting of results, providing a documentation, and so on. More specifically, we do not claim that our practice of implementing solutions to each of the addressed challenges as an individual toolbox to support readability and keep conceptually separate aspects separate, is desirable from a user's perspective. We envision that full Linked Data support can be added to a GIS in the form of add-ins, be it for ArcGIS, QuantumGIS, GRASS, or any other system. Finally, while our research is concerned with the technology stack and paradigm underlying Linked Data, most of the presented results can be generalized to knowledge graphs in general including those not using W3C technologies.

3.1 Retrieving Linked Data

There are three major ways to retrieve geographic data and use them within a GIS. The traditional and still most common approach is to download or otherwise obtain data and load them into a local GIS. Alternatively, one can connect to an API or OGC Web Service, such as a Web Feature Service, to stream data into a local GIS and thereby ensure the data remain up-to-date. Finally, and of particular importance for very large datasets or computationally very intensive operations, one can use a cloud-based GIS where both, the data and the software remain online.

Here we focus on the second approach, but instead of connecting to a WFS or a transparent proxy, we will demonstrate interactive access to Linked Data. More concretely, our toolbox issues a SPARQL buffer/range query to retrieve Geographic Linked Data of a certain type by either reacting to individual mouse clicks (interpreted as center points) or by using an existing point feature shapefile. So far, most major Linked Data endpoints such as DBpedia, GeoNames, LinkGeoData, Wikidata, and so forth, do not supports GeoSPARQL for spatial query and most data sources represent features as points. However, simple range queries are usually supported. In this paper, we focus on conceptual issues such as data (schema) modeling and interaction, hence GeoSPARQL support can be added to our retrieval toolbox in the future without requiring any changes aside of the underlying queries.

For our framework, such range queries act as initial starting points, and, thus, we only retrieve the Uniform Resource Identifiers (URIs) of places within the buffer, their `rdfs:labels`, i.e., human-readable names, and their geographic coordinates. As the amount of retrieved resources may be very large, especially if subclass reasoning is enabled, and potentially deep, e.g., for nested blank nodes, we perform (non-spatial) attribute extraction in a separate step, and, thus, in an own toolbox. This also has the advantage that the retrieved data are immediately available for analysis such as kernel density estimation (KDE) which does not require any non-spatial attributes.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>

SELECT ?entityType ?entityTypeLabel
WHERE
{
  # find all entity types which are transitatively subclass of (
    ↪ wdt:P279) geographic location (wd:Q2221906)
  ?entityType wdt:P279* wd:Q2221906.

  # retrieve the English label
  ?entityType rdfs:label  ?entityTypeLabel .

  FILTER (LANG(?entityTypeLabel) = "en")
  FILTER REGEX(?entityTypeLabel, [search keywords])
}
```

Listing 1: A SPAQRL query to Wikidata for subclasses.

Figure 2 shows an example range query with one point feature in the broader Bay Area, the type `city` (`wd:Q515`) selected by the user, as well as a search radius (here, 10 miles). The optional *Input Place Type* parameter in Figure 2 provides users with a search-by-type approach to find and select an appropriate place type for their range query. Listing 1 shows the corresponding SPARQL query against the Wikidata endpoint to search for entity types which are (transitive) subtypes of `geographic location` (`wd:Q2221906`) with the user input keywords as the class label filter. Listing 2 displays a type restricted spatial range query which will be constructed and sent to the Wikidata SPARQL endpoint when the user executes the toolbox. The *Disable transitive subclass reasoning* checkbox depicted in Figure 2 provides the function to disable the transitivity reasoning in the spatial range query. For interactive range queries, the range query centers ([lon], [lat]) are determined by the mouse click positions of a user on the base map or by a point feature class while the buffer radius ([distance]) is given by users in miles. Note that the user can provide more than one location. Using GeoSPARQL, one could also use more complex polygons instead of buffers in the future. From a geographic question answering perspective, this toolbox enables the user to ask spatial range questions (e.g., *find all cities within 10 miles of the selected location(s)*) to a geographic knowledge graph.

```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX wdt: <http://www.wikidata.org/prop/direct/>
PREFIX geo: <http://www.opengis.net/ont/geosparql#>
PREFIX wikibase: <http://wikiba.se/ontology#>
SELECT distinct ?place ?placeLabel ?distance ?location
WHERE {
  # geospatial queries
  SERVICE wikibase:around {
    # the coordinates of a place
    ?place wdt:P625 ?location .
    # make a buffer around (longitude latitude)
    bd:serviceParam wikibase:center "Point([lon]␣[lat])"^^geo:
       ↪ wktLiteral .
    # buffer radius
    bd:serviceParam wikibase:radius [distance] .
    bd:serviceParam wikibase:distance ?distance .
  }
  # retrieve the English label
  SERVICE wikibase:label {bd:serviceParam wikibase:language "en".
     ↪  ?place rdfs:label ?placeLabel .}
  # wdt:P31 means instance of
  ?place wdt:P31 ?placeFlatType.
  # transitively subclass reasoning where wdt:P279 means subclass
     ↪  of
  ?placeFlatType wdt:P279* [type].
}
ORDER BY ?distance
```

Listing 2: A spatial buffer query to get all places within a buffer defined by the center and radius and of place types which are equal to or subclass of the given place type.

3.2 Attribute/Property Extraction

The next step is to extract attribute information for each spatial entity directly from the Linked Data Cloud to store the entities and their attributes, e.g., in an ArcGIS geodatabase, in order to perform geoprocessing on these data at some later stage. This step seems easy at the first glance. However, the difficulties arise from the question of how to meaningfully convert an open-ended, not contradiction-free, and highly heterogeneous Linked Data set into a well-controlled data format. For the current prototype, we consider the following three issues:

- **Properties Selection:** Different spatial entities types will have type-specific properties, e.g., cities serve as headquarters for companies while rivers do not. It is important to identify meaningful properties and let users choose which properties they want to get from the Linked

Data Cloud. Here meaningful refers to two separate aspects: (1) Properties should be commonly shared by the retrieved spatial entities to serve for further analysis, e.g., if the headquarter (`dbo:headquarter`) relation in DBpedia would only be known for a biased set of major companies and major cities, it may lead to misleading results. (2) The properties should be about the spatial entities themselves, not about the Web page describing them (e.g., `dbo:wikiPageID`). A typical example here would be the date a certain company was founded versus the date at which this information was released on the Linked Data cloud. Due to the nature of Web-scale and distributed knowledge graphs such as Linked Data, these distinctions are not always trivial to make. For properties that have a domain and range restriction definition in the underlying ontology, one can filter by these restrictions, e.g., to only select relations that hold between places, places and events, and so on. Otherwise, we simply default to counting how frequently the properties appear.

- **Datatype Casting of Datatype Properties:** To retain as much information as possible from non-structured and semi-structured data, Linked Data providers, such as DBpedia, typically do not restrict datatype properties to a specific XSD data type. For instance, `dbo:populationTotal` will have literals in the form of doubles, integers, and strings. The intuitive approach is to use a majority vote to cast literals to the datatype which has the highest frequency. However, if – from a conceptual perspective – the property is numeric, e.g., population totals or densities, casting literals from double or integer to string will not allow for their future usage within a GIS. Casting them from double to integer will lead to a loss in precision. To give a concrete example, aggregated or vague cognitive regions often have strings for population values if these indicate approximations such as in the triple (*dbr:South_Coast_(California) dbp:population "∼ 20 million"*). We measure the information value of current property values IV_i by the number of GIS computation operations Com_i they can be used in and the degree of information precision $Prec_i$ they have.

$$IV_i = w * Com_i + (1 - w) * Prec_i \quad (1)$$

- **Spatial Relation Normalization:** In relational database design various steps are performed to ensure efficient performance, storage, and maintainability. Examples include reducing redundancy by proper handling of relationship cardinality (1-1, 1-N, N-M), functional dependency, joins, and so forth. Since Linked Data allows properties to have multiple values, we cannot keep all those values in the same table. In order to decide on the cardinality of a specific property, we can employ the ontologies used to describe the queried datasets, if any. To give a concrete example, consider the property `birthPlace` and a historic figure such as Alexander von Humboldt as well as the property `population`. One would expect to receive Berlin as the place of birth and a population of around 3.7 million. In practice, however, one may receive many population values and even more than one birth place, if a dataset, here a Gazetteer, decides to distinguish between Berlin, Kingdom of Prussia and Berlin, Germany. Both cases are not uncommon. From a conceptual point of view, people can only have one birth place (which may be contested or unknown) and cities should only have one population count. Ontologies often express this 1-1 relation in the form of declaring a functional property (`owl:FunctionalProperty` and `owl:InverseFunctionalProperty`). If such relations are modeled explicitly, we append them to the main attribute table of spatial entities. Otherwise, we will separate the property from the main attribute table. Following Linked Data principles, URIs uniquely identify entities. Hence, we use URIs of spatial entities as primary keys for the main attribute table and foreign keys for each non-functional property's separated table. In ArcGIS file geodatabases, we dynamically generate a relationship class to indicate the foreign key information between the main attribute table and each non-functional property's separated table.

By following the three criteria outlined above, we implemented a spatial entity attribute extraction toolbox (see Figure 3) that takes the feature of a particular place type (entity set A) generated from the range query toolbox. Several SPARQL queries are generated to get the common properties of these spatial entities and their subdivisions. The interface divides these properties into four categories: common properties, inverse common properties, expanded common properties, and inverse expanded common properties. The first two property sets are composed of properties whose subjects (or objects for inverse common properties) are entities in A. The last two property sets are about properties whose subjects (or objects for inverse expended common properties) are entities which are transitively part of (`dbo:isPartOf`) entities in A. These expanded properties are optional for the users to extract. For all four property sets, we rank properties in each set by the number of entities in A with this property[3]. The user can select the appropriate properties to be extracted by selecting the checkbox of a specific property. Additionally, the property-value information we get for functional and non-functional properties are treated differently as we described in the third point above. Listing 3 shows one example query to obtain the information of a property in the inverse expanded common property set for all spatial entities in *A*.

From a question answering perspective, this toolbox allows the user to ask questions about the attributes of spatial entities (e.g. *What is the population/elevation/area/location of Los Angeles*) from a geographic knowledge graph. The extracted attribute information will be stored in a file geodatabase in real time and the information will be automatically updated when executing the

[3]This information is shown in parenthesis in the property set select box in Figure 3.

toolbox again, thereby keeping it in sync as the Linked Data Cloud evolves.

```
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX dbo: <http://dbpedia.org/ontology/>

SELECT ?wikidataSub ?subDivision ?o
WHERE
{
  # Obtain DBpedia IRI of spatial entities given its Wikidata IRI
  ?s owl:sameAs ?wikidataSub.

  # Obtain the spatial entities which is transitively part of the
     ↪ current entity
  ?subDivision dbo:isPartOf+ ?s.
  ?o [property] ?subDivision.

  VALUES ?wikidataSub
  {
    [....list of wikidata URIs of spatial entities in A....]
  }
}
```

Listing 3: A SPARQL query to get the information of a property in the inverse expanded common property set for entities in A.

3.3 Non-functional Property Conversion

For non-functional properties which are stored in separated tables, we provide the user with some basic statistical information to ease decision making, i.e., determining which (or how many) to use for a subsequent analysis. Examples include the mean value for population, the number of sister cities, and so forth. This step enables users to convert non-functional properties to functional properties using joins between the main attribute table and a non-functional property's separated table using custom merge rule. In our prototype, we provide SUM, MINIMUM (MIN), MAXIMUM (MAX), STANDARD DEVIATION (STDEV), MEAN, COUNT, FIRST, LAST, and CONCATENATE. Put differently, if a city has 8 sister cities, a GIS user can select to either keep the information about these sister cities separate and then perform operations on the resulting point pattern or decide to convert the information into a functional attribute such as the *count* of sister cities. In the first case, the user does not flatten the graph, while the second case essentially converts the richer feature-based representation into a non-spatial attribute for further analysis, e.g., sorting. Figure 4 shows the non-functional property conversion toolbox. The input feature class is still the feature class generated in the spatial range query which is enriched with some attribute information from the attribute extraction toolbox. The toolbox will list all the extracted non-functional properties according to the established relationship classes between the main attribute table and the separated tables of non-functional properties. Note that the operation SUM, MINIMUM, MAXIMUM, STANDARD DEVIATION, and MEAN are only available for properties whose converted field have a computable data type, e.g., integer or double. Combining this toolbox with the attribute extraction toolbox makes it possible to answer geographic questions which include aggregation or merging operations, e.g. *how many famous explorers/geographers traveled to today's Peru*.

Combining the attribute extraction and non-functional property conversion toolboxes also highlights the importance of considering both the ontology underlying a dataset and spatial reasoning on the data itself. To use the example above, the fact that the American explorer Hiram Bingham III re-discovered `Machu Picchu` during the 1911 Yale Peruvian Expedition implies (by transitivity) that he visited `Peru` without this information being directly present in the form of materialized RDF triples. This shows yet another advantage of retaining the graph structure as long as possible over approaches that flatten the data.

3.4 Relationship Exploration Between Spatial Entities

Strictly speaking, some of the examples given above, such as population counts, could be addressed using other techniques and data providers such as Esri's recent work on GeoEnrichment. The main difference here would be between a curated dataset on the one hand and an open-ended, partially uncurated dataset on the other hand. The differences between these two are analogous to authoritative data versus volunteered geographic information, where the latter gives up homogeneity and (potentially) accuracy for increased temporal and spatial coverage as well as multiple perspectives.

In the following we will explicitly address cases for which no counter-part exists in GeoEnrichment, Web Feature Services, and so on, namely path queries that span actors, events, and objects. In fact, our paper started with such an example – the regions studied by explorers that were influenced by von Humboldt. Put differently, to conduct a spatial analysis of these study regions requires a property path query of the form `Actor-InfluencedBy-Actor-studyRegion-Region`. Our prototype can consider various kinds of these n-degree path queries but we will limit ourselves to those that start and end with places so that the initial places can be selected using the retrieval toolbox and the path-ending places can be used for further analysis, e.g., studying point-patterns. These path queries can be entirely based on object properties (`owl:ObjectProperty`) or end with a datatype property `owl:DataTypeProperty` such as the area of the aforementioned regions.

Consider the example of sister cities. Wikidata models them by relation called `wdt:P190`. To return all sister cities of, say, Santa Barbara, CA, a user would first retrieve the resource for Santa Barbara using the retrieval toolbox and then translate the sister city relation to a data type property to compare to other cities or use our relation exploration toolbox (shown in Figure 5) to get the actual cities and their geometries instead. More interestingly, the user could define a longer path length and query for 2nd degree sister cities, i.e., cities that are sister cities of cities that are sister cities of Santa Barbara; see Figure 6. Such a query would correspond to the path `?inputCities wdt:P190/wdt:P190 ?outputCities`. Since such path queries can grow exponentially, we limit the maximum degree to four in our interface. We can even specify the direction of this relationship as ORIGIN, DESTINATION, and BOTH which will use the input spatial entities as subjects, objects, or both situations.

Finally, our interface does not restrict the predicates in a property path, i.e., not all of them have to be sister cities. Hence, one can also query for points of interests within sister cities of Santa Barbara, i.e., combine the sister city relation with a topological relation. To give a final example, one can map the current place of residence of famous alumni of the University of California, Santa

Barbara. It is these open-ended path queries where we see the most potential for our work and a strong complement to GeoEnrichment.

```
PREFIX wdt: <http://www.wikidata.org/prop/direct/>

SELECT distinct ?place ?o1 ?o2 ?o3
WHERE
{
  ?place wdt:P190 ?o1.
  ?o1 wdt:P190 ?o2.
  ?o2 wdt:P190 ?o3.
  VALUES ?place
  {
    [....list of wikidata URIs of spatial entities in A....]
  }
}
```

Listing 4: A SPARQL query to retrieval the 1-, 2-, and 3-degree sister city from the spatial entities.

4 Relation to GeoEnrichment

The presented Linked Data connector demonstrates a workflow for geographic Linked Data retrieval, attribute enrichment & conversion, and linkage exploration within a GIS. As far as we know, this is the first work about integrating Linked Data back to a GIS and making geographic Linked Data ready for spatial analysis that does not simply flatten the data. Instead, the presented methods and their implementation create geodatabases and their schema on-the-fly while the user is exploring and visualizing Linked Data. The only system that supports a subset of the presented capabilities is the GeoEnrichment[4] service recently developed by Esri. In the following, we will discuss similarities and differences between our framework and GeoEnrichment.[5]

The GeoEnrichment service aims at providing the ability to get facts/attributes about a location or an area. Basically, it enriches geographic data by adding demographic and landscape attributes about the input places. This service is provided in two ways: 1) as an ArcGIS Javascript API and 2) as an 'Enrich Layer'[6] toolbox in ArcGIS Pro. The second approach is similar to how we integrate Linked Data into GIS. Figure 7 shows the Enrich layer toolbox of ArcGIS Pro together with data loaded from our connector interface using Wikidata. The output feature class from Figure 2 is used as the input as shown in Figure 7. Figure 8 shows the enriched attribute table as the result of executing the service. The main differences between the presented work and the Enrich layer are:

- As mentioned before, GeoEnrichment offers access to a closed and well-curated dataset

[4] https://developers.arcgis.com/rest/geoenrichment/api-reference/geoenrichment-service-overview.htm

[5] We will leave aside the (otherwise important) fact that our system is a prototypical implementation developed to showcase the need for and potential of a *deep* integration of Linked Data and GIS, while Esri's product is a mature and already deployed service.

[6] http://pro.arcgis.com/en/pro-app/tool-reference/analysis/enrich.htm

with predefined schema and attributes. This has a wide range of advantages such as controlled data quality, homogeneous data and schema, and so forth. Our work focuses on deeply integrating a distributed, Web-scale, real-time knowledge graph. Linked Data is not contradiction-free and the quality of the data may vary widely even within a particular subgraph. Put differently, Linked Data does not model facts but statements (Kuhn et al., 2014). The dense connections within and across the thousands of datahubs enable open-ended queries with an unparalleled spatial, thematic, and temporal coverage. For instance, our work can be used to do real-time sentiment analysis as a function of distance from some event site, e.g., oil spill, computing the trajectory similarity of famous expeditions, or mapping and analyzing the spatial distribution of authors of research articles about Alzheimer's disease by connecting to the IOS Press LD-Connect data hub (Mai et al., 2018). Therefore, we believe that GeoEnrichment and our work complement each other.

- The GeoEnrichment service can be used to define an area to be enriched. For example, given several point features, a 10-mile drive-time zone for each point can be defined around it and the demographic data (e.g. total households, total housing units, and population) within these areas can be calculated and appended to an attribute table. Basically, the (point) input features are converted to polygonal features for demographic data extraction and further analysis. In contrast, the position features (mouse clicking on the map) in our spatial range query toolbox are used to get nearby spatial entities. Since we find spatial entities, the attribute information for each retrieved entities can be extracted and appended to their attribute table without doing an area-weighting-based attribute calculation.

- The GeoEnrichment service supports datatype properties but not object type properties such as those linking actors, places, events, and objects together, which, in turn, is the most powerful functionality of our proposed framework.

- From a question answering perspective, our Linked Data connector framework is more suitable to answer geographic queries such as about the construction year of all major landmarks in New York City or for the oldest mission along California's coast. Interestingly, such functionality had not been the focus of GIS in general, and we hope that this will change in the future. This would allow to implement a seamless transition from geographic questions to the spatial analysis of the returned data. To stay with the mission example, a simple SPARQL query can reveal the oldest mission but nothing interesting can be done with the returned list (from a spatial analysis perspective, this is), and a simple GIS function can compute isochrone maps around these missions, but they are not readily available to a GIS user. It is the combination of both technologies that we see as the most promising path forward.

5 Conclusions and Future Work

In this work, we proposed and implemented a workflow to deeply integrate Linked Data and geographic information systems without simply flattening the retrieved data. We demonstrated how to connect to Linked Data from within a GIS, how to assist the user in loading attributes, converting and fusing them, as well as how to generate property path queries over object-type properties.

Under the hood, our system constantly creates new tables and schema for them, thereby enabling users to truly navigate the link structure of knowledge graphs and to query each node's datatype and object-type properties on-the-fly. We demonstrated how the resulting functionality is similar to Esri's recent GeoEnrichment while relying on a highly-heterogeneous, Web-scale, and open-ended knowledge graph instead of a curated dataset. Both data cultures are necessary, and, therefore, we believe that both approaches are complementary. More importantly, we demonstrated how our work enables path queries based on object-type properties, something that has not been possible to date. Such queries span multiple predicate-object pairs of either the same or different predicates. The resulting graph can either be used for further exploration, serve as (point) feature layer for spatial analysis, or be flattened, and, thereby, converted into non-spatial attribute data.

To give a final example that illustrates these options and the types of queries that our work enables, consider the birth places and states of US presidents. A path query would (starting from the United States) return all presidents, then their birth places, and then the states these places belong to. In the first case, the resulting tables could be used to further explore information about each birthplace, e.g., whether they are the locations of major companies or universities. In the second case, the spatial point patterns of the birth places and states could be used for spatial analysis. This, for instance, would reveal the uneven geographic distributions of the returned states (due to the historic westward growth of the US). In the third and final case, the user would flatten the attributes, thereby turning the individual places and states into counts and revealing that 8 US presidents were born in Virginia and seven in Ohio. Examples like these showcase the interaction of Linked Data and GIS as both these technologies could not have returned the (full) results alone.

In terms of future work, we see the presented research as a starting point towards a more question answering oriented view on GIS in which open-ended questions about geography and social processes can be approached and visualized from within a GIS. From a technical perspective, we have not addressed issues of scalability, user feedback and exception handling, provenance records, and so forth, but believe that they will be important steps towards turning the presented prototype into a deployable add-on to GIS systems including but not limited to ArcGIS.

References

Battle, R., Kolas, D., 2012. Enabling the geospatial semantic web with Parliament and GeoSPARQL. Semantic Web 3 (4), 355–370.

Berant, J., Chou, A., Frostig, R., Liang, P., 2013. Semantic Parsing on Freebase from Question-Answer Pairs. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. pp. 1533–1544.

Bizer, C., Seaborne, A., 2004. D2RQ-treating non-RDF databases as virtual RDF graphs. In: Proceedings of the 3rd international semantic web conference. Vol. 2004. Springer.

Brodt, A., Nicklas, D., Mitschang, B., 2010. Deep integration of spatial query processing into native RDF triple stores. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems. ACM, pp. 33–42.

Cyganiak, R., 2005. A relational algebra for SPARQL. Digital Media Systems Laboratory HP Laboratories Bristol. HPL-2005-170 , 35.

Diekhof, T., 2010. WRS - transparent encapsulation of semantic web reasoning services for use in spatial data infrastructures. Master's thesis, University of Münster.

Donaubauer, A., Straub, F., Schilcher, M., 2007. mdWFS: A concept of web-enabling semantic transformation. In: Proceedings of the 10th AGILE conference on geographic information science.

Fluid Operations, 2016. Fluidops information workbench. https://www.fluidops.com/en/products/information_workbench/.

Giannopoulos, G., Skoutas, D., Maroulis, T., Karagiannakis, N., Athanasiou, S., 2014. FAGI: a framework for fusing geospatial RDF data. In: OTM Confederated International Conferences" On the Move to Meaningful Internet Systems". Springer, pp. 553–561.

Haller, A., Janowicz, K., Cox, S. J., Lefrançois, M., Taylor, K., Le Phuoc, D., Lieberman, J., García-Castro, R., Atkinson, R., Stadler, C., 2019. The Modular SSN Ontology: A Joint W3C and OGC Standard Specifying the Semantics of Sensors, Observations, Sampling, and Actuation. Semantic Web 10 (1), 9–32.

Iwaniak, A., Kaczmarek, I., Strzelecki, M., Lukowicz, J., Jankowski, P., 2016. Enriching and improving the quality of Linked Data with GIS. Open Geosciences 8 (1), 323–336.

Janowicz, K., Schade, S., Bröring, A., Keßler, C., Maué, P., Stasch, C., 2010. Semantic enablement for spatial data infrastructures. Transactions in GIS 14 (2), 111–129.

Jones, J., Kuhn, W., Keßler, C., Scheider, S., 2014. Making the web of data available via web feature services. In: Connecting a Digital Europe Through Location and Place. Springer, pp. 341–361.

Koubarakis, M., Kyzirakos, K., 2010. Modeling and querying metadata in the semantic sensor web: The model stRDF and the query language stSPARQL. In: Extended Semantic Web Conference. Springer, pp. 425–439.

Kuhn, W., Kauppinen, T., Janowicz, K., 2014. Linked Data - A Paradigm Shift for Geographic Information Science. In: Proceedings of the 8th International Conference on Geographic Information Science. Springer, pp. 173–186.

Liang, C., Berant, J., Le, Q., Forbus, K. D., Lao, N., 2017. Neural Symbolic Machines: Learning Semantic Parsers on Freebase with Weak Supervision. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Vol. 1. pp. 23–33.

Liang, C., Norouzi, M., Berant, J., Le, Q. V., Lao, N., 2018. Memory augmented policy optimization for program synthesis and semantic parsing. In: Advances in Neural Information Processing Systems. pp. 10014–10026.

Mai, G., Janowicz, K., Hu, Y., McKenzie, G., 2016. A Linked Data Driven Visual Interface for the Multi-perspective Exploration of Data across Repositories. In: VOILA@ ISWC. pp. 93–101.

Mai, G., Janowicz, K., Yan, B., 2018. Combining text embedding and knowledge graph embedding techniques for academic search engines. In: 4th Workshop on Semantic Deep Learning (SemDeep-4).

Mai, G., Yan, B., Janowicz, K., Zhu, R., 2019. Relaxing unanswerable geographic questions using a spatially explicit knowledge graph embedding model. In: Proceedings of 22nd AGILE International Conference on Geographic Information Science, in press.

Ngomo, A.-C. N., Auer, S., 2011. Limes-a time-efficient approach for large-scale link discovery on the web of data. Proceedings of IJCAI 15, 3.

Olieman, A., Kamps, J., Claros, R. M., 2015. LocLinkVis: A geographic information retrieval-based system for large-scale exploratory search. arXiv preprint arXiv:1509.02010 .

Oracle, 2016. Spatial Resource Description Framework (RDF). http://docs.oracle.com/cd/B19306_01/appdev.102/b19307/sdo_rdf_concepts.htm.

Prudhommeaux, E., Bertails, A., 2008. A mapping of SPARQL onto conventional SQL. World Wide Web Consortium (W3C) .

Regalia, B., Janowicz, K., Gao, S., 2016. VOLT: A provenance-producing, transparent SPARQL proxy for the on-demand computation of Linked Data and its application to spatiotemporally dependent data. In: The Semantic Web. Latest Advances and New Domains. 13th International Conference, ESWC 2016. Springer, pp. 523–538.

Roman, D., Klien, E., 2009. SWING–a semantic framework for geospatial services. In: The Geospatial Web. Springer, pp. 229–234.

Roth, M., 2011. Geographic feature pipes. Ph.D. thesis, Diploma thesis, Institute for Geoinformatics, University of Münster, Germany.

Scheider, S., Ballatore, A., Lemmens, R., 2018. Finding and sharing GIS methods based on the questions they answer. International Journal of Digital Earth , 1–20.

Scheider, S., Degbelo, A., Lemmens, R., van Elzakker, C., Zimmerhof, P., Kostic, N., Jones, J., Banhatti, G., 2017. Exploratory querying of SPARQL endpoints in space and time. Semantic Web 8 (1), 65–86.

Stadler, C., Lehmann, J., Höffner, K., Auer, S., 2012. LinkedGeoData: A core for a web of spatial open data. Semantic Web 3 (4), 333–354.

Staub, P., 2007. A model-driven web feature service for enhanced semantic interoperability. OSGeo Journal 3 (1).

Vahedi, B., Kuhn, W., Ballatore, A., 2016. Question-based spatial computing - a case study. In: Geospatial Data in a Changing World. Springer, pp. 37–50.

van den Brink, L., Barnaghi, P., Tandy, J., Atemezing, G., Atkinson, R., Cochrane, B., Fathy, Y., Castro, R. G., Haller, A., Harth, A., et al., 2019. Best practices for publishing, retrieving, and using spatial data on the web. Semantic Web 10 (1), 95–114.

Yih, W.-t., Chang, M.-W., He, X., Gao, J., 2015. Semantic parsing via staged query graph generation: Question answering with knowledge base. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). Vol. 1. pp. 1321–1331.

Yih, W.-t., Richardson, M., Meek, C., Chang, M.-W., Suh, J., 2016. The value of semantic parse labeling for knowledge base question answering. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Vol. 2. pp. 201–206.

Yue, P., Di, L., Yang, W., Yu, G., Zhao, P., 2007. Semantics-based automatic composition of geospatial web service chains. Computers & Geosciences 33 (5), 649–665.

Zhu, R., Hu, Y., Janowicz, K., McKenzie, G., 2016. Spatial signatures for geographic feature types: examining gazetteer ontologies using spatial statistics. Transactions in GIS 20 (3), 333–355.

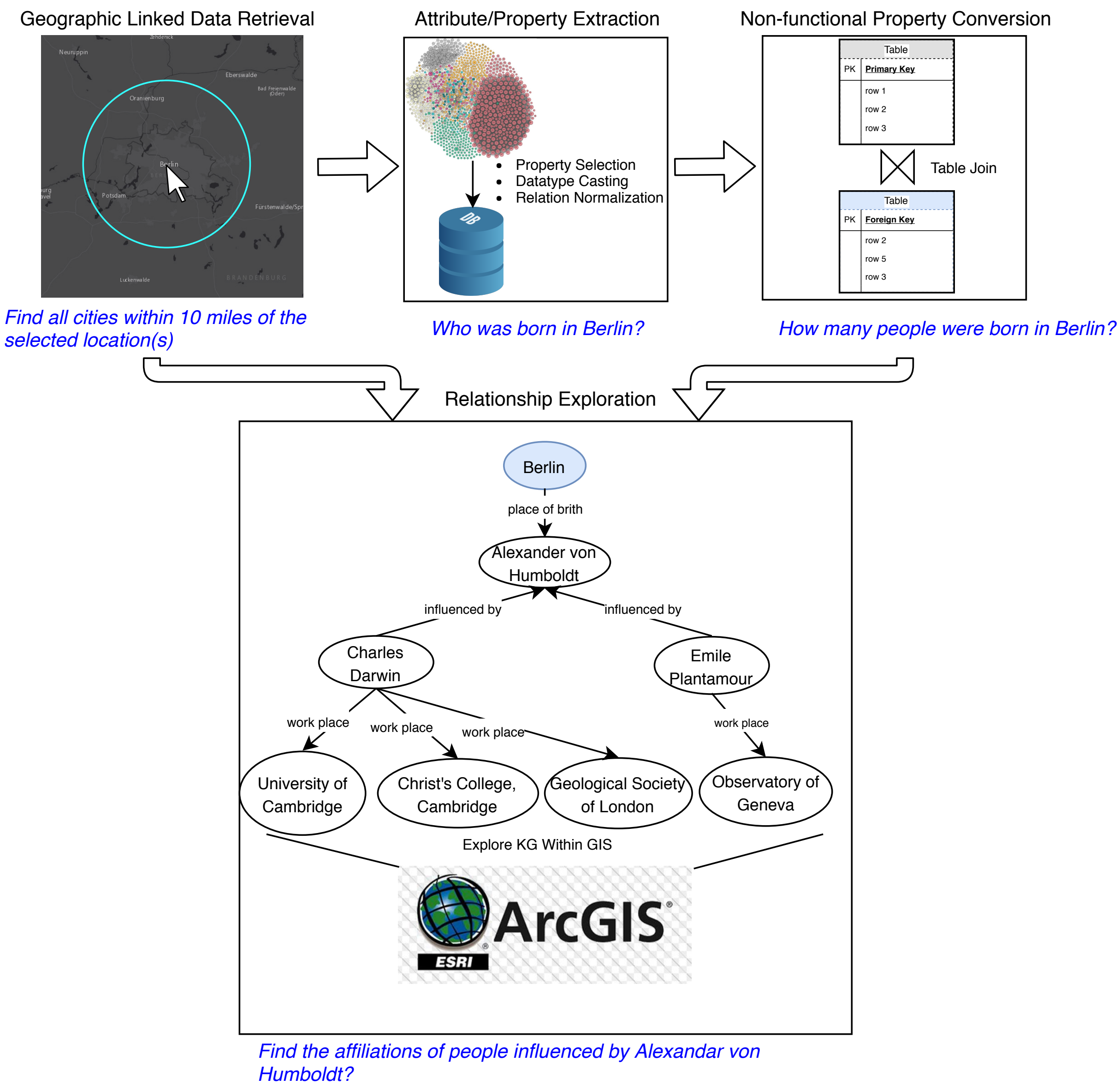


Figure 1: Basic workflow of the Linked Data connector framework.

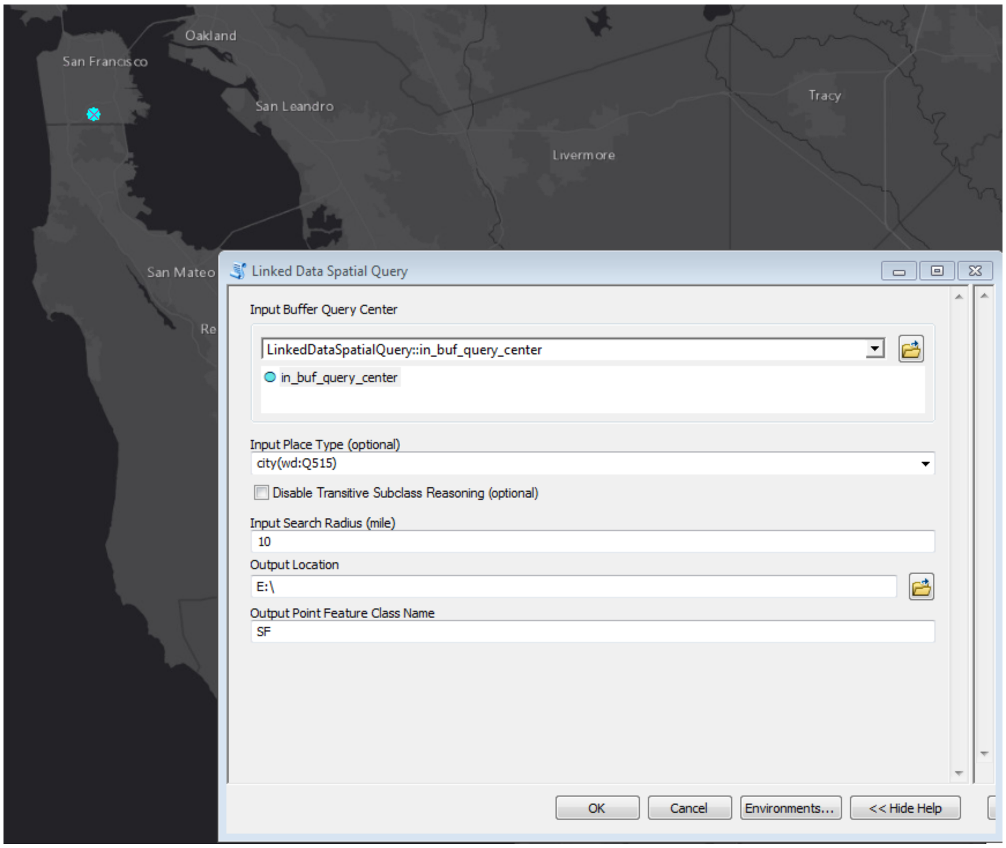


Figure 2: The Linked Data retrieval toolbox set to load entities of type `city` from Wikidata for a center point in California's bay area.

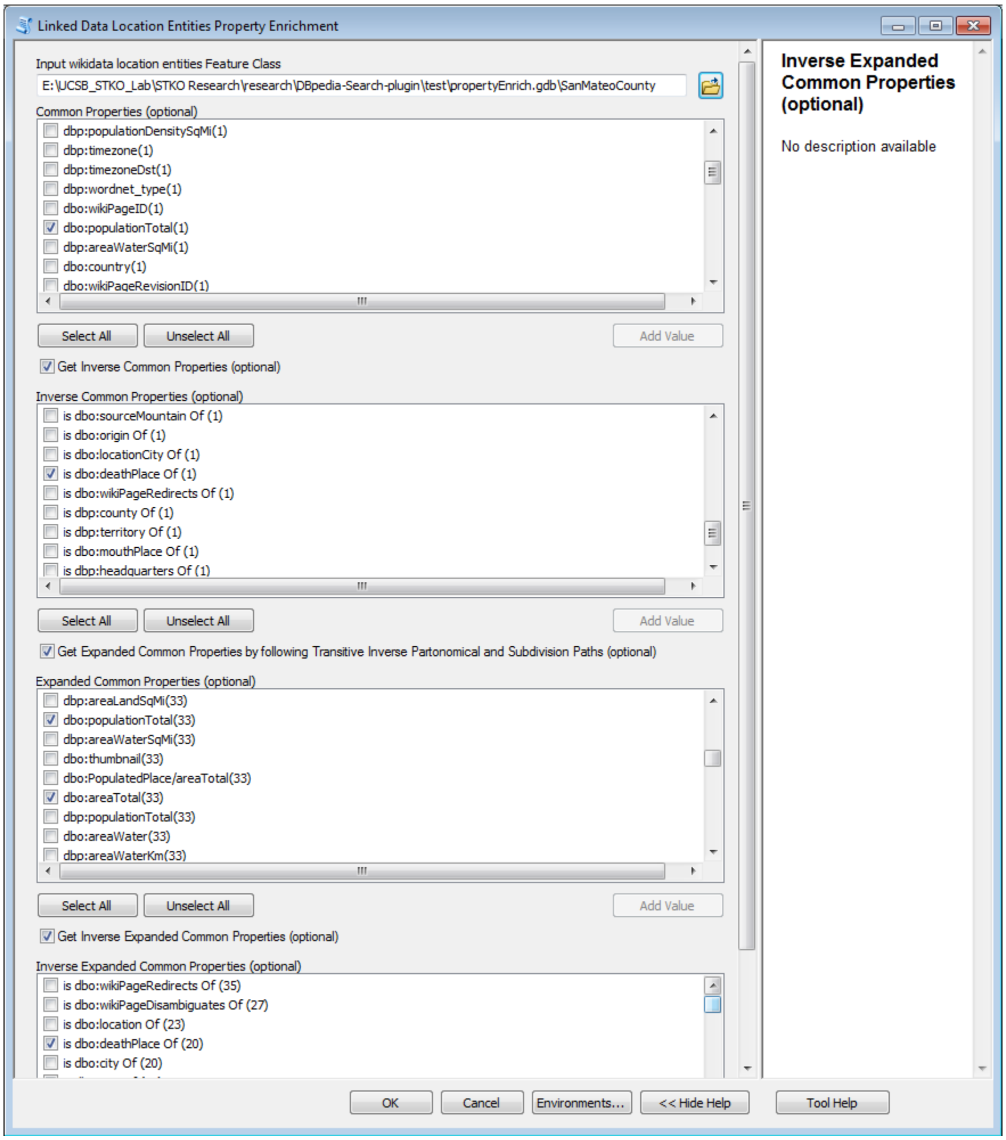


Figure 3: The Linked Data attribute extraction toolbox set to enrich the retrieved spatial entities with more attribute information.

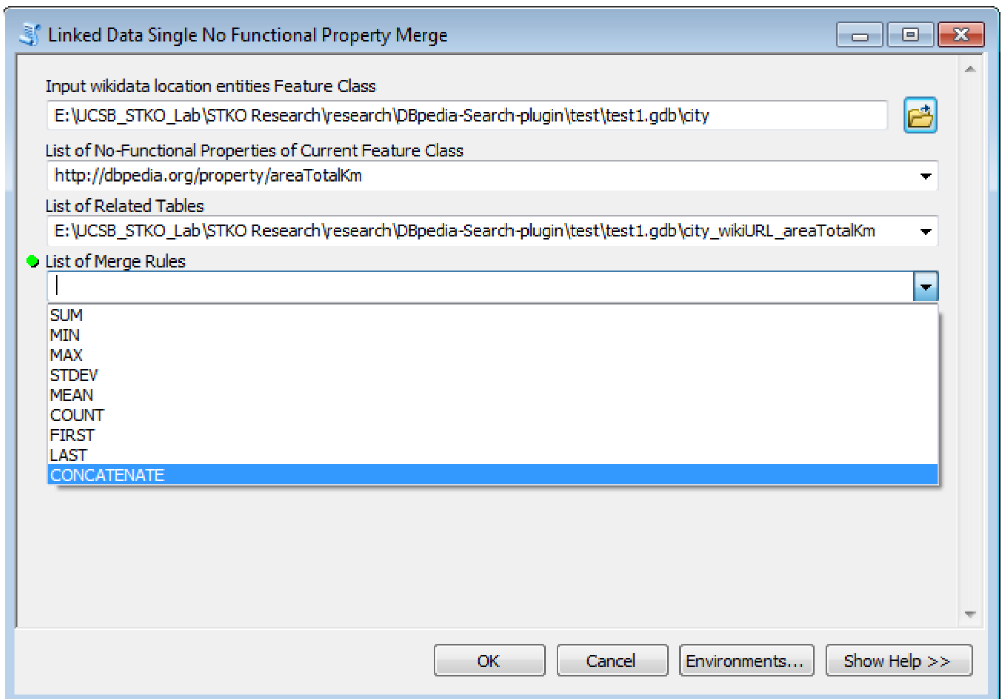


Figure 4: The non-functional property conversion toolbox to merge non-functional property values and append them to the main attribute table of spatial entities.

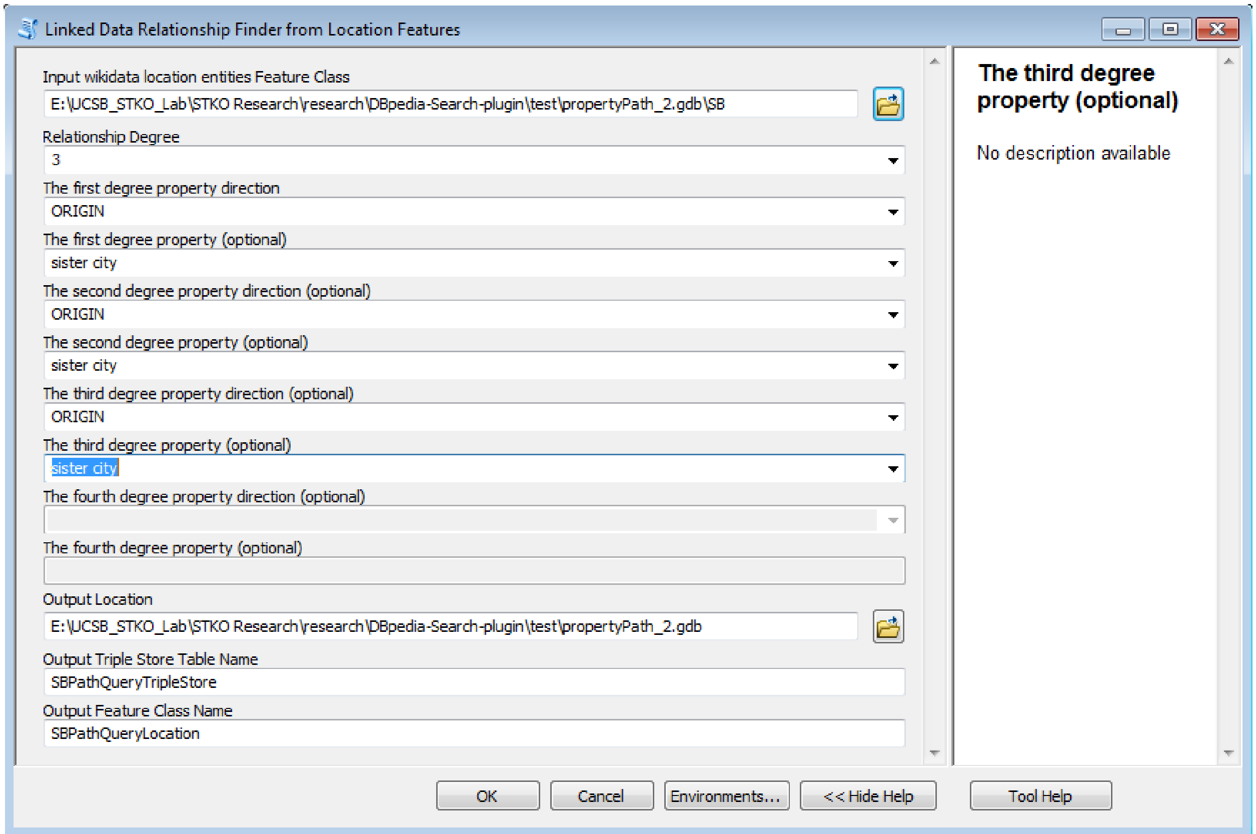


Figure 5: The relationship exploration toolbox to query for 1-, 2-, and 3-degree sister city from the input spatial entities (Santa Barbara).

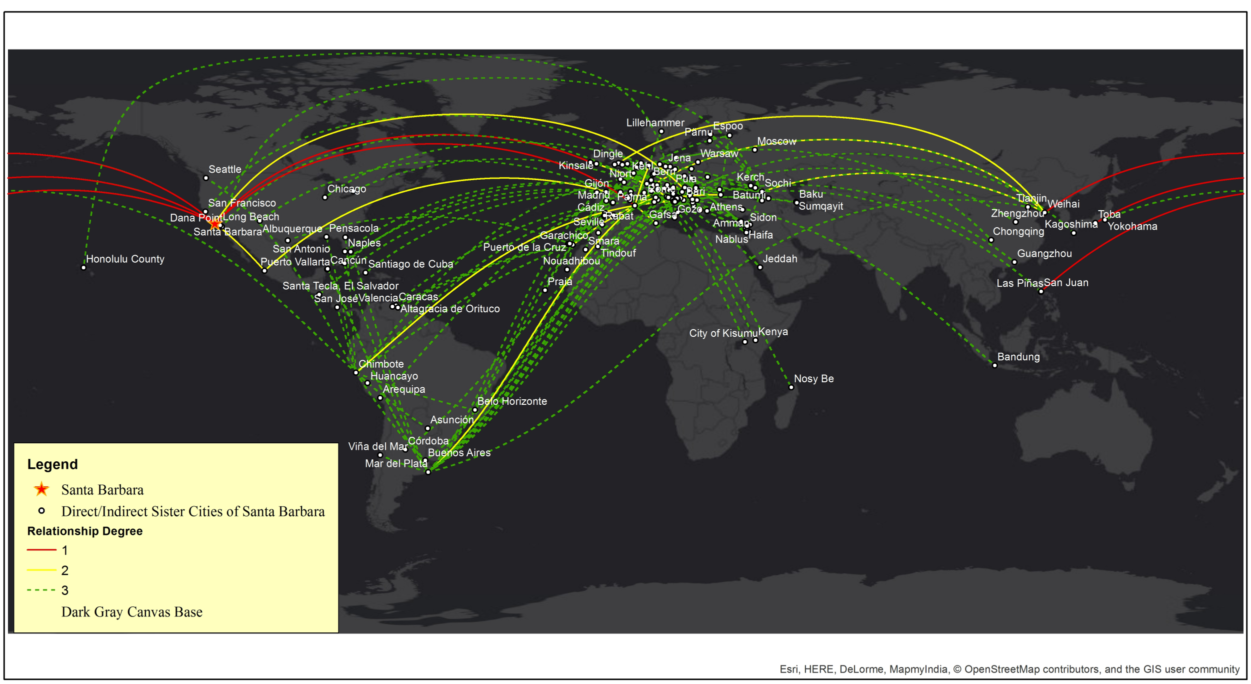


Figure 6: 1-degree, 2-degree, and 3-degree sister city relationship exploration example.

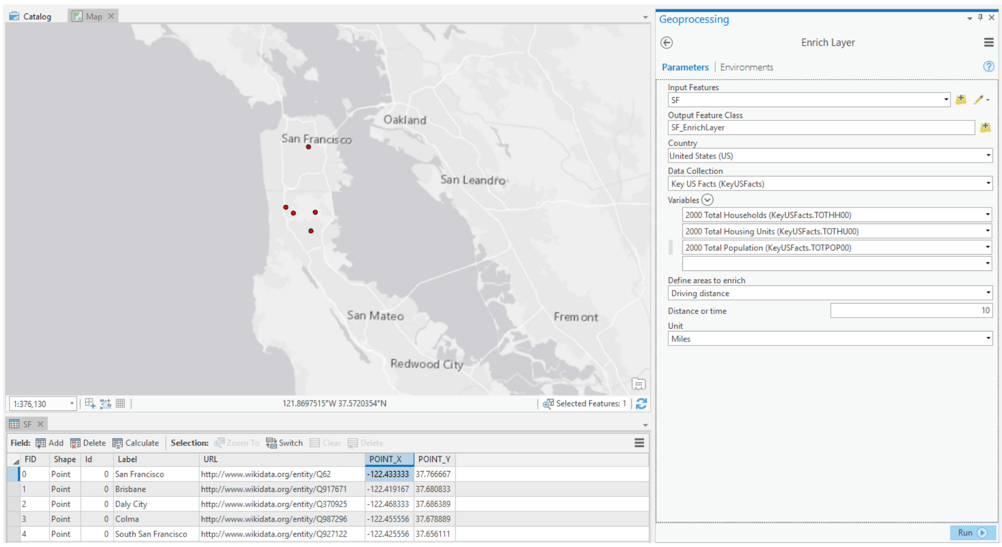
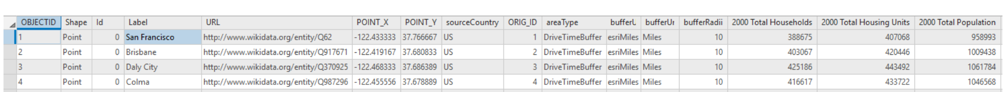


Figure 7: The Enrich Layer toolbox in ArcGIS Pro.



OBJECTID	Shape	Id	Label	URL	POINT_X	POINT_Y	sourceCountry	ORIG_ID	areaType	bufferU	bufferUr	bufferRadii	2000 Total Households	2000 Total Housing Units	2000 Total Population
1	Point	0	San Francisco	http://www.wikidata.org/entity/Q62	-122.433333	37.766667	US	1	DriveTimeBuffer	esriMiles	Miles	10	388675	407068	958993
2	Point	0	Brisbane	http://www.wikidata.org/entity/Q917671	-122.419167	37.680833	US	2	DriveTimeBuffer	esriMiles	Miles	10	403067	420446	1009438
3	Point	0	Daly City	http://www.wikidata.org/entity/Q370925	-122.468333	37.686389	US	3	DriveTimeBuffer	esriMiles	Miles	10	425186	443492	1061784
4	Point	0	Colma	http://www.wikidata.org/entity/Q987296	-122.455556	37.678889	US	4	DriveTimeBuffer	esriMiles	Miles	10	416617	433722	1046568
5	Point	0	South San Francisco	http://www.wikidata.org/entity/Q927122	-122.425556	37.656111	US	5	DriveTimeBuffer	esriMiles	Miles	10	275948	283756	777603

Figure 8: The enriched attribute table after executing the Enrich Layer toolbox in figure 7.