Big POI Data Integration with Linked Data Technologies

Spiros Athanasiou[1], Giorgos Giannopoulos[1], Damien Graux[2], Nikos Karagiannakis[1], Jens Lehmann[2,3]
Axel-Cyrille Ngonga Ngomo[4,5], Kostas Patroumpas[1], Mohamed Ahmed Sherif[4,5], Dimitrios Skoutas[1]
[1]Information Management Systems Institute, Athena Research Center, Greece
[2]Enterprise Information Systems Department, Fraunhofer IAIS, Germany
[3]Smart Data Analytics Group, University of Bonn, Germany
[4]Computing Center, Universität Leipzig, Germany
[5]Data Science Group, Universität Paderborn, Germany
{spathan,giann,nkaragiannakis,kpatro,dskoutas}@imis.athena-innovation.gr,
{Damien.Graux,Jens.Lehmann}@iais.fraunhofer.de,{mohamed.sherif,axel.ngonga}@upb.de

ABSTRACT

Point of Interest (POI) data constitutes the cornerstone in many modern applications. From navigation to social networks, tourism, and logistics, we use POI data to search, communicate, decide and plan our actions. POIs are semantically diverse and spatio-temporally evolving entities, having geographical, temporal, and thematic relations. Currently, integrating POI datasets to increase their coverage, timeliness, accuracy and value is a resource-intensive and mostly manual process, with no specialized software available to address the specific challenges of this task. In this paper, we present an integrated toolkit for transforming, linking, fusing and enriching POI data, and extracting additional value from them. In particular, we demonstrate how Linked Data technologies can address the limitations, gaps and challenges of the current landscape in Big POI data integration. We have built a prototype application that enables users to define, manage and execute scalable POI data integration workflows built on top of state-of-the-art software for geospatial Linked Data. This application abstracts and hides away the underlying complexity, automates quality-assured integration, scales efficiently for world-scale integration tasks, and lowers the entry barrier for end-users. Validated against real-world POI datasets in several application domains, our system has shown great potential to address the requirements and needs of cross-sector, cross-border and cross-lingual integration of Big POI data.

1 INTRODUCTION

Our daily lives evolve around locations. From navigation applications, to social networks, tourism, and logistics, we use information about locations to search, communicate, decide, and plan our actions. Selecting a location for a given activity has a varying complexity and significance, ranging from simple, everyday routines (e.g., where to have dinner), to more complex planning (e.g., where to open a shop), and to life-changing decisions (e.g., where to live, work or invest). Locations that exhibit a certain interest or serve a given purpose are commonly referred to as *Points of Interest* (POIs), covering anything from shops, restaurants or museums to ATMs or bus stops. POIs are complex entities that are characterized by their *geospatial shape* (points, lines, polygons) along with various *thematic attributes* indicating their name, type, functionality, services, etc., as well as their *relations* to each other (e.g., containment, part-of) with respect to spatial, temporal, and thematic contexts.

Creation, update, and provision of POI datasets is a multi-billion, cross-domain, and cross-border industry. Advances in the timely and accurate provision of POIs result into significant direct and indirect gains throughout our Digital Economy[1]. The value and impact of POIs is reflected in the complex, expensive and labor-intensive effort required for their production and maintenance, which inherently involves stakeholders and users throughout their value chain. Initial production involves field-work, constant monitoring for their evolution and accuracy, integration of user-feedback mechanisms for reporting errors, quality assurance of new data, and roll-out across a plethora of services and products. The greater the *size*, *timeliness*, *richness*, and *accuracy* of POI data, the better the product's *value*. Inversely, incomplete or inaccurate information has a profound effect on all types of end-users and applications.

The *value chain* of POI data has rapidly changed in the last few years. The advent of *open* data, *crowdsourcing*, and *social media* provides new data sources of even greater volume, heterogeneity, diversity, veracity, and timeliness. Such Big POI Data assets are harnessed by both startups and established commercial vendors alike to enrich their products, while also giving rise to new business models founded on domain-specific collection and provision of POIs (e.g., Foursquare[2], Yelp[3]). Enrichment, curation, and update of POI data is increasingly becoming collaborative, with stakeholders and end-users actively involved in all steps of the value chain. This intensifies the challenges relating to their quality-assured integration, enhancement, and sharing.

POI data is by nature *semantically diverse* and *spatiotemporally evolving*, representing different entities and associations depending on their geographical, temporal, and thematic context. Due to its use in various domains and contexts, POI information is typically fragmented across diverse, heterogeneous sources. These are common issues in Big Data integration [7], but combining and assembling POI information is further hindered in practice by the lack of common identifiers and data sharing formats. Even the way we typically identify and share information about POIs is inherently *ambiguous*. Addresses, coordinates, and place names are equally used throughout applications as pseudo-identifiers; but practice shows that they fail to effectively disambiguate POIs. Integrating POI data using current approaches remains labor-intensive and does not scale, thus limiting data coverage. The industry makes typically two compromises to reduce the complexity to feasible levels: focus on a specific domain (e.g. fuel stations), and/or restrict the spatial-, temporal-, and feature-space of data. In both cases, this leads to loss of information and thus lost value.

© 2019 Copyright held by the owner/author(s). Published in Proceedings of the 22nd International Conference on Extending Database Technology (EDBT), March 26-29, 2019, ISBN 978-3-89318-081-3 on OpenProceedings.org.
Distribution of this paper is permitted under the terms of the Creative Commons license CC-by-nc-nd 4.0.

[1] https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=COM:2018:0232:FIN
[2] https://enterprise.foursquare.com/products/places
[3] https://www.yelp.com

Series ISSN: 2367-2005

10.5441/002/edbt.2019.44

To tackle these challenges of Big POI data integration in the context of the SLIPO project[4], we transfer knowledge and apply state-of-the-art techniques and tools from the domains of Linked Data, Big Data and GIS. We argue that *Linked Data* technologies [2] are ideally suited to handle the inherent geospatial, thematic, and semantic ambiguities of POIs, without resorting to general-purpose entity matching platforms [16]. Recent advances in spatially-aware Linked Data technologies[5] address the scalability challenges of integrating, enriching, and querying semantically diverse geospatial Big Data assets and can effectively maximize the value extracted from open, crowdsourced and proprietary data sources.

The scope and ambition of our work in terms of complexity and coverage is inherently defined by current practices and needs in the industry. Indicatively, HERE Places API[6] offers information about 55 million POIs with names and categories in 237 countries; Google Places API[7] advertises over 150 million POIs globally; from OpenStreetMap, we have extracted more than 18.5 million POIs regarding specific categories[8]. Hence, we are targeting data integration concerning millions of POIs. We provide a complete suite of integrated software and services for POI data integration, supporting all stages of the POI data lifecycle (transformation, linking, fusion, enrichment). Our prototype application employs mature, scalable, open-source software specialized in geospatial linked data integration. We have tested its operation against several use cases for POIs in different application domains (geomarketing, tourism, navigation) with very encouraging results concerning execution cost and accuracy. Our experience shows that stakeholders can orchestrate those tools in coordinated, iterative *workflows* to progressively increase both the size and the quality of the integrated POI data.

The remainder of this paper is organized as follows. Section 2 presents the main challenges concerning Big POI data integration. Section 3 outlines the POI data integration lifecycle applied in SLIPO. Section 4 describes the SLIPO data model for representing information about POIs throughout the executed workflows. In Section 5, we explain the specific processes involved in the POI data integration process. Section 6 outlines the provided functionalities for extracting value-added analytics from the integrated POI datasets. Section 7 presents the current status of our prototype application. In Section 8, we report our experience from POI data integration scenarios in two real-world use cases. Finally, Section 9 summarizes the paper and discusses future work.

2 CHALLENGES IN POI DATA INTEGRATION

POIs are complex entities, described and associated with multifaceted and multi-modal information. They also exhibit complex (spatial, temporal, thematic) relationships, and they often have a long and complex lifespan. Any effective and systematic approach towards POI data integration needs to rely on robust, flexible and semantically-rich modeling of POI profiles and handling of POI identifiers, especially for applications dealing with cross-sector, cross-border, and cross-lingual content.

Consider a simple example regarding an imaginary POI. Suppose that the "Acropole Palace Hotel" was registered in 2015 at the local *Yellow Pages* directory and was assigned an identifier, along with some basic information (name, address, telephone, fax), as illustrated at the upper side in Figure 1. Later on, its address was geocoded and the resulting location coordinates were also included in the record to be used in a mobile city guide. At some point in time, this hotel was acquired by another company, which renovated and rebranded it to "Xenia Hotel", and also opened a restaurant at the roofgarden. This hotel is since listed in a *hotel reservation application* under a new identifier and with updated, multi-lingual information, which also features its current offers and services, as well as customer ratings (shown at the bottom side of Figure 1). Such changes need to be detected and included in the mobile city guide application, by matching and integrating new POI representations with old ones to avoid duplicates and to update obsolete information. In this example, such a matching (denoted with an `owl:sameAs` link) can be based on the phone number and the address of this POI (properties marked with red color in Figure 1). The challenge arises from the inherent *heterogeneity* in POI data characteristics. As shown in this example, this involves diverse attribute schemata, varying formats in values (e.g., in phone numbers) or different representations of the same information (e.g., address represented in a structured or unstructured format), which need to be resolved to determine whether these two records actually refer to the same POI.

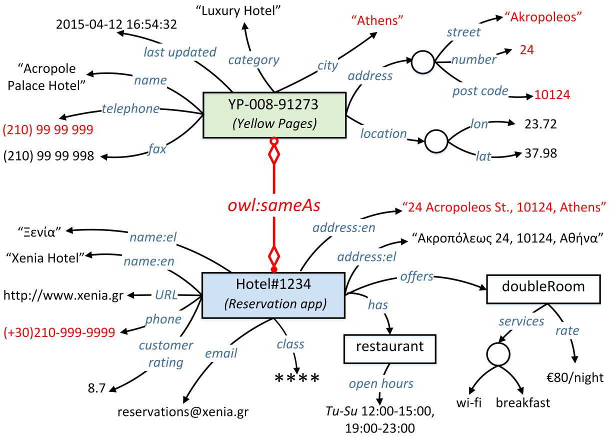


Figure 1: Matching different representations of the same POI coming from two data sources.

This example is only indicative of how different companies in different sectors and application domains collect, use, and extend information about the same POI. It also points out the importance of *volatile* data in the POI profile, e.g., its facilities, prices, events, etc. Moreover, it brings up several cases of *ambiguity* that arise and lead to data integration challenges that stakeholders have to face throughout this process. Next, we outline these challenging issues in POI data integration:

- *Lack of standardization.* Even though POI data is ubiquitous, there are no de jure standards yet for POI models, formats and identifiers. This is due to licensing and commercial competition, different hardware and software characteristics, and diverse requirements among mobile and web applications. Thus, POI datasets provided by different vendors are often not compatible with each other and require excessive effort and domain knowledge to be integrated and reused. At the most basic level, there are no (globally) *unique identifiers* assigned to POIs, making it difficult to identify duplicates among datasets and to link together different pieces of information for the same POI.

[4] Acronym for **S**calable **L**inking and **I**ntegration of big **PO**I data, http://slipo.eu/
[5] blog.geoknow.eu/the-linked-data-stack, http://aksw.org/Projects/LOD2.html
[6] https://developer.here.com/products/geocoding-and-search
[7] https://cloud.google.com/maps-platform/places/
[8] http://slipo.eu/?p=1397

- *Inherent ambiguity of POIs.* POIs are entities having a twofold nature, *geospatial* and *semantic*; moreover, their characteristics and associated information *evolves over time*. This results in multiple sources of ambiguity when dealing with POI data. In the spatial dimension, coordinates given for the same POI in different sources typically differ, while the spatial extent of a POI is often ignored, and its location is abstracted and approximated by a single point. Even if the shape is retained, it may have varying levels of accuracy. Hence, when encountering multiple POIs in different sources with slightly different coordinates or shapes, it is challenging to determine whether these refer to the same or different real-world entities. Similar issues arise when using POI names. The same POI may appear with slight naming variations in different sources, while different POIs may in fact have the same or similar names. Furthermore, different sources employ different classification or tagging schemes to categorize POIs and describe their type. This ambiguity is amplified when the temporal dimension is introduced, e.g., for determining whether two different representations refer to the same POI that evolved over time (as in Figure 1) or to two distinct POIs.
- *Long update cycles.* Due to the effort needed for maintaining and curating POI datasets, the contained information often remains relatively static. Further, it typically focuses on certain, mostly factual aspects of a POI, such as its title and a set of categories or tags. When and how this information is updated depends on the way it is collected and the available resources. Hence, for most providers, POI data is updated in yearly cycles. Moreover, if and when it is refreshed, typically the dataset will just be updated to the latest version, as it is not straightforward to apply a systematic and principled approach for recording and representing the *evolution* that has occurred, or more generally to track and record events that are related to that entity. Hence, even though such information may actually exist, there is often no historical profile of a POI that evolves over time and keeps track of associated events.
- *Fragmented POI profiles.* Based on how and for what purpose a POI dataset has been created, its contents typically cover only certain aspects of the POIs. For example, a navigation service and a city guide may have different priorities when deciding which POIs to include and what kind of information about them to collect. Although a wealth of information may exist for a POI, different parts and pieces may be found in different (types of) sources. Still, more complete POI profiles would allow more sophisticated and accurate analyses.
- *POIs treated independently and out of context.* POI datasets are typically treated as collections of individual entities. Each POI is modeled, stored and analyzed independently, without considering or establishing connections and links to other POIs. The reason is that *relationships* between POIs are more difficult to model, represent and analyze. However, this significantly limits the type of analyses that can be carried out over sets of POIs and creates gaps with the actual needs of users.

In SLIPO, our approach places particular emphasis on these issues related to POI models, identifiers, and resolving the aforementioned cases of ambiguity, as explained next.

3 THE POI DATA INTEGRATION LIFECYCLE

In this Section, we provide an overview of the POI data integration lifecycle supported in SLIPO. The underlying idea of our proposed system is to address the POI data integration challenges in the Linked Data domain [2]. Thanks to a simple, standardized, yet flexible model (*Resource Description Framework*[9]), Linked Data technologies can handle the inherent geospatial, thematic, and semantic ambiguities of POIs. Existing POI data assets need first to be transformed into RDF, so that individual POI profiles can be interlinked, fused, and enriched. This takes place in successive steps that progressively increase the size and/or the quality of POI data throughout a virtuous cycle, implementing an iterative workflow (Figure 2). Next, we outline the purpose of each stage, the processes that take place, and their output.

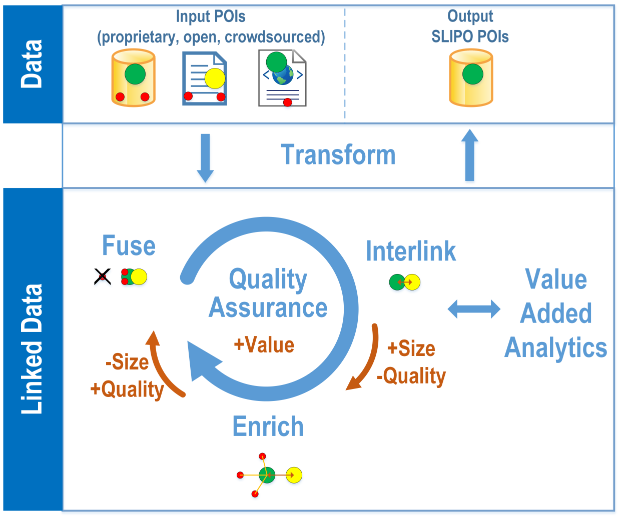


Figure 2: The POI data integration lifecycle.

The lifecycle begins with a *transformation* stage. This assumes as input POI data collected from heterogeneous and diverse data sources (proprietary, open, crowdsourced), having different attribute schemata and formats. The spatial, temporal, and thematic attributes in the input data are transformed into RDF triples conforming to a common, vendor-agnostic, well-defined, yet agile and extendable *POI ontology*. Hence, schema mappings from attributes of the original schemata to the classes and properties of this ontology are applied. After transformation, the resulting RDF triples can be stored in files or in an RDF store.

Subsequent stages are applied in the Linked Data domain against the previously transformed RDF data comprising an iterative, step-wise *workflow* that first increases the *size*, and then the *quality* of POIs. This forms a *virtuous cycle* that begins by expanding POI coverage, completeness, and richness, delivering data of greater size. Then, it focuses on increasing the quality of the POI data, fusing these intermediate results and enforcing appropriate quality assurance algorithms. This inherently reduces the size of data in absolute numbers, but increases their value. This process can be repeated in the same manner, iteratively increasing the size and then refining to increase quality, as many times as required. For example, an expert user can introduce additional data sources, apply different rules, focus on other types of metadata, etc. Such an iterative workflow involves the following stages:

- *Interlinking.* This is applied among transformed RDF datasets to link together individual RDF representations of the same real-world POI. It exploits structural properties, textual similarities, spatial proximity, etc., based on various user-specified

[9] https://www.w3.org/RDF/

metrics and thresholds. This deduplication process creates `owl:sameAs` links between matching POI entities, thus tackling the lack of common identifiers between POI entities across data sources, and enabling their management at later stages of the integration process.

- *Enrichment.* This step identifies and retrieves additional information from external sources that relates to the processed POIs, and creates extra properties for these POIs, increasing the richness and completeness of the data. It also discovers semantic relations between POI entities and other resources (e.g. areas, events, time), such as `partOf` relations (e.g., a school yard is part of a school etc.) or `occursAt` relations (e.g., events at a certain venue). Essentially, this phase enhances and contextualizes POI profiles with information extracted and assembled from other relevant sources, leading to a multi-faceted and more comprehensive view of POIs. For instance, it can enrich a POI profile with extra information concerning opening hours, price ranges, event timelines, etc.
- *Fusion.* This stage consolidates linked POIs and their properties. From two linked POI entities, it produces a unified and consolidated representation that is more complete, concise and accurate than the individual initial linked entities. For matching properties (e.g., similar names, nearby geometries) between a pair of linked entities, potentially different fusion actions can be applied. Fusion actions may include selection of properties from one of the entities, merging of properties from both entities, selection of those properties that satisfy a specific criterion (e.g., more complex, more timely), etc. Eventually, this leads to significant increase in completeness, coverage, timeliness and quality of POI data.
- *Value Added Analytics.* The POI data lifecycle includes an additional step, offering large-scale aggregation analytics in various dimensions, clustering, association rule mining, and predictive analytics. We can perform large-scale spatial, temporal, and thematic aggregation or extract *associations* between POIs and other entities. Results can be modelled as integrated sets or sequences of POIs. For instance, such a set can represent a thematic area of interest not explicitly defined in the original data, but implicitly discovered; this encapsulates a set of POIs that are spatially co-located and related thematically for direct exploitation in several use cases (e.g., geomarketing).

Quality assurance is performed throughout all steps of the lifecycle and ensures that each phase produces correct and accurate results, taking into account dataset-specific and use-case-specific quality indicators and rules, including also manual validation and authoring. Several indicators can be used, most of them already adopted by industrial vendors that manage and exploit POIs: size, timeliness, coverage, accuracy, etc.

Eventually, at the end of the POI data integration lifecycle, another transformation is also required. This involves the *reverse transformation* of the integrated POI data back to conventional formats (i.e., de facto POI formats), enabling their use in existing products, systems and services.

4 POI DATA MODEL

We have developed a comprehensive ontology for POI data to model and represent multi-faceted and enriched POI profiles and thus leverage Linked Data techniques and tools for POI data integration. This model accommodates and extends existing POI formats, providing a uniform and semantically rich model for

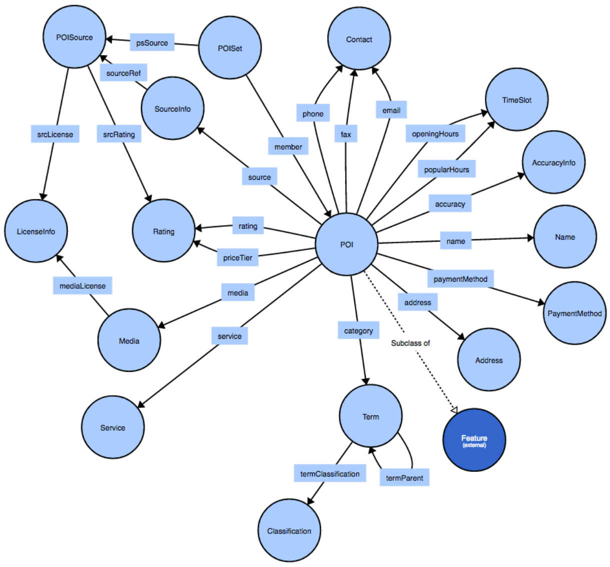


Figure 3: Main classes and properties in the POI ontology.

assembling and managing POI data from heterogeneous sources, since a widely accepted, de facto model is missing. To design this ontology, we have taken into consideration POI data representations employed by several well-known and widely used POI data sources. These included both open data sources, like OpenStreetMap and Wikimapia[10], as well as proprietary data schemata like Foursquare or Google Places. We also studied the structure of POI data assets based on samples obtained by commercial vendors in several countries (Austria, Germany, Greece). Finally, we investigated other efforts, especially an ongoing work on POI modelling by W3C and OGC Working Groups[11], as well as models for representing places proposed by collaborative projects[12].

The main challenge in POI data modelling arises from the high degree of heterogeneity across sources, given that these have different scope, goals and purposes. Thus, they model POIs from different perspectives and with varying levels of detail. Among the main issues we addressed with our POI data model are:

- *Names*: Accommodate the presence of different types of names (e.g., official vs. commonly used ones), alternate or historical names, acronyms, multi-lingual names, etc., as well as phonetic transcriptions or transliterations useful for internationalized POI information.
- *Addresses*: Represent addresses both in structured formats (separate components for country, city, street, house number, etc.) and flat formats (concatenated strings), with potentially missing information in either case.
- *Geometry*: Allow both for point geometries (longitude and latitude coordinates) and more complex geometries (lines, polygons), potentially in different coordinate reference systems.
- *Classification*: Support both hierarchical categorizations (i.e., categories, subcategories, etc.) and free tags.
- *Source* of POI data (e.g., a commercial vendor) may be specified using its title, homepage, description, license, logo, etc.
- *Media*: Support for auxiliary assets such as photos, videos or audio associated with a POI.

[10] http://wikimapia.org/api/

[11] http://www.opengeospatial.org/projects/groups/poiswg

[12] https://schema.org/Place

- *Contact information*: Include various elements, such as phone, email, fax, Web page, social media accounts, etc., with different structures and formats.
- *Amenities*: Include extra information on a variety of attributes such as offered services, opening and/or popular hours, ratings, payment methods, etc.

Figure 3 illustrates a graph with the main classes and object properties of our OWL ontology[13] as used internally in the POI data integration lifecycle. POI is the main class for representing POI features and is modelled as subclass of a spatial Feature in GeoSPARQL [24], thus directly inheriting properties regarding geospatial location. It supports multiple geometric representations, and the type of each geometry (e.g., centroid, navigation point, map pin, boundary) may be specified as well. Extra attributes for specialized use cases not covered by the ontology can be represented in the form of key/value pairs.

Overall, this ontology adheres to well-established standards (RDF, GeoSPARQL) and is geared towards processing efficiency. In addition, it can be extended and enhanced with domain-specific POI profiles, which include additional properties and relationships. We are currently working on modelling the provenance of POIs and their metadata, their evolution across time leading to different versions, as well as changes occurring to their contents and representation (e.g., extra attributes).

5 POI DATA INTEGRATION STEPS

SLIPO offers a complete and integrated suite of software tools that support all steps of the POI data integration lifecycle. All tools are available as open source software and are the state of the art in geospatial linked data integration. Specifically, the suite includes: (a) TripleGeo, for POI data and metadata *transformation* into RDF; (b) Limes, for *interlinking* of POIs; (c) Fagi, for the *fusion* of linked POI data into unified, concise and complete descriptions; and (d) Deer, for *enrichment* of POIs with implicit metadata, third party datasets and thematic, temporal and spatial metadata. Next, we describe them in more detail.

5.1 Transformation

Transformation is the entry point for POI datasets, converting them from their original format to RDF, thus enabling their subsequent processing in the Linked Data domain. We have extended our open-source *Extract-Transform-Load* (ETL) software TripleGeo [25] to enable scalable and efficient transformation of POI datasets from a variety of de facto geospatial formats into RDF triples. In TripleGeo[14], we employ adaptable, configurable, and reusable mappings from existing attribute schemata into our POI ontology (Section 4). We also support classification hierarchies for assigning categories to POIs. Moreover, TripleGeo can handle all common geometry data types and established coordinate reference systems. Its main features include:

- *Native support for a multitude of geospatial data formats.* Currently, TripleGeo supports 9 common file formats (e.g., ESRI shapefiles, GML, KML, CSV, JSON), and JDBC-based access to 8 geospatially-enabled DBMSs (e.g., Oracle Spatial, PostGIS).
- *Improved geospatial support* not only of primitive geometry types (points, linestrings, polygons), but also more complex geometries (MultiPolygons, Geometry Collections), as well as on-the-fly reprojection to another coordinate reference system.
- *User-defined mappings* specify rules that dictate how to generate RDF triples from original thematic attributes in the input POI data according to a given ontology. Such mappings allow transformation of all available attributes per POI entity and can be specified in the generic RDF Mapping Language RML [5, 6]. We also provide an alternative, simplified mapping facility specifically tailored to our ontology, offering much faster transformation even for very large volumes of POI data.
- *Classification schemes.* POI data providers employ diverse classification or tagging schemes to categorize POIs and describe their type. TripleGeo accepts specification of (possibly hierarchical) classification schemes for POIs, produces RDF triples that fully describe this information along with especially assigned URIs, and introduces extra links between a POI and its respective category under this scheme.
- *Customized URIs.* We construct HTTP Universal Resource Identifiers for POIs based on automatically generated Universally Unique Identifiers (UUIDs). This follows recommended patterns and best practices for creating persistent, unique, vendor and technology independent URIs. Thus, POI data owners have enough flexibility and control over creating and managing their own POI identifiers, while still adhering to a uniform format.
- *Reverse transformation* from RDF to all common geographical file formats. POI data that have been interlinked, fused, and enriched in previous executions of the POI data integration lifecycle can become accessible and exploitable by existing software (e.g., DBMS, GIS) or services (e.g., web mapping, route planning) commonly utilized in the industry.
- *Comprehensive configuration.* Users can control various parameters of the transformation process at different levels of complexity based on their technical background and expertise.
- *Compliance to standards.* The produced RDF geometries are fully compliant with the OGC GeoSPARQL standard for RDF spatial entities [24]. Transformation of INSPIRE-aligned data and metadata [10] is also supported [26], thus abiding by the EU Directive for interoperable Spatial Data Infrastructures.
- *Scalability.* Our experiments with various data sources show that TripleGeo is currently orders of magnitude faster compared to its original release [25], as well as faster than GeoTriples [17]. It can efficiently transform millions of POIs even without any sophisticated data partitioning schemes. Indicatively, it can transform all 7.4 million POIs extracted from OpenStreetMap for Europe into RDF triples in less than 3 minutes, effectively generating more than 715,000 triples/sec.

Using TripleGeo, we have launched a free download service[15] that offers global POI data extracted from OpenStreetMap and transformed into RDF format, retaining all original tags and also creating resolvable, machine-readable URIs per POI.

We are currently working on extending TripleGeo towards *semi-automatic workflows* to assist and guide users in creating attribute mappings for new datasets. We have built a utility that employs Machine Learning to learn new mappings from a corpus of previously specified ones, available from the various use cases of POI datasets we have handled so far. This utility also analyzes the contents of each attribute in a new POI dataset, based on its data type (string, numeric, etc.), formatting (e.g., phone numbers, postal codes), as well as the presence of special characters. Users can then verify or modify the automatically suggested mappings through a graphical interface before applying them for transforming their POI data into RDF.

[13] OWL ontology is available at https://github.com/SLIPO-EU/poi-data-model

[14] https://github.com/SLIPO-EU/TripleGeo

[15] http://download.slipo.eu/results/osm-to-rdf/

5.2 Interlinking

The aim of interlinking POI datasets is to develop scalable approaches for integrating massive heterogeneous, and incomplete POI data at a world-scale. Limes[16] is integrated in SLIPO and incorporates many algorithms for performing efficient interlinking among POI resources. In the context of SLIPO, Limes receives as input two RDF POI datasets conforming to the SLIPO ontology. Thus, Limes's input POI data are first transformed by TripleGeo, into the proper RDF format and schema. Further, apart from the two input POI datasets, Limes requires as input a configuration file containing the Limes configuration parameters. Limes's output consists in a single file, which contains the links between corresponding POI entities from both input POI datasets. The output of Limes is essential for running other SLIPO tools in a POI data integration cycle (Figure 2).

Limes v1.0.0 is the first version of Limes that has been developed in the context of the SLIPO project and focuses on POI-specific interlinking. One of the major goals was to abstract as much complexity as possible from the end users. So, in order to keep user interaction at a minimum and requiring no knowledge of Linked Data technologies and concepts, we aimed at adapting and fine-tuning Limes's functionality specifically for POI data, as well as at automating the interlinking process as much as possible. To this end, we emphasized on the development of the backend of the platform, aiming to enrich and specialize the core interlinking functionality of the framework. Next, we outline the new features and functionality of Limes:

- *POI-specific point-set distances.* New point-set distances based on the vector representations of the POI resources (e.g. *Hausdorff, mean, surjection* and *sumOfMin*). Altogether, we implemented a set of 10 point-set distance functions based on our survey published on [30].
- *Topological relation discovery* based on the vector representations of the POI resources (e.g. a POI resource contains, crosses or touches another POI resource). For instance, find all parking locations within shopping malls. Therefore, we develop Radon [8, 29], an efficient algorithm for rapid discovery of topological relations among POI resources with 2D geometries.
- *Temporal relation discovery* based on timestamps associated with the POI resources (e.g., one POI takes place after, before or during another POI). For example, a specific area is used as a parking location only during a football match. To tackle this problem, we proposed Aegle [13], a novel approach for efficient computation of links between POIs' temporal representations according to *Allen*'s interval algebra.
- *Combining the new techniques with the ones already in Limes.* In Limes v.1.0.0 we integrated the novel algorithms for the 10 POI-point-set distances as well as Radon and Aegle into the Limes core. In particular, a new mapper is implemented for each of the new relation types. Such mappers are combined with the already existing mappers for efficient link discovery of the new types of relation specific for POI resources.
- *Novel machine learning approaches for POI interlinking.* In most cases, finding a good metric expression[17] (i.e. one that achieves high F-Measure in interlinking POI entities) is not a trivial task. Therefore, in Limes we implemented Wombat, a novel machine learning approach for auto-generation of mappings among POI resources. Wombat is inspired by the concept of generalisation in quasi-ordered spaces [28]. Wombat minimizes the Limes configuration task by providing unsupervised, supervised and active learning versions.
- *Integration with the SLIPO Workbench.* Limes v.1.0.0 realizes two deployment modes: (a) standalone, as an individual software that accepts as input linked POI datasets and provides as output a mapping file containing the links between the input POI datasets; (b) deployment within the SLIPO Workbench, where Limes serves as an integral component of the SLIPO Toolkit and is loosely integrated by the SLIPO Workbench with the other software components into forming POI integration workflows.
- *Scalability.* In addition to the original Limes parallelization algorithm [31] and optimized planners [12, 22], in Limes v1.0.0 we evaluated the scalability of our novel POI-specific approaches. Our evaluation proves that Radon [8, 29] is able to outperform state-of-the-art approaches up to 3 orders of magnitude while maintaining a precision and a recall of one. Also, our evaluations of the runtime of Aegle [13] show that Aegle outperforms the state of the art by up to 4 orders of magnitude while maintaining a precision and a recall of one. Recently, we have implemented a simple, yet efficient in our setting, distributed execution scheme, which functions independently of core interlinking in Limes. Specifically, we have two implementations based on Spark and Flink frameworks. Currently, we run an intensive evaluation for both frameworks to find the pros and cons of each. Finally, we studied the effect of geometry simplification on the scalability of POI interlinking [1]. We found that a suitable simplification setting can reduce interlinking cost with a minimum effect on quality.

5.3 Fusion

The fusion process in SLIPO follows the interlinking of different representations of the same POI across data sources. Fusion addresses the problem of assembling partial and incomplete POI profiles as well as resolving conflicting information in order to derive a more complete, consolidated profile per POI.

Fusion receives as input two POI datasets as well as a set of links between them. The output is a third merged dataset, which contains consolidated descriptions of the linked POIs. Each POI in the fused dataset is described by a set of richer, non-redundant, non-conflicting and complete properties, which have been derived by merging the initial descriptions of the linked POIs. The main challenge in this task is to efficiently apply the most appropriate fusion action in such way that the best elements of individual datasets are kept in the final composite dataset.

To support scalable and quality assured fusion of large POI datasets, we have extended our fusion framework Fagi [14]. Initially, Fagi was a map-based, user-interactive platform for manually performing property matching and fusion actions on individual properties of linked geospatial entities. In SLIPO, we adapted Fagi[18] to effectively handle the fusion of POI data, while we also minimized manual user effort. Specifically, the current version offers the following main features:

- *Advanced fusion facilities for POIs.* Fagi supports the graphical authoring of POI fusion specifications. These sets of rules examine the individual properties of pairs of linked POIs and decide, for each property, the most fitting fusion action. Fagi currently incorporates 25 *condition functions* for examining the properties of linked POIs, including string similarity, geometry comparison, etc. Further, it implements 15 *fusion actions*

[16] https://github.com/dice-group/LIMES

[17] A metric expression is a logical expression that describes when two resources should be linked.

[18] https://github.com/SLIPO-EU/FAGI

(regarding both thematic and geospatial properties) for deciding how to merge the values of matching properties. Fusion actions include: aligning geometries, maintaining the most complex value, maintaining both values for the same property, etc. Finally, combining several condition functions can be used to construct more elaborate fusion rules, while fusion results can also be marked as *ambiguous* for later inspection by the end user.

- *Link validation functionality.* An important aspect of quality assurance lies in validating the fusion input and deciding whether the linked entities should be either fused, further examined, or rejected as erroneous. To this end, in Fagi we define a set of validation actions, as well as a validation rule specification scheme. Similarly to POI fusion specifications, the user can define elaborate link validation specifications that jointly examine several properties of pairs of linked POIs in order to maintain or reject the specific linked POIs.
- *Quality indicators extraction.* Further emphasizing on quality assurance, Fagi supports the extraction of more than 25 quality indicators. The user is able to review several statistics on the input linked POI datasets before performing fusion on them (*pre-fusion statistics*), as well as on the output fused data (*post-fusion statistics*). The former provide an overview of the data at hand, which assists the integrator to properly define and configure the validation/fusion rules. The latter assist the user in the inspection of the fusion results, and potentially guide her into re-configuring and re-executing the fusion process.
- *Recommendation of link validation and fusion actions.* Fagi implements learning mechanisms for training on past user actions and recommending link validation and fusion actions for new POIs. It learns binary (for link validation) and multi-class (for fusion actions) classifiers on a series of extracted training features regarding the properties of the linked POIs. Then, it recommends actions for new pairs of linked POIs.

The aforementioned functionality of Fagi satisfies commercial-level data fusion needs. In a typical fusion scenario, the user can define configurable and re-usable fusion rule specifications of varying complexity, which collect names from different datasets in multiple languages or types (such as official, international, brand-names etc.) and complete other attributes, such as address information, websites, phone numbers, emails, ratings, reviews, opening hours, image links, etc. The resulting fused dataset then contains POIs with the most complete and accurate descriptions, as well as more precise and/or more complex geometries. Additionally, the user is able to examine a plethora of quality indicators and use them to assess and potentially improve the quality of the fusion process.

Fagi v2.0 focuses on satisfying the effectiveness and performance requirements of fusing Big RDF POI data. Thus, an important effort has been made to fine-tune the underlying algorithms in order to increase their efficiency and scalability. The performance of the current implementation of Fagi is tested against real-world commercial POI datasets, by applying a custom partitioning and distributed processing scheme that occupies 10 nodes and takes less than five minutes to fuse 1 million linked POIs, which corresponds to a country-level fusion process.

5.4 Enrichment

Enrichment is one of the main parts of the data integration process. In SLIPO, enrichment focuses on POI entities that are characterized by a set of major properties (e.g. name, coordinates and category) as well as potentially several additional properties (e.g. address, telephone, email, rating, etc.). Enrichment considers one or more input dataset(s) containing POIs. The goal of enrichment is to produce one or more enriched dataset(s), containing better descriptions of the input POIs based on information retrieved from external, third-party RDF data sources (e.g., SPARQL endpoints, DBpedia). That is, each POI entity in the final, enriched dataset must be described by a set of RDF triples that have been derived by merging the initial description for the POI with those generated via various enrichment operations. Note that some enrichment approaches can define a set of triples to be removed from the original POI descriptions. Those removed set of triples are either wrong or inaccurate. The enrichment process can also replace inaccurate triples with ones with correct values. Considering the big picture of the POI integration lifecycle (Figure 2), the enrichment process is tightly interconnected with validation and quality assurance. To this end, the enrichment process needs to incorporate several mechanisms to assess the quality of the proposed enrichment operations and their results.

The enrichment task is carried out via our generic enrichment component Deer[19] [27]. Deer incorporates many approaches for performing efficient enrichment among POI resources. In the context of SLIPO, Deer receives as input one or more RDF POI dataset(s) conforming to the SLIPO ontology. Thus, Deer's input POI data are first transformed by TripleGeo into the proper RDF format and schema. Moreover, Deer input datasets may be linked via Limes prior to be enriched by Deer. Further, Deer requires as input a configuration file containing its configuration parameters. Deer's output consists of one or more files that contain the enriched versions of the respective input datasets.

Deer offers a set of *enrichment operators*, i.e., artifacts in charge of enriching input POI dataset(s). The input for such an enrichment operator is a set of one or more datasets. The output is also a set of one or more enriched datasets. In the following, we highlight some of Deer's enrichment operators:

- *The Dereferencing enrichment operator.* For POI datasets which contain similarity proprieties links (e.g. `owl:sameAs` to DBpedia resources), we deference all links from our source dataset to other datasets (e.g., DBpedia) by using a content negotiation on HTTP. The returned set of triples needs to be filtered for relevant POI resources. Here, we use a predefined list of attributes of interest. Among others, we look for `geo:lat, geo:long, geo:lat_long, geo:line` and `geo:polygon`. This list can be reconfigured via Deer's configuration.
- *The NLP Enrichment Operator* enriches POI resources by extracting embedded POI information hidden within the datatype properties and making it explicit as new triples added to the original POI dataset. For example, find all POIs embedded within the description of all hotel POIs and add them as new triple to the respective hotel POI. The current version of Deer uses the Fox [33] framework for Named Entity Recognition (NER). By default, Deer extracts the POIs based on DBpedia as the background knowledge base. As Deer is a generic POI enrichment framework, the used NER framework can be configured as well as the used background knowledge base.
- *The Geo-Distance Enrichment Operator* aims to enrich a set of POI pairs (not necessarily of the same type) with the great elliptic distance between them. For example, the geo-distance operator can enrich all hotel POIs by adding the distance to the nearest POIs of bus stations/parking lots/hospitals.

[19] https://github.com/dice-group/DEER

5.5 Quality Assurance

Each SLIPO component provides a collection of several quality indicators and statistics. They all produce verbose *execution metadata* that can either be visualized or downloaded for further inspection by the end user. Particular effort has been put in Limes, Deer, and Fagi for POI linking, enrichment and fusion respectively. In particular, both Limes and Deer implement a series of quantitative quality indicators, such as *run-time*, *number of added triples* and *percentage of data increase after linking/enrichment*. Further, both Limes and Deer provide the qualitative quality indicators of *precision*, *recall*, and *F-measure* in cases where benchmark datasets are available. In case no benchmark dasasets are available, Limes is still able to provide the *pseudo-precision*, *pseudo-recall*, and *pseudo-F-measure* first introduced in [23]. The basic assumption behind these pseudo measures is that symmetrical one-to-one links exist between the resources in source and target datasets. Our pseudo-precision computes the fraction of links that stand for one-to-one links and is equivalent to the strength function presented in [15]. The pseudo-recall computes the fraction of the total number of resources (i.e. from both source and target datasets) that are involved in at least one link. Finally, the pseudo-F-measure is the harmonic mean of pseudo-precision and pseudo-recall.

Fagi implements a series of quality indicators (e.g., percentages of fused properties vs. initial POIs/initial links, average numbers of POI property completeness) that compare the linked POI datasets and the resulting fused POIs. In particular, *attribute gain* indicates the percentage of extra properties compared to the original (e.g., a gain of 0.4 on a given POI means it was complemented with 40% additional attribute values). *Confidence* indicates the degree of similarity (in names, geometry, phone number, etc.) between the original features that were fused into a unified one, with values close to 1 indicating almost perfect match. These indicators are utilized both internally in Fagi (similarity measures, learning mechanisms) and as output for the end user, for further inspection and manual validation of fused POIs.

6 VALUE-ADDED POI ANALYTICS

At the end of the POI data integration workflow, various services are provided to perform advanced analytics and extract added value from POIs. Currently available functionality includes *Best Region Search* and extraction of *Areas of Interest*, which can be further modeled using techniques such as LDA or semantic clustering. We describe these functionalities in more detail below.

6.1 Best Region Search

Given a set of POIs $\mathcal{D}$, an $\alpha \times \beta$ rectangle R, and a utility score function $f : \mathcal{P} \rightarrow \mathbb{R}$ that assigns an objective score to any subset $\mathcal{P} \subseteq \mathcal{D}$, the goal of the *Best Region Search* problem is to find the optimal placement of R over the space containing $\mathcal{D}$ such that the value of f over the enclosed subset of POIs $\mathcal{P}$ is maximized [11]. This problem has many applications in various domains, from geomarketing and tourism to real estate and urban planning, facilitating decisions about, e.g., selecting the best location to open a new store or to place an advertisement.

However, the existing state-of-the-art algorithm for this problem [11] only computes the best, i.e., the top-1 result. This is usually not sufficient in practice. For instance, it may not be possible to open a store at the identified best location (no available facilities to rent or purchase), or all hotels in the identified best area may be occupied or too expensive. Then, the user needs to examine alternative solutions in decreasing order of quality, until one is found that meets all desired criteria.

To address this shortcoming, we have introduced the k-Best Region Search (k-BRS) problem, which computes a ranked list of the top-k best regions according to the utility score function. The main challenge in doing so, is that by simply returning a top-k list of results ordered by their objective score, typically produces highly overlapping results. Instead, our proposed algorithm is able not only to compute top-k results progressively, but also to diversify the returned results by either minimizing or completely excluding overlap among them. This is achieved by progressively retrieving subsequent results beyond the top-1, and selecting the next best candidate based on their *marginal gain*, i.e., the added value of each new result in the context of those already selected. A detailed description of the algorithm can be found in [32].

6.2 Extracting and Modelling Areas of Interest

Another functionality for POI data analytics involves the application of spatial clustering to extract Areas of Interest (AOIs) from the integrated and enriched POI dataset and then the use of topic modelling techniques to characterize and compare those areas.

For the first step, we employ *density-based clustering* to identify areas with high concentration of POIs (e.g., shopping malls, transportation hubs, touristic attractions, etc.). We use the DBSCAN [9] or HDBSCAN [4] algorithms for this purpose. DBSCAN can identify clusters of arbitrary shapes. Moreover, it does not require the user to specify the desired number of clusters in advance; instead, the user specifies two parameters that control the density. HDBSCAN extends DBSCAN by offering a mechanism to adjust the density automatically based on the data distribution.

The resulting clusters comprise a set of AOIs but provide no additional information regarding their nature or characteristics. They merely point out that these areas have a higher concentration of POIs compared to the rest of the space. To provide further insights to the user as to what these AOIs are about and how they can be compared to each other, we employ *topic modelling*. Essentially, this draws inspiration from extracting a set of topics over a document collection and representing each document according to those topics.

In our case, we represent each AOI as a “document”, with the contents of the document being a bag-of-words representation of the union (multiset) of terms appearing in the POIs contained in that AOI. These terms may refer to POI categories or tags, or any other terms extracted from POI names, descriptions, reviews, etc. We then perform Latent Dirichlet Allocation (LDA) [3] over this “document” collection. The basic idea behind LDA is that each document can be described by a distribution of topics and each topic can be described by a distribution of words. The result of the process comprises two matrices. The first is a topic-terms matrix that defines each extracted topic as a distribution over POI terms. The second is an AOI-topics matrix that represents each AOI as a distribution over the identified topics.

This approach is quite flexible in practice because it allows to model each AOI as a mixture of topics, instead of assigning each AOI to a single category. This better reflects the fact that typically a region contains a mixture of different types of POIs and serves a mixture of purposes, rather than a single one.

The resulting AOI-topics matrix can be used to compare AOIs to each other, e.g., find AOIs with similar mixture of topics, providing a means to quantify these similarities and differences.

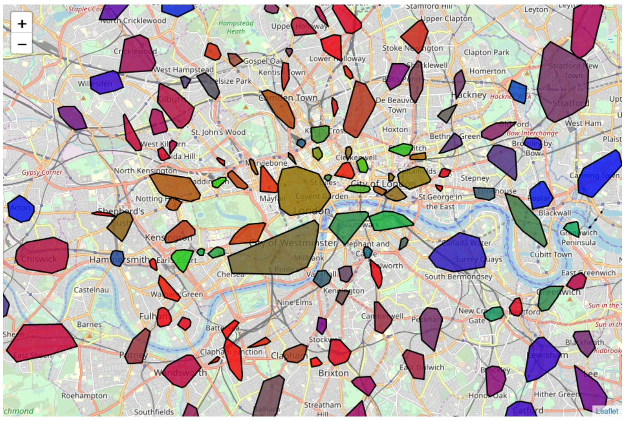


Figure 4: Example of extracted AOIs in London, with colors determined via topic modelling.

Moreover, it allows for intuitive visualizations of the extracted AOIs based on their topic distributions. An illustrative example is provided in Figure 4, where LDA was used to extract 3 topics for a given set of AOIs in London, and then each AOI was assigned a color by determining the RGB values based on its mixture of these 3 topics. Such visualizations allow not only to depict where AOIs are located, but also to intuitively identify similarities and differences between AOIs according to the underlying mixture of different types and attributes of POIs they contain.

6.3 Implicit POI Clustering

We have also developed strategies to group POIs together according to thematic, contextual and/or temporal considerations. For that purpose, we have integrated into SLIPO's toolkit the Power Iteration [19] and K-Means [20] clustering algorithms into SANSA[20] [18] since this semantic-web open-source stack provides these algorithms for RDF data out-of-the-box. We first process the input RDF data using SANSA and then apply the clustering methods. To facilitate the explanation of the algorithms, we use the following example data as input and give the corresponding results. Let's consider three POIs and their associated categories: $P_1(A, B, C)$, $P_2(A, C, D)$ and $P_3(A, B, D)$.

Power Iteration Clustering (PIC). In general, given a set of data points, we could represent the similarity between each pair of data points with a similarity matrix. For example, we could use Jaccard Similarity between two sets of categories that belong to corresponding POIs to represent the similarity between them. Therefore, a similarity matrix S, a diagonal matrix D and a Laplacian matrix $L = D - S$ are created. In Spectral Clustering [35], a subspace matrix consisting of eigenvectors corresponding to the smallest K eigenvalues was derived from the Laplacian matrix. The subspace matrix indicates the clustering results with K clusters. In PIC, the subspace matrix is an approximation to an eigenvalue-weighted linear combination of all the eigenvectors of a normalized similarity matrix [19]. The subspace matrix also indicates the clustering results. In order to prepare the correct input to the algorithm, we first collect the categories for each POI, and then we compute the Jaccard Similarity between category sets for each pair of POIs. Afterwards, we construct a similarity matrix which could be used by PIC algorithm.

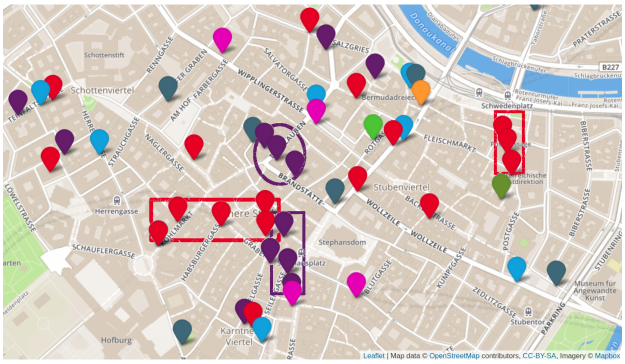


Figure 5: Example of AOIs in Vienna obtained after thematic and spatial clusterings.

K-means. This algorithm partitions POIs into different clusters such that POIs within each cluster have the smallest distance to the cluster centroid compared to placing them into any other cluster. K-means requires a distance metric to represent the distance between POIs. We use the following:

- One *hot encoding* converts categorical values into numerical vectors. Going back to our example, P_1 would be encoded into $(1, 1, 1, 0)$ since it belongs to categories A, B, C and not D.
- *Multidimensional scaling* [34] maps a distance matrix to some vectors in certain dimensions. The relative distances between different POIs are kept.
- *Word2Vec* [21] creates vector representations of words in a text corpus, which here concerns the set of all categories.

Since these methods do not consider the location of the POIs but only focus on implicit links, they produce thematic groups of POIs which are not necessarily geographically close. Indeed, since the groups are computed using categories, clusters might contain POIs sharing the exact same set of categories but actually located in different regions. For instance, Figure 5 represents the results when running PIC over POIs in Vienna. Pins having the same color are part of the same cluster. As expected, it appears that POIs of the same thematic cluster are distributed at various parts of the city center. Then, to get usable AOIs, we pipe together the two considered kinds of clustering to sub-group the thematic clusters according to their geographical locations. Thanks to that strategy, we ensure that the resulting AOIs are composed by POIs that are thematically related. This example demonstrates that the combination of these two approaches allows us to refine the thematic clusters and thus obtain the four AOIs depicted on map.

7 PROTOTYPE IMPLEMENTATION

We have been implementing a comprehensive open source software prototype aiming to support stakeholders in all stages of the POI data value chain. This prototype integrates all aforementioned tools for transforming, linking, fusing, enriching, and analyzing linked POI data. The SLIPO system (currently in beta version[21]) consists of the following main modules:

- *SLIPO Toolkit*: This is a collection of the individual software components (Section 5) applied in quality-assured POI data integration: transformation (TRIPLEGEO), interlinking (LIMES), fusion (FAGI), enrichment (DEER) and analytics (SANSA). These software components can be either installed locally or invoked

[20] https://github.com/SANSA-Stack

[21] All software is publicly available at https://github.com/SLIPO-EU

as part of the SLIPO Workbench and APIs functionality explained next.

- *SLIPO Workbench*: This is a web application, which integrates the Toolkit components to implement POI data integration workflows in a coherent, simple to use, and flexible manner. More specifically, it provides utilities for (a) uploading, searching and managing POI datasets in several formats, (b) designing, persisting and managing data integration workflows for POI datasets based on the features provided by the SLIPO Toolkit, (c) scheduling and monitoring the execution of the data integration workflows, and (d) visualizing the results of workflow executions.
- *SLIPO APIs*: This is a collection of RESTful HTTP programming interfaces for invoking SLIPO Toolkit component functionality and integrating it into third-party systems. APIs only support the invocation of simple atomic functions (e.g., POI transformation); otherwise the Workbench web application should be used. Both SLIPO Workbench and APIs are exposed through the same web application server.

Our prototype implements a *workflow engine* that executes data integration jobs and a *scheduler* for initializing workflow executions. A workflow consists of several loosely coupled tasks that together constitute a data integration process over POI datasets. A task may invoke an operation implemented by a Toolkit component (e.g., fusion), or perform secondary operations (e.g., prepare configuration files, update metadata, copy files).

The workflow engine and the SLIPO Toolkit components are deployed over a cloud infrastructure. Workbench and APIs exchange messages with the scheduler to execute workflows. The scheduler propagates requests to the workflow engine, which subsequently initiates the execution of one or more tasks. A task is executed either in-process locally on the scheduler host, or remotely using Docker containers. Each Toolkit component is responsible for providing a scalable implementation for the requested operation, inside the context of the running OS process. A Toolkit component that advertises itself as capable of partitioning its input (and, of course, merging its output) can also scale to multiple Docker containers. The scheduler only controls the total amount of resources allocated to a container, enforcing CPU/memory quotas derived from component-specific requirements and input data size.

Thanks to its modular, service-oriented architecture, SLIPO offers stakeholders the option to directly use the provided functionalities following a *Software-as-a-Service* paradigm. Alternatively, they are able to select specific tools to customize, extend and incorporate in their own POI data management workflows according to their specific needs and requirements. We expect that this will allow the rapid uptake of our innovations in a production setting without affecting any operations and processes already in place.

8 USE CASES

We have been extensively testing and evaluating SLIPO in real-world settings, against various POI data assets. These use cases cover diverse domains (geomarketing, tourism, navigation), ensuring that they reflect the requirements of cross-sector, cross-border and cross-lingual POI data integration. Next, we examine the ability of SLIPO to cope with two typical data integration scenarios against real-world POI datasets in two countries.

Table 1: POI datasets tested in the use cases.

	Dataset	*#POIs*	*Geometry*	*Thematic attributes (# in bold)*	*#triples*
Germany	D_1 (vendor)	35640	point	**(14)**: name(s), category, address, contact details	1114598
	D_2 (vendor)	24416	point	**(13)**: name, address, business data (turnover, #employees, etc.)	1098755
	OSM (open)	45750	point, line, polygon	> **25** *tags*: (multi-lingual) names, address, contact details, image, opening hours, operator, etc.	1130220
	GN (open)	7156	point	**(12):** name(s), city, zipcode, text description, last update	161473
Greece	D_3 (vendor)	72373	point	**(13):** bi-lingual names, address, category, contact details	2517030
	OSM (open)	102159	point, line, polygon	> **25** *tags*: (multi-lingual) names, address, contact details, image, opening hours, operator, etc.	2515476

8.1 Validation Settings

The first use case concerns *hotel POIs* in Germany, whereas the second deals with *general POIs* in Greece. Table 1 lists information concerning the datasets in each scenario. Note that data sources in each use case have different schemata, content and quality. Some datasets are crowdsourced such as OpenStreetMap (*OSM*) or GeoNames (*GN*), while others are offered by commercial vendors (D_1, D_2, D_3)[22]. Through SLIPO, we define integration workflows that deliver an output dataset having:

- *More POIs*, i.e., POIs missing from an original dataset are complemented from the other ones.
- Geometry representations get a *more detailed shape*, e.g., polygons obtained from *OSM* can replace (or complement) the original point (lat/lon) locations of certain POIs.
- *Extra thematic attributes* are derived by bringing together information (e.g., fax numbers, opening hours, links to photos, multi-lingual names) across all original data sources.
- Attribute values per POI are *more accurate* and complete, e.g., missing telephone numbers are filled or updated after checking against each available input.

In each use case, the original datasets are first transformed into RDF according to the SLIPO ontology (Section 4) with suitable attribute mappings. The last column in Table 1 indicates the number of resulting RDF triples. Next, each data *integration cycle* handles a pair of RDF datasets, either transformed from original data or intermediate ones derived from previous cycles. As mentioned in Section 3, a cycle involves these successive stages (cf. Figure 2):

- *Linking* two POI datasets to identify matching POI entities. The resulting RDF graph contains `owl:sameAs` links between their respective URIs.
- *Fusion* of the linked datasets into a new one according to several strategies for fusing spatial and thematic properties per POI.
- *Enrichment* of fused data with extra information from DBpedia.

Once data integration is complete, its RDF output passes through our *reverse transformation* service and delivers the integrated dataset in a traditional POI format (e.g., CSV, shapefile) readily expoitable by stakeholders.

Both scenarios were executed on a virtual machine deployed on top of a cloud stack. This VM offers an Intel® Xeon® E-52600 CPU with 16 virtual cores, 32GB RAM, 16GB swap space, and 200GB disk running Linux Ubuntu 16.04 LTS. In each case, we report data sizes, as well as the qualitative measures discussed in Section 5.5. All tests have been conducted with "cold" caches.

[22] Commercial vendors are anonymized for confidentiality.

8.2 Use Case A: Hotels in Germany

The first use case aims to integrate POIs in Germany concerning *hotels*. We assume that a stakeholder maintains a base POI dataset (D_1) and wishes to enrich it with information from other datasets (Table 1) in three successive integration cycles:

(#1) Integrate D_1 with *OSM*;
(#2) Integrate result of cycle #1 with D_2;
(#3) Integrate result of cycle #2 with *GN*.

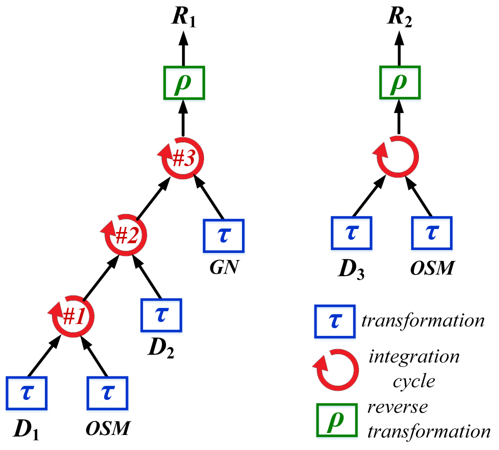
This workflow is illustrated in Figure 6(a). It includes transformation of each input dataset as well as the successive integration cycles (linking, fusion, enrichment). After reverse transformation of the integrated output, the resulting dataset R_1 is obtained. Under this scenario, the stakeholder does not wish to increase the number of POIs in its base data D_1. Instead, it only wants to enhance its content with extra thematic attributes, by filling in missing values and also obtain more detailed geometry representation where available from the other datasets.

Indicative quality and performance measurements for each cycle of Use Case A are listed in Table 2. In this use case, as all data belongs to a specific category (*hotels*), we specified that links between POI profiles should be based strictly on their spatial proximity. This explains the high linking confidence (0.98) in the resulting links. Although this choice can yield erroneous matches (e.g., two hotels may be close, but have different names), we sort them out later during fusion. As no ground truth is available, precision and recall concerning link detection cannot be estimated by Limes. In Table 2, we only provide the respective pseudo-measures (Section 5.5), but they yield rather poor estimates because the source/target datasets involved in linking at each cycle have different sizes (cf. Table 1).

Thanks to its validation rules, Fagi can filter out mismatches not only based on proximity, but also checking with POI names, phone numbers and addresses. The confidence of fused results remains consistently high across all three cycles and demonstrates the similarity among the original POIs that were fused together. At the end of the fusion process, we were able to achieve an increase in the amount of properties per POI, which at the final cycle exceeds 26% on average. Note that there are some POIs where the amount of their attributes increases by up to 47%, acquiring extra properties progressively in each cycle. The more the fused attributes, the faster the confidence stabilizes after each subsequent cycle close to 0.87. It is also important to mention that about a quarter of the POIs get more detailed geometry representations due to integration with the *OSM* data in Cycle #1. Since all other datasets include points only, no further geometric improvement occurs in subsequent cycles. Finally, we stress that the entire workflow concludes in about 6.5 minutes, delivering a unified, richer dataset that otherwise would require considerable human labor.

8.3 Use Case B: POIs in Greece

The second use case concerns general POIs in Greece of *various categories*, i.e., not only hotels as in the previous use case, but also restaurants, cinemas, schools, supermarkets, bus stops, ATMs, etc. As illustrated in Figure 6(b), the goal in this workflow (in one cycle only) is to create a single, integrated dataset R_2 that includes all available information from both input datasets (commercial D_3, open *OSM* data). In particular, apart from richer content (geometries, extra thematic attributes, and filled missing values), the resulting dataset R_2 will grow in size as well, containing many more POIs than any of the original ones.



(a) Hotels in Germany (b) POIs in Greece

Figure 6: Integration workflows for the two use cases.

Table 2: Execution Results.

Measurement	Use Case A			Use Case B
	Cycle #1	*Cycle #2*	*Cycle #3*	
# detected links	33645	15605	6448	29353
Pseudo-precision	0.69	0.81	0.61	0.65
Pseudo-recall	0.57	0.43	0.18	0.22
Pseudo-F-measure	0.62	0.56	0.28	0.33
Avg linking confidence	0.98	0.98	0.98	0.79
# fused pairs	19885	10903	1790	11036
Avg fusion confidence	0.91	0.88	0.87	0.86
Avg attribute gain	0.20	0.26	0.26	0.17
Max attribute gain	0.41	0.47	0.47	0.37
# resulting POIs	35640	35640	35640	159099
# non-point geometries	8728	8728	8728	28236
Execution cost (sec)	224.8	141.4	26.5	823.4

Statistics regarding this workflow are also listed in Table 2. In this use case, interlinking is not just based on spatial proximity but also on POI name similarity in order to avoid matching of possibly dissimilar entities that are next to each other in densely populated areas (e.g., city centers). Due to our relaxed criteria for interlinking, enough potential matches were detected (29353), although with less linking confidence (average: 0.79) compared to the previous use case.

However, we observed that a POI was often linked to multiple other POIs not always having a very similar name, but still close enough in space. During the fusion stage, most of those links were ignored based on our validation rules that also take into consideration more properties (phone, address) in similarity checks. Keeping only 11036 links that were deemed reliable, we derived fused POIs achieving strong confidence (0.86) that they actually concern the same entity. Regarding attribute gain, integrated results denote an average 17% increase in properties per POI. This result may seem poorer compared to Use Case A, but note that crowdsourced content in *OSM* for Greece is less rich than for Germany. Still, the most important outcome of this integration is that the final dataset R_2 includes more than 159 thousand POIs, which is over than double the size of commercial dataset D_3. As

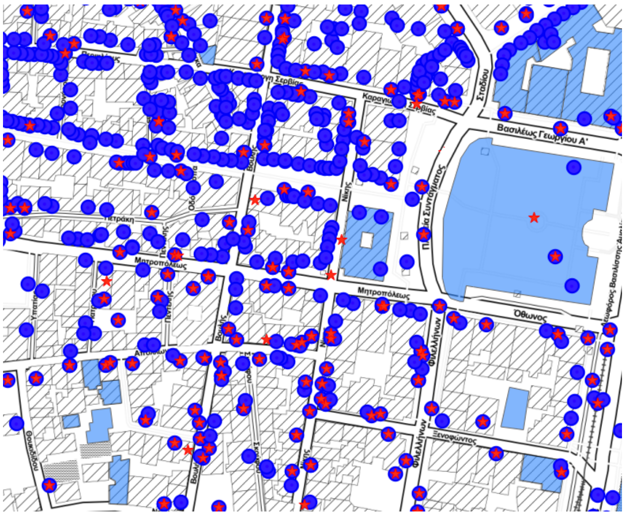


Figure 7: POIs in Athens city center before (red) and after integration (blue).

depicted in Figure 7, integration results (POIs in blue circles) supersede by far and drastically enhance the original information of dataset D_3 (shown with red stars). Furthermore, almost 18% of the resulting POIs now have more detailed geometries (shown as blue polygons) thanks to information extracted from *OSM*. Last, but not least, the fact that this workflow delivers its result in less than 14 minutes, clearly demonstrates the efficiency of SLIPO.

9 CONCLUSIONS AND FUTURE WORK

In this paper, we presented the SLIPO system, a cloud-based application encapsulating Linked Data technologies to efficiently address the challenges of large-scale integration of POI data assets. SLIPO transfers data integration to the Linked Data domain, thus allowing state-of-the-art software to be repurposed and focused to POI data, without requiring domain-specific knowledge from stakeholders or alterations in existing operational workflows. Our tests and evaluations in diverse application domains have shown that SLIPO offers clear advantages in terms of efficient, reliable, quality-assured POI data integration.

Our effort in SLIPO continues along several directions aiming to expand its relevance, efficiency, and value in an industrial setting. First, we will improve the individual software components with additional POI-specific rules and operations to increase performance and effectiveness. Further, we are working with our industrial partners to apply SLIPO in a plethora of domain-specific and cross-border commercial data integration tasks, directly comparing and documenting the gains in productivity, time-to-market, and value. In addition, we are creating periodic world-scale data integration workflows beyond the current reach of the industry, to enable low-cost and streamlined POI-based services. Finally, we are expanding the interoperability of the system to support third-party systems (e.g., signage recognition from street-level imagery) and its quality-assurance services, which will help embed SLIPO in the business workflows of most stakeholders in the value chain.

ACKNOWLEDGMENTS

This work was partially funded by the EU H2020 project SLIPO (#731581).

REFERENCES

[1] A. F. Ahmed, M. A. Sherif, and A.-C. N. Ngomo. On the effect of geometries simplification on geo-spatial link discovery. In *SEMANTiCS*, 2018.

[2] C. Bizer, T. Heath, and T. Berners-Lee. Linked Data – the story so far. *Int. J. Semantic Web Inf. Syst.*, 5(3):1–22, 2009.

[3] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, 2003.

[4] R. J. G. B. Campello, D. Moulavi, and J. Sander. Density-based clustering based on hierarchical density estimates. In *PAKDD*, pages 160–172, 2013.

[5] A. Dimou, D. Kontokostas, M. Freudenberg, R. Verborgh, J. Lehmann, E. Mannens, S. Hellmann, and R. Van de Walle. Assessing and Refining Mappings to RDF to Improve Dataset Quality. In *ISWC*, pages 133–149, 2015.

[6] A. Dimou, M. V. Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. V. de Walle. RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In *LDOW*, 2014.

[7] X. L. Dong and D. Srivastava. *Big Data Integration*. Synthesis Lectures on Data Management. Morgan & Claypool Publishers, 2015.

[8] K. Dreßler, M. A. Sherif, and A.-C. Ngonga Ngomo. RADON results for OAEI 2017. In *Proceedings of Ontology Matching Workshop*, 2017.

[9] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*, pages 226–231, 1996.

[10] European Commission. INSPIRE Directive 2007/2/EC – Infrastructure for spatial information in Europe. https://inspire.ec.europa.eu/, 2007.

[11] K. Feng, G. Cong, S. S. Bhowmick, W. Peng, and C. Miao. Towards best region search for data exploration. In *SIGMOD*, pages 1055–1070, 2016.

[12] K. Georgala, D. Obraczka, and A.-C. Ngonga-Ngomo. Dynamic planning for link discovery. In *ESWC*, 2018.

[13] K. Georgala, M. A. Sherif, and A.-C. N. Ngomo. An efficient approach for the generation of allen relations. In *ECAI*, 2016.

[14] G. Giannopoulos, N. Vitsas, N. Karagiannakis, D. Skoutas, and S. Athanasiou. FAGI-gis: A tool for fusing geospatial RDF data. In *ESWC Satellite Events*, 2015.

[15] O. Hassanzadeh, K. Q. Pu, S. H. Yeganeh, R. J. Miller, L. Popa, M. A. Hernández, and H. Ho. Discovering linkage points over web data. *Proc. VLDB Endow.*, 6(6):445–456, 2013.

[16] P. Konda, S. Das, P. Suganthan G. C., A. Doan, A. Ardalan, J. R. Ballard, H. Li, F. Panahi, H. Zhang, J. Naughton, S. Prasad, G. Krishnan, R. Deep, and V. Raghavendra. Magellan: Toward building entity matching management systems. *Proc. VLDB Endow.*, 9(12):1197–1208, 2016.

[17] K. Kyzirakos, D. Savva, I. Vlachopoulos, A. Vasileiou, N. Karalis, M. Koubarakis, and S. Manegold. GeoTriples: Transforming geospatial data into RDF graphs using R2RML and RML mappings. *J. Web Sem.*, 52-53:16 – 32, 2018.

[18] J. Lehmann, G. Sejdiu, L. Bühmann, P. Westphal, C. Stadler, I. Ermilov, S. Bin, N. Chakraborty, M. Saleem, A.-C. Ngonga Ngomo, and H. Jabeen. Distributed semantic analytics using the SANSA stack. In *ISWC Resources Track*, 2017.

[19] F. Lin and W. W. Cohen. Power iteration clustering. In *ICML*, pages 655–662, 2010.

[20] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297, 1967.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[22] A.-C. Ngonga Ngomo. Helios – execution optimization for link discovery. In *ISWC*, 2014.

[23] A.-C. Ngonga Ngomo, M. A. Sherif, and K. Lyko. Unsupervised link discovery through knowledge base repair. In *ESWC*, 2014.

[24] Open Geospatial Consortium. OGC GeoSPARQL Standard – A Geographic Query Language for RDF Data. https://portal.opengeospatial.org/files/?artifact_id=47664, 2012.

[25] K. Patroumpas, M. Alexakis, G. Giannopoulos, and S. Athanasiou. TripleGeo: an ETL Tool for Transforming Geospatial Data into RDF Triples. In *EDBT/ICDT Workshops*, pages 275–278, 2014.

[26] K. Patroumpas, N. Georgomanolis, T. Stratiotis, M. Alexakis, and S. Athanasiou. Exposing INSPIRE on the Semantic Web. *J. Web Sem.*, 35:53–62, 2015.

[27] M. Sherif, A.-C. Ngonga Ngomo, and J. Lehmann. Automating RDF dataset transformation and enrichment. In *ESWC*, 2015.

[28] M. Sherif, A.-C. Ngonga Ngomo, and J. Lehmann. WOMBAT - A Generalization Approach for Automatic Link Discovery. In *ESWC*, 2017.

[29] M. A. Sherif, K. Dreßler, P. Smeros, and A.-C. Ngonga Ngomo. RADON - Rapid Discovery of Topological Relations. In *AAAI*, 2017.

[30] M. A. Sherif and A.-C. N. Ngomo. A Systematic Survey of Point Set Distance Measures for Link Discovery. *Semantic Web Journal*, 2017.

[31] M. A. Sherif and A.-C. Ngonga Ngomo. An optimization approach for load balancing in parallel link discovery. In *SEMANTiCS*, 2015.

[32] D. Skoutas, D. Sacharidis, and K. Patroumpas. Efficient progressive and diversified top-k best region search. In *SIGSPATIAL*, pages 299–308, 2018.

[33] R. Speck and A.-C. Ngonga Ngomo. Named entity recognition using FOX. In *ISWC*, 2014.

[34] W. S. Torgerson. Multidimensional scaling: I. theory and method. *Psychometrika*, 17(4):401–419, 1952.

[35] U. von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.