

Data Access Linking and Integration with DALI: building a Safety Net for an Ocean of City Data

Vanessa Lopez, Martin Stephenson, Spyros Kotoulas, Pierpaolo Tommasi.
Smarter Cities Technology Centre, IBM Research, Ireland
{vanlopez, martin_stephenson, Spyros.Kotoulas, ptommasi}@ie.ibm.com

Abstract. DALI is a practical system that exploits Linked Data to provide federated entity search and spatial exploration across hundreds of information sources containing Open and Enterprise data pertaining to cities, which are stored in tabular files or in their original enterprise systems. Our system is able to lift data into a meaningful linked structure with explicit semantics, and support novel contextual search and retrieval tasks by identifying related entities across models and data sources. We evaluate in two pilot scenarios. In the first, data-engineers bring together public and enterprise datasets about public safety. In the second, knowledge-engineers and domain-experts, build a view of health and social care providers for vulnerable populations. We show that our approach can re-use data assets and provides better results than pure text-based approaches in finding relevant information, as well as satisfying specific information needs.

1 Introduction

Smart City applications rely on large amounts of data retrieved from sensors, social networks, or government authorities. Such information is often published in open data portals to promote transparency and enable innovation, as well as inviting a large community to explore how new insights can be derived from existing datasets and their combinations. For example, the NYC data platform [19] allows users to explore datasets through keyword search or by navigating through their catalogues.

Cities need to exploit this valuable resource in combination with data from their existing enterprise systems. Open data is often published in the form of tabular data, with little or incomplete schema information, while enterprise applications typically rely on complex relational schemas. There is a clear need to make city-specific information easy to consume and combine at low cost, but this proves a difficult task. To fulfill the potential of exploiting large volumes of data and obtain insights, in response to complex information needs, the following challenges are to be tackled:

- *Data Discovery.* How to discover datasets and facts for different user tasks, given the complexity of the domain, extreme heterogeneity, diversity of the data, lack of a priori defined schemas, and poor semantic catalogues.
- *Data Integration* How can data be understood in order to uncover relationships, in face of a dynamic and open environment, the infeasibility of creating a single model to cover the entire domain and the poor scalability of N-to-N integration approaches.
- *Data Exploitation.* How to create actionable views to provide relevant insights across all data sources, for a broad set of tasks, with minimal user effort?

In this ocean of data, Linked Data technologies can improve interoperability and discoverability of datasets by reusing standard vocabularies, linking to external sources, as well as enabling richer querying [5][17]. In this paper, we present and evaluate DALI, a system that puts together existing semantic techniques to offer a lightweight and incremental information sharing approach, on top of heterogeneous enterprise city data and selected well-formed open data in tabular form, as well as an end-user application, to search and consume city data online. Our contributions are:

- *Open Distributed Modeling*. Organizations can expose structured and semi-structured information based on their models (ontologies) of choice. The system ingests and integrates data in an incremental manner, lowering the entry cost by importing datasets as they are, and mapping them to other sources as needed.
- *Web of Data Integration*. By lifting data to existing models and exploiting overlap across ontologies, hidden links across entities are uncovered, in response to user searches or explorations in the context of an existing dataset. Also, using LOD URIs as target vocabularies enables to uniquely identify and organize topics and to access more information about them when needed, fully reusing the Web-wide wealth of resources.
- *Fit-for-Use*. Search and exploration interfaces allow users to profit from the expressive power of semantic standards, answering to complex information tasks, while hiding the complexity behind the data representation and services exposed.

This paper is structured as follows. Two motivating scenarios, not currently addressed without the adoption of semantics, and our approach are presented in Section 2. The architecture and components for lifting, mining annotations and contextual retrieval across distributed sources are presented in Section 3. Experimental evaluation, discussion and our position against related work are presented in Sections 4 and 5.

2 Motivating Scenarios and Approach

We present DALI in the context of two representative industry scenarios, driven by IBM solutions, that require tackling the *discovery-integration-exploitation* challenges discussed above. The first is to allow data-engineers using IBM Intelligent Operation Center (IOC) [11] enhance enterprise data with open data. The second is to support data-engineers build a Safety Net of health and social care providers and community services from public sources. This Safety Net can be used to support care workers finding services targeted to vulnerable populations in a city, and to create personalized care plans based on patient needs, in the context of IBM Cúram [13].

Scenario 1. Enterprise data obtained from IBM IOC in Minneapolis, and stored in IBM DB2 relational tables, is enriched with relevant open city data, which comes in the form of spreadsheets made for consumption by humans. The enterprise data, pertaining to events in the city, describes, among others, a point in time, a location, and a type (e.g., police calls reporting different incidents, events in an stadium like a lost child or a spectator requiring medical assistance, licenses granted to establishments, etc.). These events from multiple sources can be visualized in a map or a dashboard.

Consider a field worker concerned about safety issues in the city. In order to prioritize onsite inspection, she has access to an enterprise dataset about police call

outs related to safety issues, that she would like to overlay with other open datasets with relevant information, such as places with higher average rates of crime. She may also be interested in the location of hospitals near the places generating most ambulance and emergency call outs, or with higher response times. In this scenario, users explorations have a geographical focus and relevant links are uncovered on-demand. The required information is coming from different domains and sources that one can typically find in data-hubs for a given city (covering domains such as health, environment, public safety, education, recreation, etc.). In particular, we included datasets from geocommons.com, as well as some national-level datasets for the USA.

Scenario 2. Integrated Care aims at improving patient care by coordinating social and health care services for the vulnerable populations like the elderly or homeless [15]. We exploit open data to build a Safety Net knowledge graph to support care for seniors and their families in New York City. A Safety Net is a foundational and extensible “map” of all known care services, their characteristics and connections for a target vulnerable population and area. To build a Safety Net we can bring in multiple resources for social and health related services published in the NYC data portal [19], Geocommons, as well as Medicare public data on quality of care across the U.S. [20]. Starting with this siloed, noisy, heterogeneous data with no single schema, entities are automatically extracted to understand the type of organization (e.g., hospitals, pharmacies, day care centers, meal delivery services, etc.), the services they offer and other attributes – such as contact details, opening hours, private or non-profit-, or particular attributes for each provider across the different datasets – like readmission rates and main specialities for hospitals (cardiology, neurology, etc.), or targeted

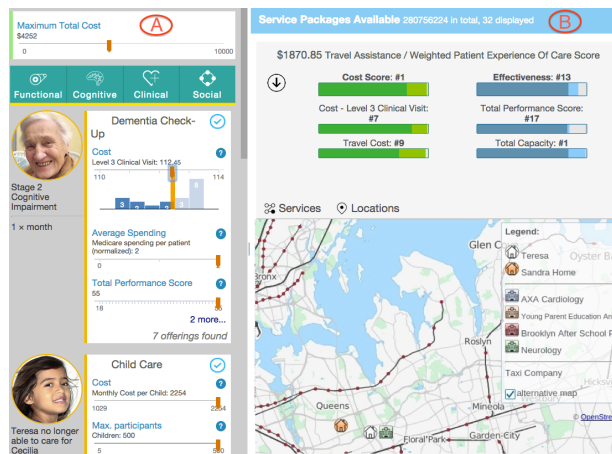


Fig 1. Use case demonstrator for social care planning

for each provider, like estimated cost, distance or hospital ratings for given medical specialities. These views can be exported in a JSON format to feed analytics and optimization algorithms to calculate the optimal services and plans, according to the combination of the various criteria and constrains on providers specified by the care team. Fig.1.A shows a screenshot on the imported views with the providers and attributes relevant to care for two needs of a given patient and her family. In this example, the selection of hospitals that care for dementia is based (among others not shown) on their average spending

populations (families, youth, immigrant, elderly) for community resources.

Existing models are used to annotate, catalog and link the data, therefore mapping each entity into a linked Safety Net graph. Data services are provided on top to create views, with information about providers and services available. Experts can then search, select and constrain the relevant criteria (facets)

for each provider, like

and total performance score, as stated across different Medicare datasets. The criteria for childcare providers are based on a cost range and max. number of participants. In Fig.1.B, one can see the combination of providers and their locations on the map, in the context of a planning component (that is beyond the scope of this paper).

The first scenario motivated the development of the system, and it is used to showcase examples in the rest of the paper. The second scenario shows that the system is not tailored to a particular use case. The resulting consolidated, multi-faceted, linked information is used to bootstrap search and exploration and expose it to users, moving from catalogue-based content management to searching and querying for entities and their relations across sources, aggregating information into customized views.

We propose a data-centric approach that consists of 3 steps as explained next. Firstly, raw tabular data is ingested and semantically lifted. Secondly, the entities and relations are automatically annotated and aligned to well-known vocabularies and widely used Linked Open Data (LOD) resources. Thirdly, different spatial views and exploration paths are exposed according to dynamically chosen models, other related datasets, and interaction paradigms, such as keyword and faceted search.

3 Architecture and Components

We present a flexible architecture (Fig. 2), in which the following functionality and main contributions are exposed:

1. *Distributed data ingestion and virtualization (Data Server)*. Enterprise relational data and tabular open data files are accessed and exposed as virtualized RDF via SPARQL end points. The distributed nature of RDF allows access to linked information across silos and from different agencies. An initial semantic uplift is done at this point, to identify entities, labels, *datatypes*, and geo-temporal data.
2. *Identification and semantic uplift of entities from open and enterprise data to an open set of specified ontologies (Application Server)*. External LOD sources and ontologies are used to annotate the data, providing meaning, context and links across sources and entities exposed from open data.
3. *Contextual information retrieval (RESTful APIs)*. Efficiently retrieve entities based on space and semantics relatedness, given a user query or through explorations of related entities within some geographic proximity. Functionality is exposed as RESTful APIs for easy developer consumption (no semantic knowledge required).

The **Data Server** component abstracts from the infrastructure of each source, the information is accessed from distributed sources as RDF by exposing virtualized SPARQL end points. The **Application Server** component then accesses the exposed SPARQL end points to extract semantic annotations (using the reference ontologies) and schema information. These annotations and schema are stored in a centralized context store based on Jena TDB, where different graphs are created and associated for each distributed source to keep provenance. The context store is indexed using LARQ [16] that enables to perform text searches on all labels, as part of SPARQL queries.

This architecture allows for incremental integration. New datasets, reference ontologies (annotation sources) can be configured and added at any time. The system automatically lifts, exposes and annotates new datasets, or if new reference ontologies

are added, the system aligns each data source with the new models, adding the new annotations in the context store. Multiple data repositories are maintained and queried in a federated manner using the REST services to exploit DALI semantic capabilities.

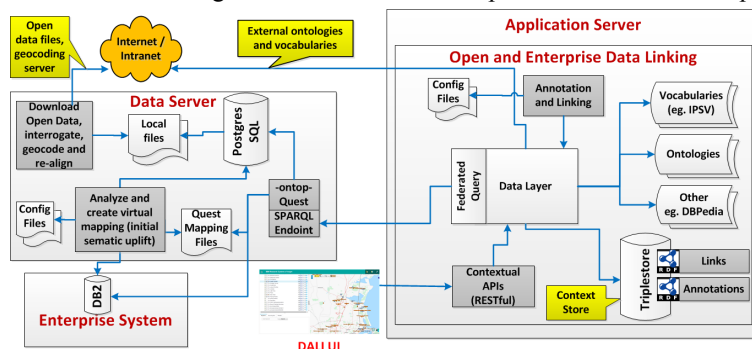


Fig 2. Architecture and component diagram

For the prototypes and setup used in our **experiments**, for scenario 1 (public safety in Minneapolis), over a hundred datasets (from one customer’s DB2 enterprise database and open datasets) were automatically integrated, semantically annotated and linked in less than one hour, producing approximately 1 million virtualized data triples in two SPARQL end points (one for open data and one for customer data). For those datasets, almost 190.000 triples and annotations were extracted and stored in the centralized and indexed context store. For scenario 2 (NYC Safety Net) 34 datasets were ingested into one SPARQL end point, consisting on almost 114,440 data points (“rows” of data) and 3.5 million triples. However, any number of SPARQL end points can be configured, in order to meet potential scalability requirements.

3.1 Distributed Data Ingestion and Virtual RDF (Data Server)

The semantic layer enables de-coupling from the infrastructure of each source. While original enterprise data resides in the original relational systems and is accessed through virtual RDF, tabular files are automatically downloaded and linked to a relational database (currently PostgreSQL9.3). As stated in [26] having a semantic representation on top of a relational one improves data quality without adding much overhead when converting CSV to a simple database schema. The datatypes are determined by examining the data: numbers, booleans, dates are converted into the correct format.

We use *-ontop-Quest*[24] as a virtualization technology, although, due to our flexible architecture, we can set as many different types of SPARQL endpoints as needed (e.g., interfacing directly to other triple stores, or other virtualized Enterprise DBs). Mapping files (i.e. files specifying one-to-one mappings between database property values and known RDF properties) for the virtualization servers are generated by our system using a rule and pattern-based entity extraction mechanism to detect: (1) **geographical entities** (using WGS84[29] to create properties for certain header labels with cell values corresponding to decimals number between -90 and 90); (2) the column with **names for instance labels** (*rdfs:label*); (3) columns (properties) with **contact** information: emails, addresses, phone numbers (named using VCARD[28]); (4) **temporal properties** (named using OWLTime[21]). Often, temporal properties

(dates, month, year) are not part of the table itself but they need to be inferred from the table titles (e.g., *Crime Stats May 2013*); and (5) **object properties**, those columns for which values are mapped to instance URIs instead of literals, as for **datatype properties**, that's the case if the column is a foreign key, or for string (non-numeric) repeating values (below a threshold variance percentage) - e.g., city names. In addition to virtualization, this step includes **geocoding** of addresses.

3.2 Entity Uplift and Linking to the Web of Data (Application Server)

For each dataset in the virtualized RDF repositories, the schema information is extracted and stored in the centralized Context Store: types, datatypes and object properties, and their set of possible instances, domains and ranges, together with entities' labels, if known, for indexing purposes. While some of the properties could be mapped to the W3C vocabularies in the previous step, to create a richer representation the entities in the Context Store are annotated and linked to an open set of both general and domain specific ontologies (that may vary according to the application domain). We use index searches and string similarity metrics [2] on the localname or label to annotate classes and properties with URIs found in the external sources used as annotators, as well as to find *owl:sameAs* links across instances. As such, we can detect synonyms and interpret acronyms (e.g., an instance named "PTSD" will be annotated with "Posttraumatic stress disorder", a DBpedia [1] redirect of the former).

Reusing well-known external sources to annotate the data adds significant value in terms of interoperability and discoverability, providing global meaning and common anchors across otherwise isolated data sources, without requiring the creation of a common model. In our scenarios, we use the Integrated Public Service Vocabulary (IPSV)[14], schema.org, WordNet[9] and DBpedia, which provides a wide domain coverage and geographical information. Specialist domain-knowledge models can also be used according to the use case. In particular for the Safety Net scenario we added the Social Care Taxonomy extracted from [27].

The annotations obtained for class labels that correspond to the table titles, often indicate the topic of the dataset. They are used to populate the Dublin Core [6] property *dcterms:subject*. Besides string similarity, the structure of the ontologies is used to disambiguate and assign a confidence score to the candidate annotations. For example, for the dataset class labeled *Minneapolis Crime Data* various *dcterms:subject* properties are added to link this term to the DBpedia terms *Category:Crime_Data*, *dbp:Crime*, and *dbp:Minneapolis*, as well as the IPSV term *Crime*, among others¹. The annotations *IPSV:Theft_and_burglary* and *dbp:Robbery*, for a property in this dataset labeled *Robbery*, will have a high confidence score both as a good syntactic and semantic mappings, capturing how semantically close the URIs are in the original graph (the subject annotation *IPSV:Crime* is a broader term for the property annotation *IPSV:Theft_and_Burglary*, similarly *dbp:Crime* relates to the property annotation *dbp:Robbery* through a common broader term *dbp:Criminal_law*). These annotations, linking the source URI and the annotation URI, and their assigned confidence scores are stored in an annotation graph in the Context Store.

¹ IPSV and dbp correspond respectively to <http://id.eds.org.uk/subject> and <http://dbpedia.org/resource/>

3.3 Contextual access and retrieval (RESTful APIs)

User needs are (a) *complex*; often, they cannot be expressed in a single query and exploration mechanisms are needed; (b) *not known in advance*; and (c) *comprising many factors* and requiring related information coming from different domains. Contextual retrieval requires understanding space, time, identity and links between entities. Annotations are used to capture the meaning of content in our RDF stores, by making explicit how entities are connected. The linkage is based on inference along linguistic relations in thesauri, taxonomies (e.g., *skos:broader/narrower* in IPSV, hyper(hypo)nymy in WordNet) and any kind of semantic relationships, such as *DBpedia redirects*, *dcterms:subject*, *owl:sameAs*, etc. In this sense, two disparate datasets about diverse topics, like *Ambulance Call Outs* and the *Register of Fats and Oils Licenses* may both be relevant in the context of a user correlating the location of ambulances call outs and hospitals, because the latter contains the locations of establishments for which a license has been granted, where establishments can be filtered by type (hospitals, restaurants, etc.). User search and exploration needs while interacting with the data are captured and translated into structured queries. The retrieved information can be visualized on tables, maps, charts or as a ranked list of search results and saved into views, which can be exported in JSON or RDF. The following contextual APIs are exposed through REST services and integrated into a web based UI².

Catalogue-based dataset and entity explorations. Datasets can be explored according to the virtualized repositories where they belong (e.g., for Scenario 1 we have two repositories for customer data and open data) or by following any given reference taxonomical models. In both scenarios, the IPSV hierarchy is selected as the reference model for thematic catalogue exploration because of its wide coverage of city related topics, and a subset of DBpedia, namely all entities of type “PopulatedPlace” and their *PartOf* taxonomy, is selected as the geographical model. For the Safety Net scenario, the domain-specific Social Care Taxonomy is also used to catalogue the data. Datasets are organized into a hierarchical view of subcategories in the reference model(s), allowing an easy and thematic browsing of the data. The alignment is done automatically when the entity representing the dataset type is annotated with the model (*dcterms:subject*). Thus a dataset may sit in more than one subcategory, if appropriate. To avoid users having to navigate through empty categories, only the part of the catalogue tree for which there are datasets is shown.

In our Minneapolis scenario, the user can explore all datasets under the IPSV term “Safety” and subcategory “Emergencies” to find the *Fire and Ambulance call outs* dataset sit under the subcategory “Fire and rescue services”, the term its been annotated with. The user can also view at a glance all known datasets under a given location, e.g., all datasets for USA, state of Minnesota, Minneapolis city. The user can click on a dataset to display the tabular data (generated from their representation in RDF), explore the annotations, or plot spatial entities in a map. By clicking on any of these entities in the map the user can also explore its properties and attributes.

² Videos showcasing DALI: <https://www.youtube.com/playlist?list=PL0VD16H1q5INAARBVy4GtTSRLN4EWmcqF>

Semantic keyword search and structured filtering. Full-text search based on LARQ is used to discover entities matching the keyword search. The domain knowledge ontologies and models, used to annotate the data, are also used to expand the query with lexically and semantically related words. For example, the *Crime Data* dataset is returned as a result for the keyword search *Fire*. This is because *Fire* is semantically related to *Arson*, a datatype property in the dataset. *Fire* is lexically annotated with, among others, the IPSV term for *Fire* *IPSV:613* that is related to the term *IPSV:612*, also known as *Arson*, through the property *SKOS:related*. Datasets are ranked by number of matches (classes, properties and instances), weighted by the average syntactic score given to each match. If no matches are found for compound terms, they are recursively split into their constituents, e.g., *hospital health centers* would get datasets with results for both *hospital* and *health centers*. Each partial term is also semantically expanded and results are ranked considering also which part of the compound is matched (e.g., a match to *health center* is ranked higher than one to only *centers*). As per user request the matches can be plotted on the map and their provenance (semantic relatedness to the keyword) displayed.

While keyword search is a popular paradigm to retrieve data, structured queries provide the expressivity to specify complex information needs. Keyword search can be combined with faceted and spatial explorations in an iterative process, where the user can enter keywords and further refine the query by applying faceted filtering on the results, or any other dataset of interest. For example, the search for *Fire* gives back several matches in the *Police CAD* dataset, namely various instance values of the object property *Problem – Fire Assault*, *Assist Fire Personnel*, etc. – for each retrieved police call entity. The user can select to plot in the map only the entities related to *Fire Assault*, and overlay them with crime locations, from the *Crime Data* dataset, with more than a given number of arson crimes (specified by the user). In this case, *Arson* is a numerical datatype property and thus an equal/greater/lower than operator is suggested by the system to create the facet (different facets are suggested according to the datatypes –numerical/ boolean- and for object properties with a set of possible values). Users can also overlay spatial entities from any dataset in a map using common constructs such a bounding boxes. This is implemented by executing an single SPARQL query to filter all entities URIs (and labels) with *Wgs84:lat* and *Wgs84:lon* values within the bounding box geo points.

Related dataset and entity search. As datasets are aligned with ontologies through annotations, these annotations can be used to identify other datasets closely related to a given one, based on **topic** (datasets share the same or linked topics), **content** (datasets with related properties or content, even in different topics) or **entities in common** (same entities described in different datasets). In our scenario, the user can look for all datasets related to the *Crime Data* dataset with statistics on different kinds of crime. The *Police CAD* dataset is obtained as it is annotated with semantically related topics (*dcterms:subject* property). The user can explore the relatedness graph showing how the two datasets are linked, e.g., through the IPSV term *Crime and Law enforcement* as shown in Fig. 3.A for the topic-based criteria. Also, these two datasets are content-based related because they have properties or entities sharing the same annotations, as shown in Fig. 3.B, the instance value *Theft* for the property *Problem* in *Police CAD* has an *owl:sameAs* link to the DBpedia term *Category:Theft*, which is a broader term of *Category:Robbery*, an annotation property in *Crime Data*. The relatedness graphs for

each annotator source are obtained on demand through a SPARQL federated query to find if there is a path (directly) linking annotations from a given dataset to annotations in other datasets from the same background source (e.g., DBpedia). Properties extracted based on rules, such as LAT/LONG and contact details (as defined in schema.org) are not considered relevant to identify related datasets.

Related datasets are ranked by summing the relevance weights for the relatedness graphs (pv) obtained for each criterion. The weight is calculated according to how significant the entity-level matches are – i.e., for the content-based relatedness graph: how many annotations are matched (num_anns_common) out of the total for the input dataset ($total_anns_input$), as well as, considering the average confidence score (WSc) of each matched annotations to assign weights to the different criteria when combined. The following formula, used to calculate the score (Sc) of a related dataset with respect to an input, responds to the intuition that datasets are more similar if they share more labels /annotations and share labels/annotations with large weights. The most relevant datasets have the highest score: $Sc_dataset = \sum_{pvi} Avg(WSc_anns_common)^* \frac{num_anns_common}{total_anns_input}$

4 Experiments

In the first part, we perform a user study to evaluate the usability of the services exposed by means of the user interface, through a set of tasks that require retrieving complex information to create the relevant views. For the second part, we are demonstrating the effectiveness of the semantic search, whether adding related datasets has potential to improve search results, and the semantic cataloguing. We quantify the improvement on performance with respect to a non-semantic baseline. It is not our purpose to evaluate each step of the process or component independently, but to evaluate the relevancy of the results searched over distributed city data lifted into a knowledge graph, and in the context of a user-task.

4.1 Contextual exploration: usability

Evaluation Set-up. To evaluate performance in a more comprehensive manner, we have simulated a scenario, where evaluators are asked to use the system in order to answer the given complex information needs (simulating the role of a knowledge engineer). To test the ability of our system to retrieve this information, users are given a brief demo of the system and told they can use all the functionality available (Section 3.3). We have asked 5 users (all IBM employees and IT experts but not knowledgeable about semantic standards or the datasets) to retrieve the answers to the tasks in Table 3, which may span across more than one dataset. Queries 1-5 are part of Scenario 1, while Queries 6-10 are part of Scenario 2. The questions were given by experts of the respective commercial products with extensive hands-on experience in the domain. The order of the queries presented to each user was randomized. We evaluate on:

- Average number of tasks for which users found satisfactory answers vs. the ones for which they gave up or report a wrong result.

- Time to get the answer. We started a timer once the user was given the question and stopped the timer when the user would give up or report an answer.
- Which explorations and features were used to get the answers and the number of failed attempts.

Results. The results are shown in Table 2. We counted the features the users used to answer each query: semantic search (SEM), catalogue exploration (CAT), displaying tabular data, matches or entity information (DIS), plotting entities in a map (MAP), selection and faceted filtering (FAC), drawing a bounding box to visualize all entities within (BOU), and looking for related datasets and relatedness graphs (REL). The queries that were answered faster are those for which only one dataset is required to find the answer (Q3 and Q10). All users were successful in all tasks, except for Q2, Q3 and Q9 for which three different users (one for each query) gave up. For Q2 users would often attempt to find the answers in the *Police CAD* dataset (one of them gave up when she could not find it there), while the answer is found by applying faceted filtering on the property *robberies* in the *Crime* dataset (also returned as a result from the keyword search *robbery*). In Q3, one user gave up before realizing he could apply more than one facet filter in the same dataset. For Q9 a user failed to find the *readmission rates by heart attack* property in *Medicare Hospital Outcome of Care Measures*, picking instead the general *expected readmission rate* specified in the *Medicare Hospital Readmission Reduction* dataset. Facet filtering can then be applied to get the entities with the minimum readmission rates value.

Table 1. Test questions with the minimum n° of relevant sources and navigation links (that is the minimum number of steps as determined by the authors to obtain the answers).

n° sources	Question (min. navigation links to answer them)
2	Q ₁ : The sport stadium in Minneapolis near one of the most dangerous pedestrian areas in the USA(4)
2	Q ₂ : Which cafes are near robbery crimes areas in Minneapolis (5)
1	Q ₃ : Locations with more than 10 car thefts and 10 arson crimes (3)
2	Q ₄ : All places holding both a liquor license and sidewalk permits (2)
2	Q ₅ : All police disturbances near the Creekview center (5)
3	Q ₆ : Community programs for the elderly population in Queens (5)
2	Q ₇ : Readmission rates for hospitals with emergency services (4)
1	Q ₈ : Home delivery meal services in the Bronx (2)
2	Q ₉ : Non profit hospitals with the minimum mortality rates from heart attack (3)
1	Q ₁₀ : After school programs for middle school kids (2)

For most tasks, users started by using keyword search or catalogue exploration. When catalogue exploration fails, such as when looking for *Home Delivered Meals* in Q8, users will use keyword search to find information hidden in the datasets (in this case the entities in the *DFTA_Contracts* dataset, which value for the object property *Contract_Type* is *Home_Delivered_Meals*). Besides searching and catalogue exploration, plotting a dataset or search results on the map and displaying entity and tabular data were features used in all queries (the latest, often used just to figure out if the provenance of the given answer, or search result, is sufficient). Facets were used in almost all queries. The query with the second largest number of attempts, Q7, is because first the *boolean* property specifying if a hospital has emergency services is found in a different dataset to the readmission rates one; and second, it took a while for a few users to understand that they can plot in the map the entities in common for both datasets, by combining the faceted and co-reference filters. In general, all queries related to hospitals took longer in average because of the large amount of clinical data,

both in terms of number of datasets about hospitals, and the number of properties within each dataset (more than 50 in some of the Medicare datasets). For these cases, search is more efficient than catalogue browsing.

The bounding box feature was rarely used, even if it is the faster way to answer queries such as Q1 (e.g., by drawing a bounding box near the most dangerous pedestrian area in Minneapolis), or even when the users knew where to look in the map. For example for Q5, once the Creekview center was found (in the dataset *Minneapolis parks and recreations*) only one user attempted to find reported disturbances by searching in the datasets with entities near by. The reason behind Q5 largest number of failed attempts, is because users searched for the answers in both the *Police CAD* and *Crime* datasets, while only the former has the answer for disturbances. Nonetheless, most users would prefer to use faceted search to filter by area (if the property exists, such as Queens for Q6), rather than a bounding box. Relatedness was also hardly used, even if it is a useful feature for queries such as Q9 (the query that took the longest in average) to find all other hospital datasets with related properties to the *Medicare Hospital Readmission Reduction* dataset, e.g., those describing different measurements on readmission rates. Users preferred to do a semantic search and explore all matches till they find the one that is most appropriate. Finally, all queries were answered in average in less than 4 minutes, although the deviation between users varied greatly (with a max. time of 6 minutes), and required less than 7 steps in average (links) to retrieve the answers to a query. In total, the average of the sum of all failed attempts per query for all 5 participants is 7.5.

Table 2. Results from left to right for each query: the average of the sum of all features used by the 5 participants per query; the average num. of links used by user to get an answer; the average of the sum of all failed attempts for all 5 participants per query; the num. of users which succeeded; the average time to answer the queries and the deviation.

Features used (for all 5 users per query)							Avg. links (For user)	Failed attempts (for all 5 users per query)	Success	Avg. time	Deviation	
SEM	CAT	DIS	MAP	FAC	BOU	REL						
Avg	8.6	4.1	9.4	7.6	4.9	0.5	0.8	6.7	7.5	4.7	0:02:37	0:03:04

4.2 Semantic search: performance

Evaluation Metrics. We compare the precision and improvement on recall of our semantic approach with respect to a syntactic baseline based on *Lucene* full-text searches, without semantic expansion. We measure **precision** (P), defined as the number of relevant datasets with respect to the number of datasets found, and **recall** (R), defined as the number of relevant datasets found with respect to the total of relevant datasets. Total recall cannot be measured (no gold standard to evaluate against), so we consider as total all the unique relevant datasets found using both approaches, and measure the improvement in recall of the semantic approach (R_s) with

$$\text{respect to the baseline } (R_B) \text{ as: } R_{\text{improvement}} = 1 - \frac{R_B}{R_s} \quad P = \frac{\text{Total}_{\text{relevant}}}{\text{Total}}$$

Increasing recall often comes with a decrease in precision, which is affected by the quality of the annotations, noisy mappings and ambiguous lexically related words. Therefore, precision is also measured for the top N of results: **TOP-1**, **TOP-3** and **TOP-5**, in order to evaluate the efficiency of the ranking.

Evaluation Set-up 1: Semantic Search. We evaluate the impact of semantic query expansion on 20 keywords, obtained randomly from the logs generated after the evaluation in 4.1. We distinguish between *coverage*, *correctness* and *relevance*. Coverage is measured by counting the number of results. However, this does not indicate whether the results are relevant. To evaluate correctness and relevance we have engaged three of the previous five evaluators. Each of them has assigned a score with a discrete value in $\{0,1,2\}$ for each datasets retrieved as a result, where: 0 implies the proposed dataset is based on semantically incorrect assumptions, i.e., due to an ambiguous annotation; 1 implies the proposed dataset is based on semantically correct justifications, but it is not relevant; and 2 implies the proposed dataset is correct and relevant to complement the information of the original dataset. For users to judge relevancy they can relate to the tasks presented in Section 4.1. Given the three user evaluations, a result was considered correct if at least two evaluators were rating it with values higher than 0, and it was considered relevant if at least two evaluators were rating it with 2 and the remaining evaluation was not 0.

Set-up1 Results: as shown in Table 3. The semantic approach improves the average recall of the system by 33%, as it is able to find all the relevant datasets found using the baseline approach plus some additional ones, without much loss in precision, from 70% in the baseline to 69% using the semantic approach. Furthermore, as shown in Table 4 the semantic ranking increases the precision from the 0.7 average for the baseline to 0.9, 0.78 and 0.75 for the top-1, top-3 and top-5 results respectively. In sum, a semantic approach helps increase recall, while also increasing the precision for the top ranked results, i.e., those users are likely to check. The syntactic errors are mostly due to ambiguous terms, e.g., the keyword *Fire* was mapped to the *Fire Station* dataset, but also to the instance *Fire restaurant* in the *Fats and Oils Licenses* dataset. The users rated the first match as relevant (2) and the second as incorrect (0). While these syntactic mappings were also captured by the semantic component, the inaccurate mappings were ranked lower than the more accurate semantic matches. In addition, other relevant datasets were only found using the semantic approach, like the *Crime Stats* with the property *Arson*. In two cases (*pedestrian* and *liquor licenses*) the syntactic approach performed slightly better than the semantic. This is because all relevant datasets were syntactically matched, and the semantic extension retrieved inaccurate lexically related results – that although they scored lower they were part of the top-5 (as less the 5 datasets were found in total). The semantic approach improves over the baseline in particular when asking for schema elements (types, properties) or term combinations (e.g., *sport stadiums*) rather than instance labels. In the latter case, both approaches perform the same (returning the instances with matching labels).

Table 3. Comparison between baseline and semantic approaches

Query	Baseline		Semantic Approach		
	Coverage	Prec./Recall	Coverage	Prec./Recall	Recall Improvement.
20 total					
TOTAL	3.15	0.70 / 0.66	5.65	0.69 / 1	0.33

Table 4. Comparison for the top ranked results

Qi	Baseline			Semantic Approach			Relatedness-TOP3		
	TOP-1	TOP-3	TOP-5	TOP-1	TOP-3	TOP-5	#Total	#New	#Rel
Average	0.7	0.69	0.70	0.9	0.78	0.75	7.15	5.45	3.25

Evaluation Set-up 2: Semantic Relatedness. We evaluate whether our algorithms that retrieve highly related tables can also improve the result of searches (by pulling up datasets based on their relatedness to top ranked datasets). To measure improvement on recall: (1) we find all related datasets for the TOP-3 ranked results in each of the previous queries (#Total); (2) from all related datasets, we select only the ones that did not appear in the top-5 search results (#New); and (3) from all the #New datasets, we select the ones rated as relevant (#Rel);. As before, we randomly ordered the searches and asked the 3 users to rate the results between 0-2. A dataset selected as related to a given one based on semantically correct assumptions (e.g., common annotations) may not be relevant, because the content of the dataset is not specialized enough to give any extra information, or the commonalities are not significant. In the same way, a disparate dataset from a different topic may be relevant because it describes similar entities from different points of view relevant as part of an exploration task.

Set-up 2 Results: in Table 4, on the right, we show the number of unique related datasets added by looking at related datasets from the TOP-3 (#Total), that do not appear as part of the TOP-5 semantic search results (#New) and that the users rated as relevant (#Rel). In average 5.45 datasets out of 7.15 found were not in the original TOP-5 results. A significant portion of the new related datasets are considered relevant (3.25 obtained an average user rating of 2). Thus, relatedness can be used to suggest further relevant explorations from user searches by making “hidden semantic links” across entities explicit, based on different *relatedness* criteria. In this experiment only the related datasets with an average score over a 0.5 threshold were selected. As an example, *Crime Data* is one of the TOP-3 results for *Fire*, because of the annotation property *Arson*. A related dataset, not part of the top ranked search results, is *Law Enforcement*. This is because, both datasets share the same IPSV subject *Crime*. Ranked before *Law Enforcement* is *Police CAD*, as their respective annotations are linked through IPSV and DBpedia, as shown in the relatedness graphs in Fig. 3.A&B. The total score for a related dataset is calculated by considering the combined weight of all relatedness graphs across the two datasets. As such, like in semantic search, datasets with inaccurate matches from ambiguous annotations are often ranked in lower positions. Users judged the relevance of the results, assessing if the relation between two datasets is meaningful considering the tasks in Section 4.1. The Fleiss’ kappa agreement between users was moderate/substantial ($k=0.6$)[8].

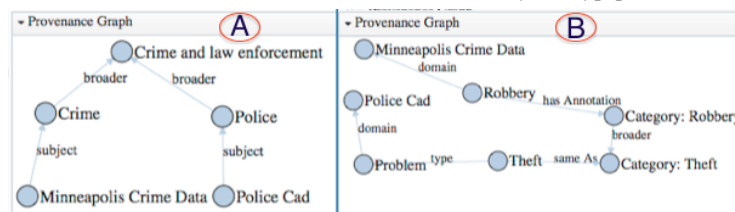


Fig 3. Relatedness graphs for related datasets

Evaluation Set-up 3: Semantic Catalogue Search. IPSV is selected as the ontology of reference for browsing and searching datasets by category. For a total of 30 randomly chosen datasets, we ask the same 3 users to rate how well the dataset has been classified for each category (note a dataset may belong to more than one) with a value between [0-2], where 0 means the category is inaccurate, 1 that the dataset is correctly

classified but it is not too helpful (e.g., not at the right level in the taxonomy), and 2 the category is relevant to find the dataset by browsing through the hierarchical tree. We measure precision as the number of datasets under a relevant category from the total.

Set-up3 Results: all selected 30 datasets are categorized (100% coverage), each dataset is categorized into an average of 4.33 categories (the maximum being 11 and the minimum 1), and 29 datasets out of 30 have at least one relevant category. If a minimum score is applied as threshold (experimentally selected to favor precision, i.e. less datasets placed under an incorrect category, with a little negative effect on recall), to ensure that the categories matched to a given term have certain quality, then we get 2.16 categories in average (with a maximum of 6 and a minimum of 1), and 28 datasets out of 30 have at least one relevant category. The precision for the relevant results in the non-filtering approach is 0.54; while in the score-based one is 0.78. The precision for the correct results is 0.84 and 0.93 respectively. The loss in recall is 0.76 on the score-based approach if we consider the non-filtering one as perfect recall (1). As an example, to specific categories may not be relevant if they are too specific, such as *Swimming pools* as the category for the dataset *Leisure Facilities*, or too general, like *Leisure and culture*. For this dataset, *Leisure center* was rated as the relevant category.

5 Related work, discussion on usefulness and future lines.

We address the timely issue of data consumption, exploration, search and linking in the context of cities, through a practical approach based on Linked Data. There are a number of unique challenges and opportunities for the IR and semantic communities in order to make heterogeneous city data searchable, and to address complex information needs that require analyzing the relationships between entities in context. In this light, we propose a lightweight and incremental information sharing approach directed towards leveraging the information spaces defined by the LOD datasets and city data of diverse ownership, to give meaning to the latter.

Various publishing platforms exist for automating the lifting of tabular data into semantic data, and interlinking datasets with existent LOD datasets [25][18]. In [18] Google Refine is used to allow expert users to clean and export tabular data into RDF through a reconciliation service extended with Linked Data sources. Following the tools and recommendations by the W3C used for automatically converting tabular data (mostly CSV) and relational tables into RDF[12][10][7], in [26] a set of tabular data from the Norwegian directorates FactPages is transformed into a LOD dataset. The DataLift project [25] goes a step forward by transforming the raw RDF extracted from the source format to well-formed RDF by mapping to selected ontologies. These approaches are based on the assumption that each row is an entity and columns are RDF properties for the first RDF conversion; In [25] the user is asked to input a set of vocabularies to describe the lifted data and the target RDF is then generated through a set of SPARQL construct queries. In QuerioCity[17] we proposed a platform to provide semantic context for city data and metadata by following a centralized and incremental graph-based approach. The focus is on data-view manipulation by different publishers while tracking provenance. However, the drawback is the significant added cost on indexing this data. Unlike previous approaches, DALI

presents a light-weight approach that considers the distributed nature of RDF and it is able to ingest any customer or open data available, as long as it follows a tabular representation. The data is (1) virtualized into RDF, extracting spatial/temporal entities, datatypes and object properties for each entity (row), even if often entities are not linked (no foreign keys); (2) exposed and contextualized with any reference ontologies and models of choice in the Web of data; and (3) combined in arbitrary ways across data sources, through semantic services that support users, without knowledge of SPARQL, to refine explorations and federated queries.

Extracting structured data from tables on the Web and semantic search has also attracted interest from search engines [26][3]. In [26] columns in web tables are associated with types (automatically extracted from web pages), if the values in that column can be matched as values of the type. In [22] table rows containing entities of specific types derived from an ontology are automatically annotated. In [3] an approach is proposed for finding related tables on the Web based on: (1) Entity complement: union of entities with similar schemas; (2) Schema complement: joins of columns about the same entities. The BBC has annotated its world service radio archive with DBpedia topics. These associations, stored in a shared RDF store, are used to improve search and navigation within the archive [23]. Different from these works we propose an application that covers not only the annotating and semantic querying across a diverse set of heterogeneous, distributed enterprise and well-formed open tabular data for cities, but also the lifting of these data silos to Linked Data.

There are no tools that we can use for a meaningful comparison and we still are a long way from defining standard evaluation benchmarks to evaluate search methods for urban data platforms and comprising relevance judgment of similar datasets. From information platforms, such as DubLinked.ie, one can obtain the most common keyword searches and most downloaded datasets. More effort is needed to better capture the users' intention and experience while using the tool, in the context of complex real tasks that involve more complicated manipulation and combinations of datasets. Illustrated in the context of two real commercial use cases, our user-based evaluations are a first step to probe the feasibility and effectiveness of a knowledge-mining prototype to query entities distributed across datasets, without the need to ETL data. We also conducted a small-scale study with experts from the urban planning and the health domains to further validate the system. The experts were three city planners from Dublin City Council and three staff members from the Department of Nursing and Midwifery from Trinity College Dublin, evaluating scenario 1 and 2 respectively. We have used the widely cited methodology from Davis[4] on predicting how much people will use a new product. Given our focus on functional components, rather than user interfaces, and the higher importance of usefulness compared to ease of use (also reported in [4]), we report numbers on the former. In Table 5 we report combined numbers since there were no significant differences between the scenarios. Overall, users gave positive scores (avg. =5.85, on a scale of 1-7). Critically for the goals of our system, users gave the system a high score for allowing them to work more quickly.

Table 5. Questions and usefulness scores [1-7] for both scenarios.

<i>Quality of work</i>	6	<i>Control over work</i>	5.5	<i>Work more quickly</i>	6.5
<i>Critical to my job</i>	5.5	<i>Increase productivity</i>	6	<i>Job performance</i>	5.5
<i>Accomplish more work</i>	6	<i>Effectiveness</i>	5	<i>Makes job easier</i>	6
<i>Useful</i>	6.5				

There is an abundance of research to be pursued: off-the-shelf co-reference tools can be used to make links more dense, as long as they do not depend on the availability of training data; incorporating social media data; exploring interactive ways to build NLP queries, exploiting user feedback to improve the machine-generated ranking, and supporting dataset discovery in the Web of Data using data-hubs such as CKAN.net.

References

- [1] Bizer, C., Lehmann, J., Kobilarov, G., Auer, S., Becker, C., Cyganiak, R., Hellmann, S. DBpedia - A crystallization point for the Web of Data. *Web Semantics*, 7(3), 154–165, 2009
- [2] Cohen, W., Ravikumar, P., Fienberg, S., E. A Comparison of String Distance Metrics for Name-Matching Tasks. In *IJCAI Workshop on Information Integration*, 2003
- [3] Das Sarma, A., Fang, L., Gupta, N., et al. Finding Related Tables. In: *SIGMOD 2012*
- [4] Davis, F. D. Perceived usefulness, ease of use, and user acceptance of information technology. *MIS quarterly*, 13, 319-340, 1989.
- [5] Ding, L., Lebo, T., Erickson, J.S., DiFranzon, D., et al. A portal for linked open government data ecosystems. *Web Semantics* 9(3), 2011
- [6] Dublin Core: <http://dublincore.org/documents/dcmi-terms/>
- [7] Ermilov, I. Auer, S., Stadler, C. Csv2rdf: user-driven csv to rdf mass conversion framework. In: *ISEM 2013*
- [8] Fleiss, J. L. and Cohen, J. The equivalence of weighted kappa and the intraclass correlation coefficient as measure of reliability. *Educational and Psychological Measurement*, 33, 613-619, 1973.
- [9] George A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11), 39-41, 1995.
- [10] Han, L., Finin, T., Parr, C., Sachs, J., Joshi, A. Rdf123: From spreadsheets to rdf. In: *ISWC 2008*
- [11] <http://www.ibm.com/developerworks/industry/library/ind-intelligent-operations-center/>
- [12] <http://www.w3.org/2001/sw/rdb2rdf/> and <http://www.w3.org/TR/r2rml/>
- [13] IBM Curam: <http://www-03.ibm.com/software/products/en/social-programs>
- [14] IPSV: <http://doc.esd.org.uk/IPSV/2.00.html>
- [15] Kotoulas, S., Sedlazeck, W., Lopez, V., et al. Linked Data for Citizen-Centric Care. In: *MIE 2014*.
- [16] LARQ: <https://jena.apache.org/documentation/larq/>
- [17] Lopez, V., Kotoulas, S., Sbodio, M.L., Lloyd R. Guided Exploration and Integration of Urban Data. In: *Hypertext 2013*
- [18] Maali, F., Cyganiak, R., et al.: A publishing pipeline for linked government data. In: *ESWC 2012*
- [19] NYC Open Data portal: <https://data.cityofnewyork.us/data>
- [20] Official datasets provided by Medicare and Medicaid services: <https://data.medicare.gov/>
- [21] OWL Time: <http://www.w3.org/TR/owl-time/>
- [22] Quercini, G., Reynaud, C. Entity discovery and annotation in tables. In: *EDBT*, 2013
- [23] Raimond, Y., Ferne, T. The BBC World Service Archive Prototype. In *ISWC Semantic Web Challenge 2013*
- [24] Rodriguez-Muro, M., Rezk, M., Hardi, J., Slusnys, M., Bagosi, T., Calvanese, D. Evaluating SPARQL-to-SQL Translation in Ontop. In: *ORE 2013*
- [25] Scharffe, F., Ateazing, G., R., T., Gandon, F., et al. Enabling linked-data publication with the datalift platform. In *Workshop on Semantic Cities , AAAI' 2012*
- [26] Skjaveland, M., Lian, E., Horrocks, I. Publishing the Norwegian Petroleum Directorate's FactPages as Semantic Web Data. In: *ISWC 2013*
- [27] Social Care taxonomy from the UK Social Care Institute for Excellence: <http://www.scie.org.uk>
- [28] VCARD: <http://www.w3.org/TR/vcard-rdf/>
- [29] WGS84 geo-coordinates: <http://www.w3.org/2003/01/geo/>