SPUD - Semantic Processing of Urban Data

Spyros Kotoulas, Vanessa Lopez, Raymond Lloyd, Marco Luca Sbodio, Freddy Lecue, Martin Stephenson, Elizabeth Daly, Veli Bicer, Aris Gkoulalas-Divanis, Giusy Di Lorenzo, Anika Schumann, Pol Mac Aonghusa

Smarter Cities Technology Centre, IBM Research Ireland

Abstract

We present SPUD, a semantic environment for cataloguing, exploring, integrating, understanding, processing and transforming urban information. A series of challenges are identified: namely, the heterogeneity of the domain and the impracticality of a common model, the volume of information and the number of data sets, the requirement for a low entry threshold to the system, the diversity of the input data, in terms of format, syntax and update frequency (streams vs static data), the complex data dependencies and the sensitivity of the information. We propose an approach for incremental and continuous integration of static and streaming data, based on Semantic Web technologies and apply our technology to a traffic diagnosis scenario. We demonstrate our approach through a system operating on real data in Dublin and we show that semantic technologies can be used to obtain business results in an environment with hundreds of heterogenous datasets coming from distributed data sources and spanning multiple domains.

1. Introduction

Urban data comes in many forms, shapes and sizes. Government agencies are increasingly making their data accessible to promote transparency and economic growth. Since the first data.gov initiative launched by the US government, many city agencies and authorities have made their data publicly available through content portals: New York City\(^1\), London\(^2\), San Francisco\(^3\), Boston\(^4\), and Dublin\(^5\), to name a few. In the meanwhile, Linked Data has emerged as a way to integrate information across sources and domains. Managing Open and Linked data require that publishers put significant resources. A critical question for government agencies is what return-on-investment they are getting for resources spent in making their data open. This may come as an increase in economic activity in their constituencies, decrease in administration costs and increased transparency. User generated content can provide information outside of the scope of traditional data sources. For example, a traffic jam that emerges due to an unplanned protest may be captured through a twitter stream, but missed when examining weather conditions, event databases, reported roadworks, etc. Additionally, weather sensors in the city tend to miss localised events such as flooding. These views of the city combined however, can provide a richer and more complete view of the state of the city, by merging traditional data sources with messy and unreliable social media streams.

The urban data emerging from such sources may be used to support various operations such as exploration, visualisation, querying and diag-
Nevertheless, the cost associated with integrating all of this information is prohibitive. Our claim is that semantic technologies can be used to drastically lower the entry cost to accessing the information of a city. We demonstrate a technology platform to address key business challenges for urban information management: (a) Publication of a dataset, focusing on privacy protection and semantic annotation, (b) Reporting and Consolidation of multi-faceted information, focusing on searching and visualizing heterogeneous data from several sources, including social media, linked data and government data and aggregating this information into a single view and (c) In depth analysis of this information, to derive conclusions with significant business value. Example of such conclusions include detection and diagnosis of events or anomalies.

The novelty of **SPUD** lies in the ability of the system to ingest highly heterogenous data and process it in an incremental manner. Unlike other approaches, the cost of entry is minimal (i.e. datasets can be imported as they are), and processing (annotation, linking, integration) can be done incrementally, while fully exploiting the power of semantic technologies. In addition, we are showing how a stack based on semantic technologies can go a long way, without the need for global integration, or even linking the entire input. We demonstrate **SPUD** using hundreds of real-world datasets, published by 4 local authorities, on an open data platform\(^6\) datasets from the Semantic Web and data retrieved from Social Media and other Web sources. **SPUD** incorporates several research efforts for which we provide extensive descriptions in [1, 2, 3, 4, 5].

The rest of this paper is structured as follows: Section 2 presents a set of motivating use-cases centered around Dublin. Section 3 outlines our approach. The main research methods and technologies applied in **SPUD** are outlined in Section 4 and a deployment is presented in Section 5. We present related work in Section 6 and conclude in Section 7.

2. Use-cases

We are presenting **SPUD** through a series of business cases pertaining to ambulance response times in Dublin. The target audience has various roles within public administration (or contracted entity working with public authorities) and varied competency with regard to semantic technologies. For each use-case, we are outlining how **SPUD** addresses the related challenges.

2.1. Publish Data

A city official wants to publish a dataset about ambulance callouts\(^7\). The capability of the user is limited to Web browsing and using spreadsheets. The goals in this case is that publication should be easy while conveying as much semantics as possible and protecting privacy-sensitive information. In addition, given the cost associated with publishing data, the city authorities need to evaluate the return on investment for the publication of each dataset. In this particular case, this can be measured by how much the information or any information derived from it has been used. This case is facilitated by an easy-to-use, form-based interface with recommendations for metadata terms, taken from the Semantic Web (e.g. IPSV, DBPedia, Dublin Core). In addition, we provide functionality to identify and protect sensitive information and automatically extract some semantic information (e.g. geographical coordinates). We provide a ranking of original or derived data sources by their use as a measure of their value to the community.

2.2. Report and consolidate multi-faceted urban information

A city executive reads an article about ambulances missing targets for response times\(^8\) and

---

\(^6\)http://www.dublinked.ie

\(^7\)due to privacy considerations, we can not provide the original dataset, and have thus generated a synthetic dataset based on realistic values. An example of an anonymized dataset from Dublinked can be found in http://dublinked.ie/datastore/datasets/dataset-027.php

\(^8\)https://ibm.biz/Bdx2FJ
tasks a city official with creating a thorough report, including locations of critical infrastructure, problem points, citizen-centric information and some Key Performance Indicators. The capability of the user is as in the previous case, but the user has better insight in the organization of the municipal authorities. The goal in this case is to retrieve all relevant information. In addition, this information should be composed so as to get a single, thorough, view (for example overlaying delayed ambulances with known traffic jams from traffic systems and citizen reports from social media). In addition, to be able to process data in a quantitative manner, it should be possible to merge data into a single view. Our exploration panel allows searching on content and metadata, on multiple data sources (including social media) and matches information on multiple levels - lexical, spatial and semantic. The retrieved information can be visualized on tables, maps and charts.

2.3. Analyze and correlate information

As input to emergency authorities and traffic systems, the city authorities want to move further into understanding why a given traffic situation exists. Building on the output of the previous case, a data modeling engineer is tasked with creating a diagnosis component for traffic situations around sensitive infrastructure. The goals in this case are to fuse and semantically lift the data which is in turn fed to an automated reasoning component. Our system analyses and ingests data from social streams, linked data sources and information published by municipal authorities. This information is either used as input or for validation of results obtained by the diagnosis reasoning engine. The result is a substantiated view of possible causes of traffic problems around sensitive health infrastructure.

Through the aforementioned cases, we demonstrate how we can get value out of open, linked and social data in an urban setting. SPUD facilitates the entire data processing lifecycle, from information publication to gaining insight through highly expressive reasoning.

3. Approach

In this Section, we are describing the general approach we are taking in SPUD. Figure 1 summarizes the steps taken to go from raw data to a useful business result, from a data management perspective.

3.1. Format

Datasets in the system are preserved in their original formats. In addition, we transform datasets with known file formats to a simple RDF representation. This representation is not intended to capture semantics, but rather to provide a convenient and uniform way to represent the content of files, which are further processed as described in the following sections. See Figure 1 for an example of the used representation.

3.2. Structure

Once homogeneous data areas are identified and validated with the user, pre-defined templates can be used as semantic masks that capture the intent of the publisher and guide the extraction of entities, making explicit the relations that hold between the entities described on the tables. Templates can be defined a priori but they can also be learned through user interaction and saved to be reused. The three dominant structures for the tabular data used in SPUD are: geographically referenced entities (i.e. tables with two columns for longitude and latitude), measurements with a single entity per row and a column indicating the temporal aspect, and structures representing measurements that reference time through both a given column and a row.

3.3. Links & Semantics

The platform leverages semantic data types (geographical coordinates, dates, etc.) and automatically converts units of measurement. Owl:sameAs and owl:equivalentAs properties are used to link entities, eliminating the need for tight physical integrations imposed by relational databases and adopting a pay-as-you-go approach. These properties are discovered using a combination of existent reconciliation and state-of-the-art mapping techniques to detect common
types and entity co-reference as well as user input. In addition, we can consume services provided by the http://sameAs.org web site for getting co-referent URIs.

3.4. Views

To abstract from the complexity of the domain, SPUD uses semantic views as a way to expose the relevant information to applications. Instead of being closely coupled to the data layout, applications define how their input should look like and, using a pay-as-you go paradigm, SPUD populates those views.

3.5. Insight

Applications leveraging those semantic views cover a broad domain. In this paper, we will focus on our diagnosis component. Diagnosis [6] is the task of explaining anomalies, in our case road congestions, given a set of observations. Interpreted in the context of SPUD, anomalies are \( k \)-invariant road congestions. Observations are captured from background knowledge (e.g., any bus is conducted on roads) and dynamic knowledge (e.g., a bus is in heavy traffic and a sport event is active in some snapshots).

3.6. Data model

Throughout these steps, SPUD follows a pay-as-you-go paradigm: the steps above are only performed as required. For example, the cataloguing part is performed for all datasets, format homogenization is performed for datasets with a fixed set of formats (for which we have converters in place), linking is done as required by the views,
which are, in turn defined by the applications we want to run.

Figure 2 shows an example for our data model. Data is stored in Graphs. DataViews are the access methods for data in our system, each referring to one or more graphs. Graphs are shared between DataViews (i.e. a Dataset may reference multiple Graphs, and a Graph can be referenced by multiple Datasets), using union semantics. Data manipulation tasks entail creating new Graphs, which are then referenced together with existing graphs. For read-stability reasons, the only operation allowed on Graphs is splitting and re-writing existing references to the graph. The said design avoids data duplication while inducing as little overhead as possible.

Graphs are physically stored as named graph on the underlying infrastructure. Management information, i.e. the information about DataViews, Graphs etc, is also stored in RDF, on a separate named graph.

3.7. Provenance

We keep both dataset-level provenance and graph-level provenance, storing derivedFrom relationships for both Datasets and Graphs. Graph-level provenance is tunable to the resolution required, by splitting Graphs. In the extreme case, we can keep a single graph per triple, so as to have triple-level provenance. Needless to say, this will have a negative impact on performance and we have yet to encounter the need for it. In SPUD, provenance is operational with regard to privacy. When a privacy threat is detected for a given Dataset, it is not sufficient to protect this Dataset in isolation, since the process to generate it could be repeated. We use the derivedFrom relations and protect all Datasets that were used to create the Dataset with privacy vulnerabilities.

4. Technologies

In this Section, we outline the core technologies used in SPUD, providing pointers to more thorough descriptions, where available.

4.1. Cataloguing

We provide a rich publishing interface that allows annotating datasets with relevant metadata from any vocabulary (we currently use IPSV\(^9\), DBpedia, and Dublin Core). Datasets are described using DCAT\(^10\), VOID\(^11\) and PROV\(^12\). Our interface helps the user select appropriate terms by providing a semantics-augmented autocomplete function and a contextual view of the selected terms. In addition, SPUD semantically lifts the metadata already in Dublinked through the techniques presented in [5]. A panel gives an overview of the available datasets and allows navigation based on publisher, categories, provenance etc. Given the wealth of information in a city, retrieving related datasets is another important capability of SPUD. This is done based on the semantic technique described in [1].

4.2. Data integration

SPUD is using data integration techniques for linking data (using standard techniques known from LOD) and for semantically processing semi-structured input from social media. For example, one source of information in our scenario is the social media data obtained from the LiveDrive traffic update service\(^13\) that provides information about the city in the form of messages. This data is lifted into an events ontology. A hierarchical clustering technique, which we refer to as the taxonomizer, is used to generate a domain-specific sub-ontology. Tweets are mapped to this ontology automatically. In turn, this information is used by the diagnosis component. (see [2] for details).

4.3. Social Media Mining

SPUD merges city data sources with social media in order to capture relevant insights providing real-time explanations of abnormal traffic conditions, such as delays. Natural language processing

\(^9\)http://doc.esd.org.uk/IPSV/
\(^10\)www.w3.org/TR/vocab-dcat/?
\(^11\)www.w3.org/TR/void/?
\(^12\)www.w3.org/TR/prov-o/
\(^13\)https://twitter.com/LiveDrive
is employed to geocode these user contributed updates and reports are classified into events such as accidents, break-downs and other unplanned traffic obstructions. When an anomalous congestion is detected, social media updates in proximity which might explain the delay are surfaced (see [2] for details).

4.4. Trajectory Miner

The trajectory mining app is using geo-located tweets to mine user augmented trajectories and give insight to the distribution and mobility of citizens. The application visualizes the intensity of users (tweets) activity in each specific region every 15 minutes. In addition, the origin-destination flow is mined and visualized with its associated tweets for different times to day, and particular events, illustrating insight that is typically not captured by government sources such as censuses (see [3] for details).

4.5. Privacy

Privacy-preserving data publishing [7] is of great importance when dealing with city data [5]. In SPUD, we support vulnerability identification, data masking and anonymisation tools that enable data publishers to protect their data from re-identification and sensitive information disclosure attacks, prior to data publishing. SPUD operates by first identifying privacy vulnerabilities in the data and then sanitizing the data based on the datatype and the intended purpose of use. For example, SPUD can detect unique values in columns of data such as phone numbers, name/surname combinations or latitude/longitude combinations. For the last case, SPUD can group locations based on proximity so that, for any given location, there are at least $N$ different records.

4.6. Diagnosis

For the traffic diagnosis component, SPUD compiles off-line all historic diagnosis information into a deterministic finite state machine, following the structure a road network. The latter state machine is augmented with respect to all RDF-described events, road works, incidents where a subset of them are connected to historic traffic congestions and the probability with which they have indeed caused it. Pure AI diagnosis approaches [6] are not able to retrieve any diagnosis result of quasi real-time conditions (e.g., events or road works for which we do not have any historical records) if the latter do not exactly match at least one of the existing historical conditions. We tackle this problem by means of existing semantic techniques and define a matching function for matching new Description Logics concept-based real-time conditions and historic conditions. Conditions, defined along city events, road works, incidents, are all represented using existing vocabularies such as DBpedia, SKOS, Talis Address Vocabulary\(^\text{14}\), basic geo vocabulary and internal IBM ontologies for handling basic generalization/specialization of new and historic conditions. The semantic similarity function is based on the matchmaking functions introduced by [8] and [9], while concept abduction [10] is used for retrieving the difference between real-time and historical event descriptions, and then used for reporting back the results of diagnosis reasoning. The overall approach is inspired and adapted from case-based reasoning (see [4] for details).

5. Deployment

We present a high-level architecture of the components and technologies presented in the previous sections in Figure 3. The main elements of our architecture are a set of APIs (mostly REST), where an HTML5-based front-end interfaces, a IBM WebSphere Application Server, where the main application logic and the Enterprise Apps such as the Diagnoser and the Trajectory Miner are running, a publishing container to facilitate transfer of large files and an enterprise SAN for file storage. We, have used the IBM DB2 RDF store, since recent experiments have shown that it has performance advantages over competing solution for dataset sizes similar to those in this paper (see [11] for details). IBM Tivoli Access Manager and WebSEAL is used to provide secure access to the API. We are using open-source libraries such

\(^{14}\)http://schemas.talis.com/2005/address/schema
as Apache POI and PDFBox to maintain full-text indexes for files and expose this functionality using custom predicate handling in our SPARQL endpoint.

Our technology stack is based on well-established commercial software from IBM. Critical components (HTTP/Application Server, RDF Store, Storage) can be clustered as required to ensure scalability and robustness.

The input for our system comes from many sources. A large corpus of datasets, published by a number of different government authorities has been retrieved from dublinked.ie. Due to the distributed nature of the publication process, some datasets needed to be joined, since they would cover the same information for different constituencies in Dublin (e.g. there are four different datasets regarding lighting pole locations, published in four different formats by four different authorities). In addition, we have used Social media sources such as twitter, along with other Web data, such as event websites. From Linked Data, we have used vocabularies and ontologies such as IPSV, the basic WGS84 ontology\(^ {15} \), PROV-O and VOID, and corpora such as DBPedia and LinkedGeoData. A number of sources used in SPUD are not publicly available, due to privacy and governance restrictions. Thor-

\(^ {15} \)http://www.w3.org/2003/01/geo/
ough descriptions of the datasets used can be found at [1, 2, 3, 4, 5].

Figure 4 shows the main layout of our user interface: On the left, the interface allows the user to explore datasets, based on a series of criteria (category, publisher, area), merge datasets and activate application components (such as the Diagnoser and the Trajectory Miner). The other panes allow for examining data and metadata, doing basic data manipulation and visualizing data on a map. An interface allows searching based on metadata, structure and file contents.

In Figure 5, we show a screenshot from the diagnosis component. For a given anomaly, we are indicating the part of the road network that is identified as relevant. In addition, we display a set of icons corresponding to potential diagnosis results, along with the confidence for the diagnosis. Each of these icons carries additional information regarding the diagnosis (not shown in the figure), such as the time of the event and a link to the source.

In terms of performance and efficiency, our system is easily able to cope with the volume of information in Dublinked, when deployed on a virtual machine with 2 vCPUs and 64GB of RAM. Converting the input files to a simple RDF format results in an expansion by more than an order of magnitude, in terms of number lines and size on disk. The vast majority of CSV files contain between 3 and 7 columns, corresponding to 10-20 triples per line. The Diagnoser has a runtime ranging from a couple of seconds to 20 seconds. Given that the update frequency of the data in the system (e.g. Bus traces) is longer than this, SPUD can support quasi-realtime processing of traffic information from Dublin. A more detailed performance evaluation of our system can be found in the respective publications [1, 2, 3, 4, 5].

6. Related Work

Often, urban data is sourced from legacy non-relational systems or spreadsheets made for consumption by humans. The data is potentially very large, highly heterogeneous, spanning different domains, and with unknown structure (from
static data to spatial-temporal data obtained from physical sensors). Moreover, the users who want to consume these data are not data integration experts and are not necessarily able to query data using structured query languages. We look at existent semantic approaches for processing, integrating and analyzing such data.

Semantic technologies have been proposed to enrich the unstructured information space with ontologically annotated data, to introduce the necessary coherence, organization, data integration and dynamism to reach a highly-effective data model that can be used not only for exploration, but also to perform complex and unambiguous queries. However, converting raw government data to high quality Linked Data is costly [12] and approaches to do that at scale are limited. State of the art semantic approaches for urban data assume the existence of reference ontology(ies) to guide the conversion to RDF [13], or assume the input is a relational database [14], where the first row is used to suggest properties and each row refer to entities. The latest approach is used in the Datalift project [15] to automate the conversion from the source format (e.g. CSV, XLS, SHP) to the raw RDF, before transforming it to well-formed RDF by mapping to selected vocabularies through the use of SPARQL construct queries. The process used in Datalift is similar to ours. Datalift also integrates with the SILK framework for facilitating mappings between datasets. The LOD2\(^{16}\) stack is also offering a variety of tools, supporting a series of information management and integration tasks.

The approach in [16] is based on Google Refine for data cleaning and a reconciliation service extended with Linked Data capabilities to enable exporting tabular data into RDF. However, in our experience, this tool has limited fitness-for-use for the non-expert users.

Open content portals for cities such as London, Chicago and Dublin, to name a few, allow users to explore the relevant datasets by searching through keywords or by navigating through the different categories (e.g. weather stations, airports, arts, demographic, environment, housing, health, recreation). Datasets are searched by names (titles) and semi-structured metadata catalogues, but not by content (column names or values). Users can select a dataset and visualize the tabular data, plot it in a map or chart, and filter by column values, but they cannot aggregate data or refine exploration/queries across sources.

Content platforms for urban data require novel search and exploration models based on a hybrid space of data structure in any format (mostly tabular) and in any domain, and unstructured information, often in the form of short textual descriptions. Combining open data with visualizations can surface hidden insights and trends. However, the extreme heterogeneity and diversity of the relevant data makes it hard for users to discover and consume it. Furthermore, flexible data integration mechanisms are required to support accessing information across datasets.

Traditional data integration architectures, based on creating a common virtual schema for a particular domain, and mapping the data to the schema, cannot cope with the scale and heterogeneity of urban data. Machine learning techniques, although capable of providing high-quality results, typically depend on the availability of training data, which is very difficult to acquire in such an open domain.

IBM City Forward\(^{17}\) is a Web based platform that allows users create explorations across data by selecting cities, topics, and visualization types. The explorations are restricted to a set of a-priori predefined features and categories present in data from selected cities. It does not allow users to dynamically upload data or to create views by refining, selecting and combining data across diverse datasets and attributes. A similar approach to IBM City Forward is taken in the WatchDogs video game\(^{18}\), where an appealing user interface is designed based on open data for a number of cities.

The approach in [17] extracts structured data from tables on the Web, and it allows to cluster

\(^{16}\)http://lod2.eu/

\(^{17}\)http://cityforward.org/

\(^{18}\)http://wearedata.watchdogs.com/
schemas that are semantically closed (based on the probability of seeing two certain attributes appearing together in a table), to suggest schema autocompletion to the users, and to propose semantic mappings between schemas, for example by identifying tables containing uniform data types.

Google fusion tables\(^\text{19}\) enables users to upload (tabular) data and to visualize it in several ways (maps, timelines, and other charts), along with the ability to aggregate data across sources. It does not require the user to declare a schema upfront, but the burden to explore the relevant sources is shift from the system to the user. There is no semantic meaning or description associated to the datasets, column names or values (no legends), and no mechanisms to help with the discovery and ranking of related datasets, or for exploring and redefining views within datasets according to user needs (e.g., spatial or content based queries).

Diagnosis has been largely studied by the Semantic Web community, but mainly in the context of an ontology (e.g. in [18]). There are no other approaches that integrate semantic and diagnosis techniques. However, this integration is needed to handle an open set of events and observations such as the ones considered in this work. They all assume a closed world scenario where the set of possible causes that could explain the effects is well defined and where cause-effect relationships can (at least with unlimited computational resources) be established. The closest diagnosis works to our approach are the ones that tackle the complexity problem of diagnosis approaches [19] by precomputing diagnosis results for some anomalies. If other anomalies are detected some machine learning methods are used to estimate the diagnosis result [20]. However, this estimation consists only of a numeric value rather than an expressive (semantic) explanation as in our case. Furthermore these approaches consider only the problem of mapping anomalies to well defined sets of possible causes rather than to new causes as in our case. In the context of a city, and in our particular use-cases, this is not sufficient.

7. Conclusions and Future Work

In this paper, we have presented an end-to-end semantic approach to extract interesting business results for a combination of open datasets, proprietary datasets and social media, all pertaining to urban information. We have illustrated that Semantic Technologies are indeed applicable to complex business problems, and can cope in scenarios such as the one presented in Section 2 with acceptable performance overheads, at least for cities in the size of Dublin and a focused domain. We have outlined a series of key technologies that enable our platform, spanning different domains.

There is an abundance of research to be pursued in this area: Federated querying across cities; integration of streams and social data which can be merged with traditional sources to provide a richer and more complete view of the city; natural and user-friendly query building that scales; extensions for efficient geo-spatial processing; seamless integration with applications that consume the data by re-exposing the semantic data through legacy interfaces; additional enterprise applications that operate at the semantic level.

References


\(^\text{19}\)http://tables.googlelabs.com


