Guided Exploration and Integration of Urban Data

Vanessa Lopez, Spyros Kotoulas, Marco Luca Sbodio, Raymond Lloyd
Smarter Cities Technology Centre, IBM Research, Ireland
{vanlopez, spyros.kotoulas, marco.sbodio, raylloyd}@ie.ibm.com

ABSTRACT
Governments and enterprises are interested in the return-on-investment for exposing their data. This brings forth the problem of making data consumable, with minimal effort. Beyond search techniques, there is a need for effective methods to identify heterogeneous datasets that are closely related, as part of data integration or exploration tasks. The large number of datasets demands a new generation of Smarter Systems for data content aggregation that allows users to incrementally liberate, access and integrate information, in a manner that scales in terms of gain for the effort spent. In the context of such a pay-as-you-go system, we are presenting a novel method for exploring and discovering relevant datasets based on semantic relatedness. We are demonstrating a system for contextual knowledge mining on hundreds of real-world datasets from semantic relatedness. We are evaluating our semantic approach, using query logs and domain expert judgments, to show that our approach effectively identifies related datasets and outperforms text-based recommendations.

1. INTRODUCTION
Since the first data.gov initiative was launched by the US government, many city agencies and authorities are increasingly making their data accessible through content portals to promote innovation, transparency and drive economic return, by inviting the community at large to explore how new value can be created from existing datasets and their novel combinations. A critical question for government agencies is what return-on-investment they are getting for resources spent in making their data open. The urban data emerging from such sources may be used to support various operations such as exploration, discovery, visualization, merging, querying, and forecasting. Nevertheless, the cost associated with integrating all of this information is prohibitive. Any approach that attempts to leverage and fuse this data cannot rely on building a complete model of the data ahead of time. Our claim is that semantic technologies can be used to lower the entry cost to accessing the highly heterogeneous information of a city.

In [8] we proposed QuerioCity, a Linked Data platform to publish, search and link city data. We initially create a semantic catalogue describing the content of the datasets, improving interoperability and discoverability by using standard vocabularies, linking to external sources and enabling rich queries over this meta-information. Then, we combine this with full-text ranked searches using Lucene [16] indexes. However, this is not enough to enable queries that span across datasets. As described in [8], a set of challenges in managing urban information remains: the openness of the domain, the lack of a common model, the diversity of the data formats, the volume of information and the requirement for a low entry threshold. To address these challenges, we have qualified the advantages of using an incremental semantic approach based on a graph data model to go from raw data to structured data consumption, eliminating the need for tight integration imposed by relational databases.

Building on QuerioCity, we developed a next generation content aggregation platform to address key business challenges for urban information management. Unlike other approaches, the cost of entry is minimal (i.e. datasets can be imported as they are), and processing (entity extraction, annotation, linking, integration) can be done incrementally, while fully exploiting the power of a Web-wide wealth of semantic resources, rich in meaning and structure. In this scenario, a critical issue remains to be addressed: identifying relevant datasets to guide the exploration and integration processes. As we will show, given the complexity of the domain, this lack of semantics and a-priori defined schemata make text-based approaches weak in this environment. In this paper, we describe a novel approach and components for guiding the exploration and integration of urban data. Our approach combines semantic, spatial and textual information to feed a pay-as-you-go system that allows users to create meaningful views over data by discovering and fusing related data as needed. We have integrated these components into an exploratory interface that allows users to navigate, search, query, and visualize the content of existent data views, while at the same time incrementally building new knowledge through users validating annotations, publishing new datasets, or creating new data views by merging existing ones. We validate and demonstrate our approach using hundreds of real-world datasets, published in dublinked.ie by four local authorities in Dublin.

The paper is structured as follows: related work and open issues are presented in Section 2. The approach to build and manipulate knowledge from raw data is described in Section 3. The components for mining semantic annotations and dataset recommendation are described in Sections 4 and 5 respectively. The interface and experiments to showcase the platform are presented in Section 6. We conclude with in Section 7.

2. RELATED WORK AND OPEN ISSUES
Often, urban data is sourced from legacy non-relational systems or spreadsheets made for consumption by humans. The data is highly heterogeneous, spanning different domains, and with unknown structure (including spatial-temporal data from sensors).

Converting raw government data to quality Linked Data is costly [4] and approaches to do that at scale are limited. State of the art approaches for urban data assume the existence of reference ontology(ies) to guide the conversion to RDF [1], or assume the input is a relational database [6], where the first row is used to suggest properties and each row refer to entities. The approach in Datalift [11] automates the conversion from the source format to the “raw RDF”, before transforming it to “well-formed” RDF by mapping to selected vocabularies.

The approach in [9] 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[17], Chicago[18] and Dublin[19], allow users to explore the relevant datasets by keyword searches or by navigating through the
categories in the 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 short textual descriptions). Traditional data integration architectures, based on creating a common virtual schema for a 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 quality results, typically depend on the availability of training data, which is difficult to acquire in such an open domain.

The approach in [3] extracts structured data from tables on the Web, and it allows clustering schemas that are semantically closed (based on the probability of seeing two certain attributes appearing together in a table and by identifying uniform data types), to suggest schema autocompletion to the users. Google fusion tables [20] enables users to upload tabular data and to visualize it in several ways (maps, timelines and charts), along with the ability to aggregate data across sources. It does not require the user to declare a schema upfront, but the burden of exploring the relevant sources is shifted from the system to the user. There are no mechanisms to help with the discovery and ranking of related datasets based on content, or for redefining views within datasets according to user needs.

The challenges are twofold: On the one hand, the design of frontends that support users in expressing complex information needs and explorations going beyond hand crafted facet metadata or domain independent clusters of information, with no explanations given of how items relate together. On the other hand, there is little proactivity/guidance to discover semantically related sources or tables based on the coverage and content of the datasets to create novel aggregated views.

3. APPROACH

In what follows, we describe the approach adopted as part of our content aggregation platform to incrementally transform raw city data to meaningful linked data, at enterprise scale. The components presented here exploit a combination of select and construct SPARQL queries, structured data indexes and off-line pre-computation to operate within the response time constraints of the interactive scenario.

3.1 Publishing, cataloguing and indexing

Following Linked Data principles, we previously created a metadata catalogue to index and search heterogeneous city data coming from the DubLinked Web portal, as described in [8].

The RDF metadata catalogue is automatically generated and updated by cleaning the original metadata, standardizing datatypes, defining entities and ranges, and linking to authoritative and external sources on the Web. Currently, we have focused on the IPSV [7] vocabulary used by UK public sector organisations, although the architecture is open to any vocabularies. The metadata describing the datasets is converted to RDF by reusing existent vocabulary (such as DCAT [15] and Dublin Core Terms [14]) and existent entities matched in the metadata or IPSV hierarchy, or by creating new ones. The added value in creating consistent and standardized metadata is that datasets are linked through shared entities representing categories, publishing agencies and regions, among other, thus improving data consumption and publishing. In addition to semantically mapping and sanitizing the meta-data, full-text indexes over the metadata and files containing textual information (e.g. PDF, CSV, Office) are maintained. These indexes represent the least common denominator for processing data (since it is practical to convert most file formats into indexable text). A custom ranking method that considers matches in the metadata and Lucene scores is employed to combine the semantically augmented information with lexical matching.

3.2 Entity extraction and knowledge building

In this Section, we are giving a brief overview of our incremental knowledge building approach, summarized in Figure 1. Datasets are originally represented in the system as files, linked to the metadata catalogue. Depending on the type of the file, the system converts them to an internal common representation that allows incremental annotation and entity extraction. For example, tabular files and their structure are represented as RDF as sets of columns, rows and cells: 

\(<c1, a, Cell>, <c1, col, x> <c1, row, y> <c1, value, "value">\). In this manner, the original structure of the "raw" tabular data is stored in the triple-based repository, regardless of whether it is possible to automatically convert the files to entities (the next step). Additionally, indexes are created to allow full text searches combined with structure information. E.g. the system can return the exact row, and column where matches were found, searches can be scoped to a given group of cells (e.g., all cells matching “Name” in row 0) or according to a proximity criteria (e.g. cells in row 0 when row 1 contains numbers).

![Figure 1. Knowledge building approach.](image)

The next step is to identify entities. Depending on the structure of a file, entities can be contained within each row, each column, each cell etc. In addition, each entity is associated with ancillary information, such as values for headers. The methods for identifying entities go beyond the scope of this paper, so, here, we will assume entities are entirely contained within rows of a normalized tabular structure and are also associated with a header row. For example, an entity could be represented in RDF as 

\(<\text{E}, \text{type, Entity}>\), \(<\text{E}, \text{inRow, Row1}>\), \(<\text{E}, \text{headerRow, Row0}>\).

After identifying entities, the system can start building knowledge on top of them through sets of annotations, and by extracting properties and types for each entity, as explained in Section 4.

Finally, given the complexity of the domain, the system supports capturing meaningful semantic views over the data. For example a user might want to save a data view that contains only the entities with a given type in a given location, so as to facilitate use of the related information by applications.

The underlying data model for the above process consists of a set of immutable named graphs that are referenced by a separate data management model. For every step of the above process, a separate graph is generated and querying implies querying the union of these graphs. This avoids data duplication and allows having multiple possible interpretations of the same data and tracking provenance (using the latest W3C working draft).

3.3 Manipulating data views

Contrary to traditional database aggregation platforms where an administrator integrates databases with great precision and great cost, in our scenario, the fusion of datasets or data views is relatively imprecise and inexpressive, allowing users to perform partial integrations by selecting a subset of the tables (i.e., set of
given columns filtered by a given row value).

The approach taken in our system is an architectural hybrid between database views and an ETL process that materializes the relevant data and loads it in a data store. As outlined in the previous section, data is stored on named graphs. Each data view references a set of named graphs and each named graph may be referenced by multiple data views. Merging two data views essentially consists of creating a new view that references all the graphs of the merged views. Common RDF properties will naturally already be “integrated”. Other properties may be mapped according to user input (or any schema matching method) by creating new graphs where this mapping is expressed. Analogously, entity matching methods can be used to match entities and store “sameAs” relations in a new named graph. Extracting a sub-view from a view is achieved by splitting each relevant graph into two parts. The original data view will now reference both parts, while the sub-view will only reference the part that matches the selection criteria for the sub-view.

4. SEMANTIC ANNOTATION

Data is semantically enriched and incrementally linked by providing common “anchors” from relevant external collections on the Web. A set of annotation components assigns semantic types and meaning to table headers by using both a rule-based mechanism, to recognize commonly used concepts, or by finding matching entities in Linked data sources such as DBpedia.

There is a distinction between annotations and candidate annotations (or recommendations/suggestions). Annotations are those that the system is confident enough about, and therefore the extracted entities are populated with the corresponding value for the given annotation. This is the case for rule-based extracted annotations, e.g., if we have a header labeled “LAT” and the corresponding cell values are decimal numbers between -90 and 90, the system has high confidence that this refers to geographical latitude. Candidate annotations require to be validated by the user. For example, we have a possible property mapping but the property label does not exactly match the column label and/or the domain or the range of the property does not match. The system helps the user verify annotations and resolve ambiguities by proposing a set of attributes with a drop-down list of candidate entities. By choosing one of the suggested item types, the user can create an annotation of the chosen type.

As such, we take an incremental approach to data integration based on user interaction, meaning that our data could become ontologically inconsistent. For example, it is possible that by mapping a column to a given property selected by the user, the range constraints of the property will be violated. Domains / ranges could be taken into account to filter out inconsistent annotations. However, this would have a negative impact on recall, and it does not take into account what users might do with such data. Therefore, we adopted a practical approach that differentiates between annotation types and ontological properties. As such, column headers can be annotated with any kind of ontological entity (properties, classes or instances). If the annotation corresponds to an actual RDF property, and the values assigned on the table satisfy the range of the property to a certain extent, which accounts for noisy data and empty cell values, the annotation becomes also an RDF property for each entity on the table (populated with the corresponding value on the table). Otherwise, the annotation is kept as such, as it is an indication of the type. In the future, different heuristics can be used that take into account the annotation types to normalize the table or further convert table cells into a complete RDF representation, but for the moment, it is still relevant to understand datasets. For example, in the dataset about fats, oils and greases licenses granted to food service establishments, there are several columns denoting the type of establishment (hospital, restaurant, etc.) through a boolean value, as well as their location. The column header “hospital” has as a candidate annotation the entity dbpedia:Hospital, which the user selects as valid. This annotation is useful to discover this dataset in the context of a user looking for locations of hospitals, even if the purpose of the dataset is not to explicitly list hospitals.

The system has a modular architecture with regard to annotation and mapping, currently it is able to extract the following entities:

Spatial entities: a rule-based mechanism is used to search for table headers representing geographical coordinates in the structure indexes, using regular expressions to recognize all form of keywords such as “lon”, “lat”. In addition, if no lat/lon values are specified, geocoding (based on open street maps for Dublin) is used to automatically extract geographical coordinates from addresses, which may span across more than one column. Lat/lon values are populated for all spatial entities using wgs84 [21].

Entity labels and subject: we use a rule-based mechanism to search for columns representing the label (by recognizing column headers such as “Name”) of the entity. If found, rdfs:label properties are created for each instance with names. Often there are no external resources to adequately describe the type (rdf:type) of the instances described in a table. However, the metadata titles often describe the subject of the entities in the tables accurately. Title values are literals, represented by text. For example, the dataset with title “street furniture licenses Dublin City” describes licenses given to businesses wishing to place table and chairs on the pavement. No external entities were found representing street furniture licenses, however, from the title we can extract the entities in common between DBpedia and the IPSV hierarchy, namely: street furniture and license. These entities are used to populate the property dcterms:subject for all entities extracted and described in the dataset.

Entity properties: external vocabularies are used to find adequate meanings for the table headers. To go beyond syntactic mappings, the metadata text fields (the description provided by the publisher) often provide the context to disambiguate and filter relevant entities that describe the content. To extract entities from unstructured descriptions we use DBpedia spotlight [10]. Then, we search if any of those DBpedia entities matches the row headers. If no matches are found for the headers, then we search for syntactic matches (entities with similar labels to the headers) on SPARQL end points [13], the IPSV hierarchy and existent metadata entities. Syntactically ranked matches can be ambiguous and noisy, thus, we need the user to validate these suggestions.

The breadth and domain variations of the data, as well as the diversity of publishers, make it challenging to create a comprehensive and clean annotated model. However, entity extraction over descriptions and reference semantic schemas like IPSV are useful for meaningfully annotating urban data.

5. IDENTIFYING RELATED DATASETS

The main challenge for data source discovery is to be able to determine whether a particular data source is relevant for a given task and contains data that should be integrated with other sources. For example, in the context of a task where a decision about levels of environmental pollution in the city has to be taken, two different data sources, one containing information about noise and another one traffic measurements in the same region, are both relevant. There are several semantic relatedness measures that can influence this decision, among them: (1) annotations are used to identify correspondence between the content described on
different data sources; (2) metadata is used to identify correspondence between topics; (3) entity co-reference is used to find data sources that provide information for values corresponding to entities in the other source.

5.1 Annotation-based Semantic Similarity

Datasets can be related based on the content-based annotations (properties) or candidate annotations they have in common. Given datasets selected by the user this service gets all (non-spatial) annotations and candidate annotations, and for each one it uses a SPARQL query to find all data views with the same annotations. It provides justification of why the datasets are semantically related. Therefore, it retrieves, together with the URI of the relevant data view, the provenance of the matching(s), that is the URI of the cell associated to the annotation (i.e., the column header), the original value of such cell, the annotation and its type (annotation vs. suggestions). Additionally, an annotation could be linked to entities in the catalogues, in this case the provenance (justification) of the matching also includes the URI of the metadata entry and the metadata entity matched to the annotation. For example, the dataset describing the play areas is related to the parking meters dataset, because the former contains annotated information about availability of disabled parking near by.

Lastly, for spatial annotations, a SPARQL query can extract all entities with latitude and longitude values inside a bounding box. The user can further refine the query by filtering entities belonging to datasets in a given category, e.g., health and safety; as such it can plot in a given area on a map all spatial entities representing fire stations, health centers and garda stations.

5.2 Metadata and content topic correlation

The previous method searches for related datasets based on the content, as described by annotations. Using this method, two disparate datasets like health centers in Dublin and the register of fats, oils and greases licenses issued to food service establishments in Dublin and containing the addresses for hospitals, are identified as related because of the annotation overlap. However, datasets may use different annotations to describe related topics. Here, we look at topic correlation.

Besides looking at the co-occurrence of metadata entities used to describe categories and keywords, we need to compare the subject of the entities (given by the dcterms:subject annotation as explained in Section 4). To do that, this method first extracts for a given dataset all metadata entities described using IPSV entities, including the broader and narrower entities in the IPSV hierarchy, and the subject (as given by the dcterms:subject property, if any).

Then, given these set of entities it searches all data views by (1) content: data views containing any of these entities as annotations, recommendations, or subject; and (2) metadata: data views sharing a similar set of entities to describe their metadata. As such the “Fire brigade and ambulance call outs” dataset is related to the “Fire stations” dataset because they share the same DBpedia and IPSV entity (“Fire”) as the value of the property dcterms:subject, despite describing different content (no common properties) and being described with different metadata keywords.

5.3 Co-reference of entities and correlation

Datasets with different annotations and about different topics may describe the same entities (e.g., from different points of view). If the entities in the selected data view (where each entity corresponds to a row in a table) have been annotated with a label, besides doing a syntactic co-reference of entities across data views using the property label, this service also searches for data sources with other properties with values referencing some of the entities in the source dataset. As an example, the “health centers” and the “on street disabled parking bay” datasets are related because a significant number of the instances of health centers are described as locations for parking spots. Pairwise comparison of entities would make the complexity of the procedure quadratic with respect to the input size. In order to avoid that, candidate matches are selected using a SPARQL query, which retrieves the data views and entities with properties whose values match to any of the entity labels described in the original dataset. In practice, this means that instead of employing a common technique from information retrieval (such as blocking), we are relying on the query optimizer of the store to limit the number of comparisons.

5.4 Combined Ranking

Depending on the number of results, an unsorted list of recommended datasets, together with supporting evidence, may be beyond what a user would be willing to examine. In this Section, we propose a ranking of results based on combined results from the previous methods. The aim of the ranking measure is to assign a relatedness score to each dataset with respect to the input dataset, based on the relevance weights assigned to the different results obtained through different approaches. The list of datasets ranked in the first 10 positions represents the best subset of results according to the combination of the different semantic measures. The different nature of the approaches dictates using weights when combined. For the combination strategy we have used the weighted Borda method [2], in which votes (in this case, matches or entities co-occurrence for a dataset) are weighted taking into account the nature of the semantic measure. Weights were given by empirically testing the results obtained using the different criteria and the following assumptions:

- Co-occurrence of annotations (@annotation) and candidate annotations (@recommendation). The weight of matches for [user-validated] annotations is higher than then one of recommendations.

- Co-occurrence of entity labels (@label) across datasets and number of entities in the original dataset described as values of ad-hoc properties in other datasets (@value). The number of overlapping entity labels, and therefore entities, has a higher weight than the number of labels matching values across datasets.

- Topic similarity between annotations and metadata entities (@metadata), such as category and keywords; and similarity of the entity for the entities described in different datasets (@subject).

The score is computed using the Borda count, in which each candidate data view gets a certain number of points by counting the weight given by each match (voter). The combined weight for the matching i within the related data view d is computed as:

\[
W_{d,i} = 1 \times (x = 1, i \in \text{@metadata}) + 2 \times (x = 1, i \in \text{@subject}) + 2 \times (x = 1, i \in \text{@annotation}) + 1 \times (x = 1, i \in \text{@recommendation}) + 2 \times (x = \text{n\_cells} \times 1, i \in \text{@label}) + 1 \times (y = 1, i \in \text{n\_cells} + 0.1, i \in \text{@value})
\]

Once all matches have been counted, the candidate with the most points is the winner. However, we normalize each score i to be distributed between 0 and 1, by taking the score of the winner (S_{max}) and the lowest score (S_{min}) and calculating the range between them (S_{max} - S_{min}) using the following function:

\[
S_{c\_\text{normalized}} = \frac{|S_i - 1| - |S_{\text{min}} - 1|}{\text{Range}}
\]

6. VALIDATION AND EVALUATION

In this Section, we are providing a validation of the approach and components described in the previous sections by means of a prototypical user interface. In addition, we are performing an evaluation of the main contribution of this paper, the semantics-based suggestion of datasets, as described in Section 5.

6.1 Exploratory Interface

Value is not in the raw data alone but on the interaction with the
users, providing proactivity and guidance to increase user engagement, while at the same time creating and making sense of the integrated information through user-centred interactions. We developed a set of REST services to expose the core functionalities of the system. All REST services can produce either XML or JSON, so as to simplify integration with other back-end components, and with a prototypical AJAX presentation layer. We have developed a Web demonstrator to act as a guide on how to use the backend services [12]. It presents to the user a dataset explorer that gives a hierarchical view of the datasets based on linked data (categories, publishers and regions) combined with a search box. As a result of a search, the system presents the relevant datasets highlighting the results, and allows a simple faceted-filtering by further clicking on dataset explorer.

Clicking on the relevant datasets will open the respective data views. Datasets also link to each other through common entities. Therefore, following links brings up new tabs. Right-clicking on a dataset brings up a context menu that allows the user to load the tabular data, download the original file, or visualise its spatial entities in a map. Tabular data is dynamically generated from their RDF representation. When the user right-clicks on a column header, the system suggests annotations for that column. When the user selects one of the suggestions, the corresponding column header will change to a URI (linking to the semantic entity representing the suggestion). Embedded into this interface are also dataset recommendations, as the user is navigating through dataset, the system suggests other potentially interesting datasets. Complementary to this, the interface allows the user to merge datasets by dragging into the "merge bin". By clicking the "merge" button, a new data view will appear with the merged data. Merged data views can also be loaded or displayed in a map.

6.2 Experimental Setting and results

The interface is tested with real-world urban data coming from dublinked.ie and published by four different county councils in Dublin. At the time of writing, our corpus consists of 228 datasets represented in 1656 files and more than 200 million RDF triples. After annotating all tabular data, we obtained 575 candidate annotation and 97 annotations. We test the ability of our system to suggest and find relevant related datasets based on the added value of the extracted semantics. As there is no gold standard corresponding to relevance judging of similar datasets, we perform a user-based evaluation. To find a set of real user queries, we took the keyword searches, from which the user downloaded one of the datasets obtained from her search, from the DubLinked logs. However from these logs we cannot make assumptions about whether consecutive searches by the same user are related, or whether the users have been satisfied with the results. Therefore, we manually evaluate whether the suggestions to discover related datasets are reasonable, and compare its performance with respect to a non-semantic baseline based on Lucene fuzzy searches.

In the non-semantic baseline, a user can find related datasets by using a combination of: (1) metadata keywords to find related datasets, ranked by the number of keywords in common, and (2) all datasets obtained as result of the same syntactic keyword search (performed by the user before she downloaded the dataset), both over the metadata and content indexes. Datasets with higher number of matches for the user keywords or metadata keywords are ranked first. In the semantic approach datasets are ranked using the combined ranking presented in 5.5, and justifications are given of why each dataset is related, according to each criteria.

The experiments pursue to measure the coverage, the correctness and the relevance. Coverage is measured by counting the total of results. The average coverage for the syntactic baseline is 44, while for the semantic approach is 97. However, this does not say whether the results are relevant, and a large number of results can overwhelm the users. To evaluate correctness and relevance we have engaged three different evaluators. Each of them has evaluated the relevance and assigned a score with a discrete value in \(\{0,1,2\}\) for the first ten results, where each number implies:

0: the proposed dataset is based on semantically incorrect justifications, i.e., due to an ambiguous annotation.
1: the proposed dataset is based on semantically correct justifications, but it is not relevant (i.e., the content of the dataset is not specialized enough to give any extra relevant information).
2: the proposed dataset is correct and relevant to determine or complement the information of the original dataset.

Given the three user’s 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. We computed the average over the first 10 results for 30 datasets. For the semantic approach 82% were considered correct and 65% relevant enough. On the syntactic baseline, the respective scores were significantly lower: 60% correct and 43% relevant.

<p>| Table 1. Comparison between baseline and Semantic approaches |</p>
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Baseline</th>
<th>Relevant</th>
<th>Correct</th>
<th>Semantic Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 total</td>
<td>130 (43%)</td>
<td>102 (60%)</td>
<td>146 (65%)</td>
<td>247 (82%)</td>
</tr>
</tbody>
</table>

There was a moderate/ substantial agreement among users. Fleiss' kappa statistic [5] measuring user agreement was \(k=0.6\). Values of kappa can range from -1.0, indicating perfect disagreement below chance, to 1.0, indicating perfect agreement above chance. We observed discrepancies between evaluators regarding the relevance of datasets sharing entities in common (Section 5.3). For instance, play areas were related to both parks and disabled parking locations with an almost similar score, because of a significant number of entities in common among them (e.g., Malahide castle, Millenium park, etc.). Evaluators rated the former as relevant, while the later was rated as irrelevant.

Incorrect results in the syntactic baseline were usually due to fuzzy searches that produce incorrect matches, for example, *street works* matching with *street lighting*. In the semantic approach, incorrect results are often due to ambiguous suggested annotations. Note that this evaluation was performed before the users validated any of the suggested annotations, we expect the performance of the system to improve over time thanks to user feedback while validating annotations, as validated annotations have a higher weight than candidate annotations and noisy suggestions will be removed. Furthermore, a simple syntactic approach sometimes worked better because a combination of two factors. First, the semantic approach discarded datasets that do not follow a tabular format, but that were voted as relevant in the syntactic approach. Second, datasets with various candidate annotations that resulted to be not so relevant (such as locations, email, website, opening hours) were selected. Even in the cases were the relatedness score was low, such datasets were selected for lack of others with higher scores. Since there are not always enough datasets that are relevant to a given one, we measure if there is a correlation between the score of the semantic results and the user relevance rating. As we can see for the sample results in Table 2, a high score average is an indication of relevant results.

| Table 2. Comparison between the semantic score and results |
|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| | q9 | q10 | q11 | q12 | q13 | q14 | q15 |
| Sem.Score | 4.2 | 5.6 | 11.89 | 14.65 | 67.79 | 8.6 | 3.6 |
| Relevant | 0 | 8 | 9 | 8 | 10 | 9 | 3 |
| Correct | 4 | 10 | 10 | 9 | 10 | 9 | 3 |
7. CONCLUSIONS
Semantic techniques and Linked Data sources, rich in meaning and structure, are used to go beyond semantic cataloguing, full text search, syntactic ranking and visualizations of isolated published datasets to incrementally expand the functionality of information-intensive platforms, while reducing the significant up-front cost associated with traditional data integration techniques. In particular, the components presented in this paper enable a web-scale intelligent content platform to provide guidance to ease the burden of discovering, exploring and integrating new data for users. Semantics and meaningful annotations are applied to describe, discover and integrate raw datasets. Through our experiments, we show that these semantic components can be used in a scenario with hundreds of heterogeneous real datasets spanning multiple domains.

We plan to perform more rigorous experiments, in collaboration with the authorities in Dublin to evaluate the user experience over time while using the tool, and to better capture the user’s intention in the context of complex real user tasks that involve not only single searches, but more complicated manipulation of data views.

References
[13] DBpedia SPARQL end point at: http://dbpedia.org/sparql
[17] London: data.london.gov.uk/
[18] Chicago: data.cityofchicago.org/
[19] DubLinked: www.dublinked.ie