Facetted Browsing of Extracted Fusion Tables Data for Digital Cities

Gianluca Quercini[1], Jochen Setz[2], Daniel Sonntag[2], and Chantal Reynaud[1]

[1] Laboratoire de Recherche en Informatique
Université Paris Sud XI, Orsay, France
[2] German Research Center for Artificial Intelligence (DFKI)

Abstract. Digital cities of the future should provide digital information about points-of-interest (POIs) for virtually any user context. Starting from several *Google Fusion* tables about city POIs, we extracted and transferred useful POI data to RDF to be accessible by SPARQL requests. In this initial application context, we concentrated on museum and restaurant resources as the result of a precision-oriented information extraction part. With the current application system we are able to retrieve, filter, and order digital cities POI data in multiple ways. With the help of facets, users can do more than just browsing the museums and restaurants. They can filter the relevant objects according to available metadata criteria such as city, country, and POI categories. Different views allow us to visualize the objects of interest as tables, thumbnails, or POIs on an interactive map. In addition, any complementary information on the cities where museums and restaurants are located are retrieved from DBpedia and displayed at query time.

1 Introduction

Digital Cities of the Future should feature a democratic city space through a citizen-centric model, which is the vision of the EIT action line Digital Cities[3]. Citizen participation could take different forms, e.g., the execution of necessary actions to improve the city's performance and sustainability, or, as in the direction we pursue, the collection and usage of data to be broadcast, or used to analyse and sense the status and the dynamics of the city as a place where people live and spend their spare-time. As part of the EIT ICT Labs KIC activity DataBridges, Data Integration for Digital Cities, we are developing a framework that enables the enrichment of data related to points-of-interests (POIs) in cities (e.g., restaurants, museums, or theatres) and supports the applications which aim at using the data to provide specific and dynamic city services (e.g., city tour recommender systems). We are confronted with two major challenges on which we will focus in this demo paper:

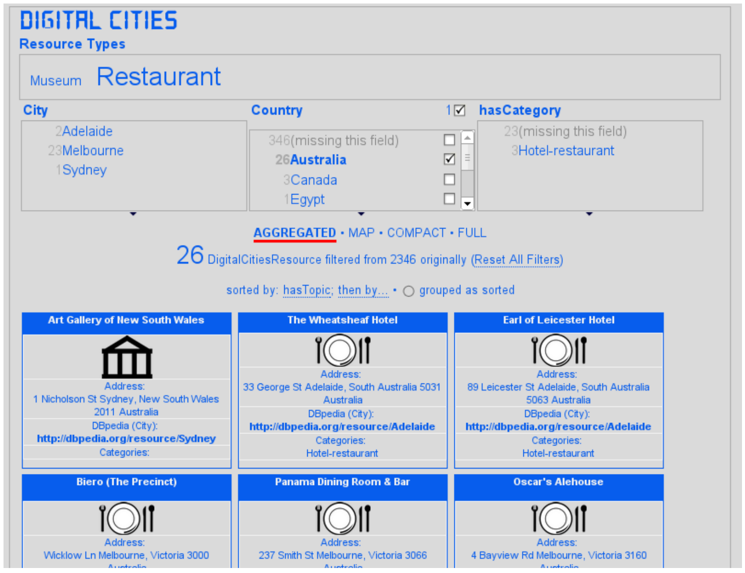
- Collecting as much data as possible about digital city POIs.
- Organizing the data into facets so that it can be easily browsed.

[3] `http://eit.ictlabs.eu/action-lines/`

2 Which Facetted Browsing Functionality do We Provide?

Starting from several Google Fusion tables, the data is being transferred to RDF to be accessible by SPARQL requests. We first concentrated on museum and restaurant resources. With the current demonstrator, we are able to filter and order digital cities in multiple ways, rather than in a single, pre-determined, taxonomic order. With the help of automatically generated facets, users can browse museums and restaurants in an elegant way, but also filter the relevant objects according to the available metadata criteria: city, country, and/or POI categories. Different views allow us to visualize the objects of interest as tables, thumbnails, and points-of-interest on an interactive map. In addition, any complementary information on the cities where museums and restaurants are located are retrieved from DBpedia and displayed at query time.

Figure 1 shows digital cities results of restaurants and museums in Australia. The facets in the upper part allow us to filter the city, country, and category resources (here restaurant types) as indicated. Different types of resources are aggregated into the result lenses shown in the lower part of Figure 1 which displays one museum and five restaurant results.


DIGITAL CITIES
Resource Types
Museum Restaurant

City
2Adelaide
23Melbourne
1Sydney

Country 1 ☑
346(missing this field) ☐
26**Australia** ☑
3Canada ☐
1Egypt ☐

hasCategory
23(missing this field)
3Hotel-restaurant

AGGREGATED • MAP • COMPACT • FULL

26 DigitalCitiesResource filtered from 2346 originally (Reset All Filters)

sorted by: hasTopic; then by... • ○ grouped as sorted

Art Gallery of New South Wales
Address:
1 Nicholson St Sydney, New South Wales 2011 Australia
DBpedia (City):
http://dbpedia.org/resource/Sydney
Categories:

The Wheatsheaf Hotel
Address:
33 George St Adelaide, South Australia 5031 Australia
DBpedia (City):
http://dbpedia.org/resource/Adelaide
Categories:
Hotel-restaurant

Earl of Leicester Hotel
Address:
89 Leicester St Adelaide, South Australia 5063 Australia
DBpedia (City):
http://dbpedia.org/resource/Adelaide
Categories:
Hotel-restaurant

Biero (The Precinct)
Address:
Wicklow Ln Melbourne, Victoria 3000

Panama Dining Room & Bar
Address:
237 Smith St Melbourne, Victoria 3066

Oscar's Alehouse
Address:
4 Bayview Rd Melbourne, Victoria 3160

Fig. 1: Facetted browsing of digital city POIs

Our web application allows us to switch between different views of a single filter being applied, namely aggregated result view, map view, compact view, and full view. Figure 2 shows a result map of 26 filtered restaurant resources. The Google map indicates Australian restaurants and museums as landmarks.

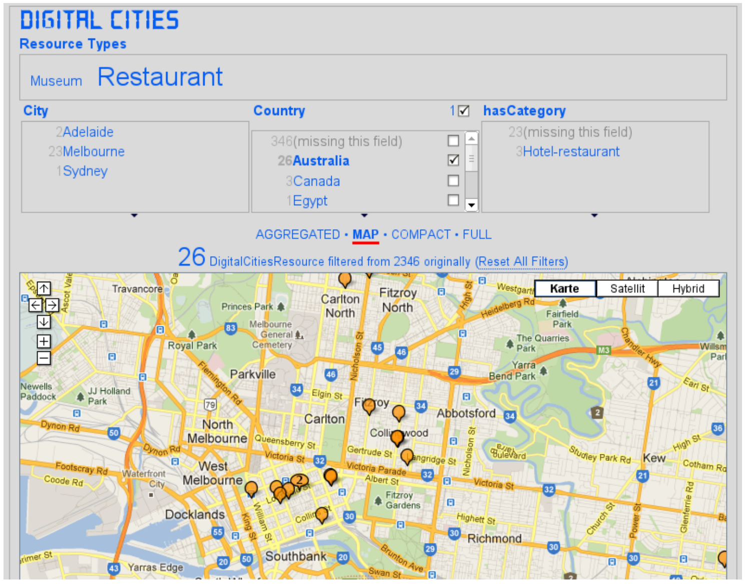
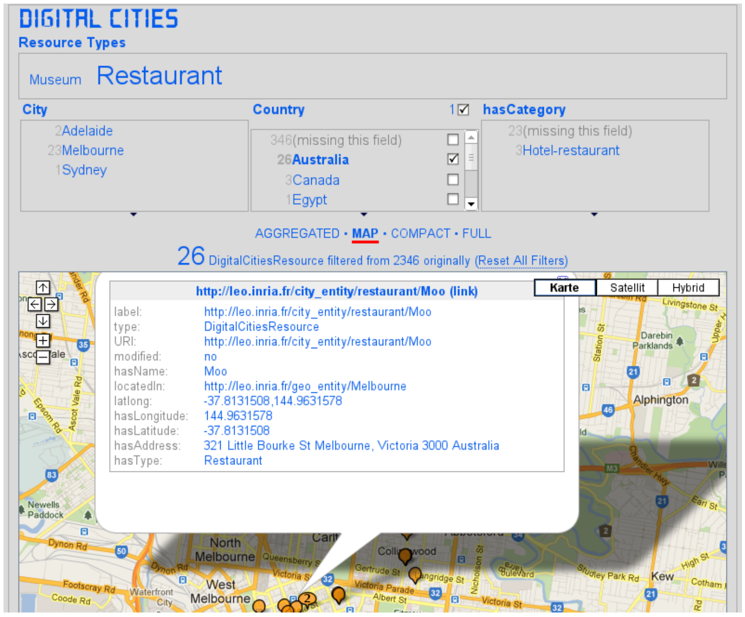


Fig. 2: Digital cities restaurant results indicated on a map

3 Which Features Have We Implemented?

We make use of multiple Linked Data sources; DBpedia contents are extracted according to semantic city links of the filtered resources. Photos related to cities described in DBPedia are provided by the Linked Data resource flickr wrappr [4]. The system's browser page is dynamically generated and updated according to the retrieval sets of the combined Linked Data queries. In addition to interactive boxes, links, and tables, the GUI uses Javascript widgets to visualize individual museum and restaurant retrieval results. External data is provided by a slideshow which enables us to jump directly to the data of interest, here DBpedia comments or Flickr photos. The slideshows can be enabled by selecting the links provided in the aggregated view. Figure 3 shows the selection of a point-of-interest (restaurant) on the map; location-based details of the resource, which are obtained from DBpedia, can be highlighted.

[4] `http://www4.wiwiss.fu-berlin.de/flickrwrappr/`



26 DigitalCitiesResource filtered from 2346 originally (Reset All Filters)

http://leo.inria.fr/city_entity/restaurant/Moo (link)

label:	http://leo.inria.fr/city_entity/restaurant/Moo
type:	DigitalCitiesResource
URI:	http://leo.inria.fr/city_entity/restaurant/Moo
modified:	no
hasName:	Moo
locatedIn:	http://leo.inria.fr/geo_entity/Melbourne
latlong:	-37.8131508,144.9631578
hasLongitude:	144.9631578
hasLatitude:	-37.8131508
hasAddress:	321 Little Bourke St Melbourne, Victoria 3000 Australia
hasType:	Restaurant

Fig. 3: Map details of one of the 26 filtered digital cities results (restaurants)

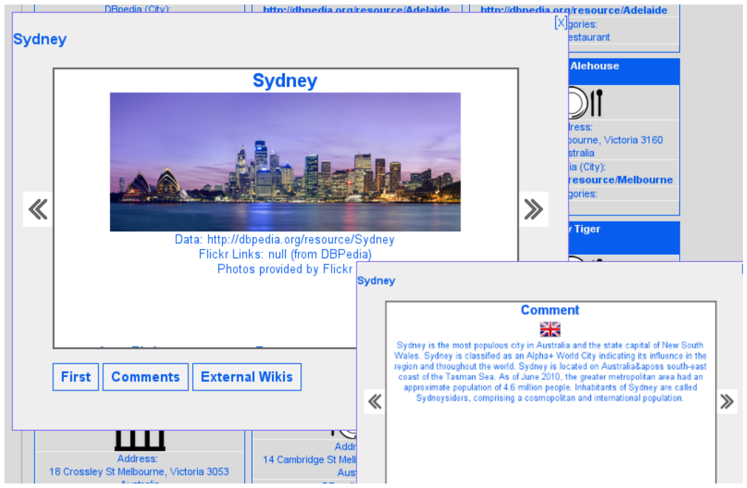
4 How Does the System Work Technically?

Google Fusion Tables (GFT) are data tables which are consolidated into a Web application that hosts a vast collection of tables contributed by people over the Internet [4]. We developed a tool that automatically converts these tables to RDF data. This problem of understanding the semantics of tables has already been addressed by numerous research groups [6,8,14]. In particular the approaches described in [6,14], which rely on probabilistic models, show promising results. Based on our extraction process from GFT tables, we implemented a graphical user interface from scratch by using the open-source knowledge management tool Exhibit[5]. After the facets and lenses have been specified, several GFT tables are converted to RDF and loaded onto our DFKI Virtuoso server. Note that the set of GFT from which data is obtained is pre-determined off-line; as a result, to add further data to the application we would need to manually obtain another set of tables. An interesting improvement of the application would consist in having tables loaded dynamically as they are added to GFT.

The interactive GUI then triggers several SPARQL queries (at query-time) and provides additional DBpedia information about cities of filtered restaurants and museums in multiple languages (according to established DBpedia links). Additionally, we use the web service of *flickr wrappr* at query-time in order to retrieve RDF links to relevant photos of DBpedia resources. Figure 4 shows

[5] `http://www.simile-widgets.org/exhibit/`

the slideshow which we built from the tables. A click on a city location where restaurants or museums are located triggers an ad-hoc query to DBpedia to fetch more information about the city. The slideshow contains comments, external links, and photos from flickr.


Sydney
Sydney
Data: http://dbpedia.org/resource/Sydney
Flickr Links: null (from DBPedia)
Photos provided by Flickr
First
Comments
External Wikis
Sydney
Comment
Alehouse
Tiger
Address:
18 Crossley St Melbourne, Victoria 3053

Sydney is the most populous city in Australia and the state capital of New South Wales. Sydney is classified as an Alpha+ World City indicating its influence in the region and throughout the world. Sydney is located on Australia&aposs south-east coast of the Tasman Sea. As of June 2010, the greater metropolitan area had an approximate population of 4.6 million people. Inhabitants of Sydney are called Sydneysiders, comprising a cosmopolitan and international population.

Fig. 4: Slideshow built from Fusion tables, DBpedia data and flickr photos

4.1 Data Extraction from Google Fusion Tables

All tables in GFT have a relational database structure. Each table is identified by a unique name, which is an alphanumeric string in GFT being automatically assigned to the table upon its creation. Moreover, each column must have a *name* (or *header*) and a *type*, of which GFT defines four: TEXT, NUMBER, LOCATION and DATE. In essence, tables in GFT have three major advantages over other tabular data (spreadsheets, HTML tables, HTML lists) that can be found on the Web:

- GFT tables have a very simple and neat structure. Columns in a GFT table do not branch into several sub-columns, like in spreadsheets;
- The columns of GFT tables are usually typed, which makes easier the semantic annotation of data;
- GFT provides a simple, efficient, and well-documented API that allows applications to query, create, delete, and update tables by using the *Standard Query Language*.

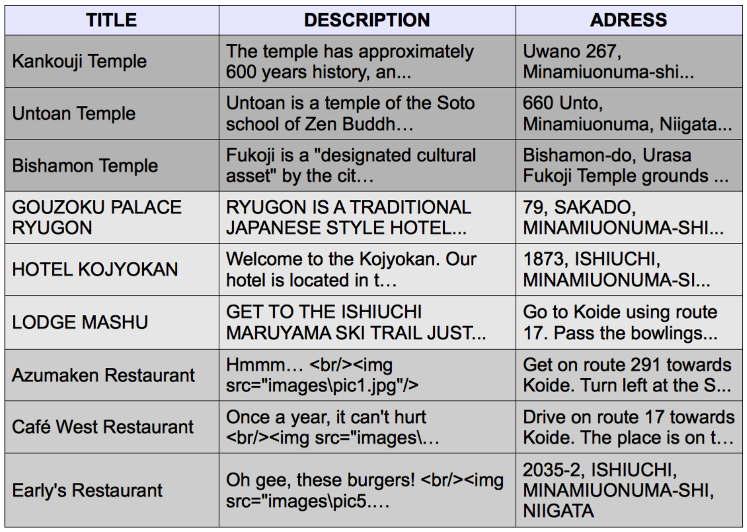
In order to extract data for the Digital Cities facetted browsing context from GFT tables, we first created an ontology which describes major Digital

Cities POIs, such as restaurants and museums in our specific case. The ontology is needed to drive the extraction of important data, such as the name of the POI, its location, contact information as well as its category (e.g. *archaeological museum*, *Italian restaurant*).

The second step consists of selecting GFT tables that contain data about museums and restaurants. The Google web search engine indexes the tables in GFT, which means that they can be searched in the same way as regular Web pages. Therefore, a search for “restaurant” returns all tables in GFT that contain the keyword ”restaurant”. However, the mere fact that a table T mentions the keyword “restaurant” does not necessarily imply that the table has actually data on restaurants.

One of the following may occur:

– T mentions the word “restaurant” only incidentally and therefore has no data on restaurants at all.
– T has data on restaurants along with data on other entities.
– T is entirely dedicated to restaurants.



TITLE	DESCRIPTION	ADRESS
Kankouji Temple	The temple has approximately 600 years history, an...	Uwano 267, Minamiuonuma-shi...
Untoan Temple	Untoan is a temple of the Soto school of Zen Buddh…	660 Unto, Minamiuonuma, Niigata...
Bishamon Temple	Fukoji is a "designated cultural asset" by the cit…	Bishamon-do, Urasa Fukoji Temple grounds ...
GOUZOKU PALACE RYUGON	RYUGON IS A TRADITIONAL JAPANESE STYLE HOTEL...	79, SAKADO, MINAMIUONUMA-SHI...
HOTEL KOJYOKAN	Welcome to the Kojyokan. Our hotel is located in t…	1873, ISHIUCHI, MINAMIUONUMA-SI...
LODGE MASHU	GET TO THE ISHIUCHI MARUYAMA SKI TRAIL JUST...	Go to Koide using route 17. Pass the bowlings...
Azumaken Restaurant	Hmmm… <br/><img src="images\pic1.jpg"/>	Get on route 291 towards Koide. Turn left at the S...
Café West Restaurant	Once a year, it can't hurt <br/><img src="images\…	Drive on route 17 towards Koide. The place is on t…
Early's Restaurant	Oh gee, these burgers! <br/><img src="images\pic5.…	2035-2, ISHIUCHI, MINAMIUONUMA-SHI, NIIGATA

Fig. 5: Excerpt of a GFT table with data on POIs

Figure 5 shows an excerpt of a GFT table which contains information on heterogeneous POIs: temples (first three rows), hotels (the three rows in the middle) and restaurants (the last three rows). Therefore, our extraction algorithm needs to process the table row by row to select those that have the desired information. As shown in Figure 6, the extraction algorithm takes as an input a table T and a list of types of POIs from our ontology and:

- Identifies the *rows* of T that contain information on POIs of any of the input types.
- Determines the *cells* that contain the *names* of those POIs.

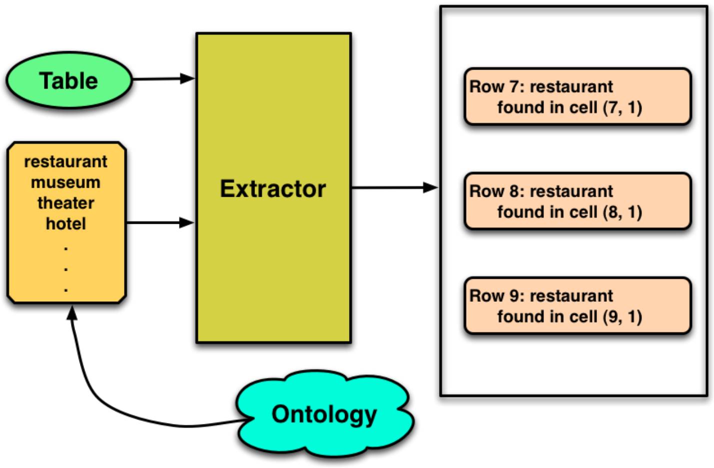


Table
restaurant
museum
theater
hotel
Extractor
Ontology
Row 7: restaurant found in cell (7, 1)
Row 8: restaurant found in cell (8, 1)
Row 9: restaurant found in cell (9, 1)

Fig. 6: Generalized extraction algorithm

As an example, our algorithm correctly identifies that the last three rows of the table shown in Figure 5 have information on restaurants and that the name of those restaurants are in the first column.

Our extraction algorithm goes through three steps:

1. **Pre-processing.** The cells that are not likely to contain names of POIs of the given types are ruled out. This is done by looking at the syntactic properties of the content of each cell, as well as the GFT types of the columns in which they occur.
2. **Annotation.** The content of the remaining cells is submitted to a Web search engine in order to obtain short textual descriptions that are used to determine whether the cells contain names of the POIs of the given types.
3. **Extraction.** The columns that provide the values for the attributes of the POIs are selected and information is extracted.

At the *pre-processing* step, our algorithm rules out the following cells:

- Cells that contain values that follow a certain pattern, that is usually captured by regular expressions: phone numbers, URLs, email addresses, numeric values and geographic coordinates.
- Cells containing long values, such as verbose descriptions (e.g., those in the second column in Figure 5).

– Cells that belong to columns with a specific GFT type, such as LOCATION, DATE and NUMBER.

At the *annotation* step, our algorithm resorts to a Web search engine to understand whether the cells that have not been ruled out at pre-processing contain names of POIs. More specifically, our algorithm submits the content of a cell $T(i, j)$ to the *Bing* [6] search engine and uses the top-10 results as an additional external context to understand the content of the cell itself. Each result returned by the search engine consists of a link to a Web page and a short description (referred to as *snippet*) of the page itself. Our algorithm uses a multi-class text classifier to determine whether a snippet is the description of a POI of a certain type t (for instance, "restaurant"); if the majority of the snippets returned by querying the search engine with the content of $T(i, j)$ are classified as descriptions of restaurants, then $T(i, j)$ is considered as containing the name of a restaurant.

Finally, we need to select the columns that provide values for the attributes or properties of restaurants. The identification of the columns is driven by our ontology, which includes the prominent properties of POIs; in the case of restaurants, these are *name*, *address*, *average price*, *category*, *telephone number* and *website*. In order to match each attribute with the corresponding column in T, if any, we use simple heuristics. The column with the *name* is retrieved while filtering the rows of T; as for *telephone number* and *website*, we use regular expressions; the *address* is usually in a column with type LOCATION and is parsed with a online geotagger such as *Yahoo! PlaceFinder*; finally, the column with the *average price* (respectively, *category*) is considered to be the one with title *price* (respectively, *category* or *type*). Once the columns containing the attributes are determined, the data in the selected rows are extracted from the table and inserted in the ontology.

With this procedure we extracted data on 1500 restaurants, 500 museums, 160 theatres, 67 hotels and 109 schools; only the data on restaurants and museums are already available in our application system.

An evaluation shows the performance of our algorithm to determine whether a cell contains the name of a POI of a given type based on the snippets returned by the search engine. The evaluation has been performed on 40 GFT tables, containing information on POIs with 5 different types: restaurants, museums, theatres, hotels, and schools. In total, we have 287 references to restaurants, 240 to museums, 160 to theatres, 67 to hotels, and 109 to schools. Each table has been annotated manually to obtain a ground truth for our evaluation. We collected names of 300 POIs for each type under examination from DBPedia and used them to retrieve snippets from Bing that we used to train two multi-class text classifiers: a SVM and a Naive Bayes classifier.

The results of our evaluation are shown in Table 1: precision is computed as the number of cells correctly identified by the algorithm over the number of cells identified by the algorithm (i.e., how many of the cells identified are identified

[6] We chose Bing because it provides an API with less limitations than other Web search engines in terms of query allowance.

Type	Method	Precision	Recall	F-measure
Restaurants	SVM	0.89	0.69	**0.78**
	Bayes	0.59	0.80	0.68
Museums	SVM	0.83	0.82	**0.83**
	Bayes	0.45	0.93	0.61
Theatres	SVM	0.83	0.76	**0.80**
	Bayes	0.34	0.89	0.5
Hotels	SVM	0.74	0.89	**0.81**
	Bayes	0.23	0.92	0.37
Schools	SVM	0.96	0.91	**0.94**
	Bayes	0.75	0.96	0.85

Table 1: Evaluation of the algorithm.

correctly); recall is computed as the number of cells correctly identified by the algorithm over the number of cells that contain the name of a POI (i.e., how many of the cells containing the name of a POI are correctly identified); the f-measure is the harmonic mean of precision and recall.

4.2 Facetted Browsing Web Architecture

The facetted web browsing architecture described in this paper follows a line of location-based mobile application frameworks in which the majority is equipped with a full-fledged Web browser that enables us to provide platform-independent graphical user interfaces (GUIs) by means of DHTML-based Rich Internet Applications (RIA) or the like [5,10].

Work in those mobile application frameworks is often concerned with physical location-based issues at query/interaction-time, e.g., supporting wayfinding with tactile cues [9] or interactive experiences for cyclists [12]. Other work concerns mobile search scenarios and incidental information [1] where contexts such as location and time play major roles in information discovery [2] or the design of Web-based mobile services [11].

The Digital Cities facetted web architecture follows a different usage strategy: the ubiquitous access to location-based information from virtually every browser with new information repositories which address POI related information needs on the large scale. Previously, in the context of the query/interaction-time retrieval needs [13], we used Google Maps Local Search [7] and two REST services provided by GeoNames [3] (i.e., the *findNearbyWikipedia* search and the *findNearbyWeather* search). But when comparing different POIs and planning routes and/or restaurant information via the facetted browsing functionality, we need a spatially inclusive and comprehensive set of POIs and category information automatically extracted from other (online) repositories.

The facetted web browsing is realised as a Java-based web-server which provides a HTTP/ReST semantic webservice to the Digital Cities data. The JavaScript client implements the control logic and compose facets, views and lenses to layout them with help of a huge set of widgets.

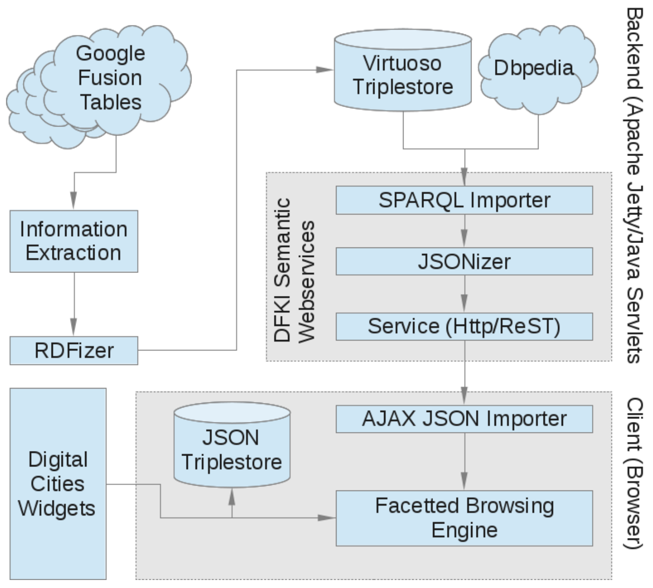


Fig. 7: Technical components of the Facetted Search Web architecture

On start-up, the web-application requests the semantic webservice for available restaurant and museum data. The data is converted and delivered in the Exhibit-JSON format and stored into an In-Memory JavaScript TripleStore provided by the Exhibit framework. The information to be queried for is specified by Exhibit's graph expressions on the Digital Cities RDF graph. Facets to filter the fetched results are defined as HTML-Templates. Filters are defined by Exhibit's path expressions, which allow us to specify paths through an RDF-Graph considering forward and backward properties.

The DBpedia information is provided by the DFKI Semantic webservices which requests the DBpedia SPARQL endpoint at run-time and provide the data as JSON-strings to our semantic widgets. The semantic widgets then sort the DBpedia information about a city by their properties to set up an interactive slideshow. The DFKI semantic webservices in the backend do not hold any application logic, but rather a model-based logic to fetch, store, compose, and convert RDF-data. The RDFizer imports GFT tables from the Information Extraction into RDF triples (NT-Notation) to store the triples into a local TripleStore (Virtuoso) on the server. Virtuoso's native JDBC Interface is used to make the data accessible by SPARQL queries.

Finally, a SPARQL importer fetches RDF data from the DBpedia SPARQL endpoint and the local endpoint by resource-type or URI. Our semantic webservices also provides a converter from RDF into the Exhibit-JSON format. The service component wraps the query mechanism into an application-specific "restful" interface to request for structured museum, restaurant, and city information.

5 Conclusion

The Digital Cities facetted web browsing architecture follows the idea of providing ubiquitous access to location-based information from virtually every browser with new information repositories which address POI related information needs on the large scale. In this paper, we described our live system: starting from several GFT tables about city POIs, we extracted and transferred useful POI data to RDF to be accessible by SPARQL requests in the context of an interactive facetted browsing application. In particular, different views allow us to visualize the objects of interest as tables, thumbnails, or POIs on an interactive map or slideshow which also takes dynamic data from other Linked data sources (DBpedia and flickr) into account. Knowledge extraction from these structured and semi-structured documents on the Web should now be complemented with a special focus on scalability. The facetted browsing tool would highly benefit from further GFT tables to be extracted automatically and used in the context of a digital city search application. Besides further structured data cells, we plan to automatically extract information from textual data cells in the near future. Thereby, the structured data cells of a specific record should improve the performance of the text mining processes of the related unstructured data cells. A demo of the Digital Cities facetted web search can be found at the following URL: `http://digitaleveredelung.lolodata.org:8080/DigitalCities/page/index.html`.

6 Acknowledgements

This research has been supported in part by the THESEUS Programme in the Core Technology Cluster WP4, which is funded by the German Federal Ministry of Economics and Technology (01MQ07016) and the EIT ICT labs activities 2012 The responsibility for this publication lies with the authors.

References

1. Arter, D., Buchanan, G., Jones, M., Harper, R.: Incidental Information and Mobile Search. In: Proceedings of the 9th International Conference on Human Computer Interaction with Mobile Devices and Services. pp. 413–420. MobileHCI '07, ACM, New York, NY, USA (2007)
2. Church, K., Neumann, J., Cherubini, M., Oliver, N.: SocialSearchBrowser: a Novel Mobile Search and Information Discovery Tool. In: Proceedings of the 15th International Conference on Intelligent User Interfaces. pp. 101–110. IUI '10, ACM, New York, NY, USA (2010)
3. GeoNames: WebServices Overview (Jul 2010), `http://www.geonames.org/export/ws-overview.html`
4. Gonzalez, H., Halevy, A.Y., Jensen, C.S., Langen, A., Madhavan, J., Shapley, R., Shen, W., Goldberg-Kidon, J.: Google Fusion Tables: Web-centered Data Management and Collaboration. In: Proceedings of the 2010 International Conference on Management of Data. pp. 1061–1066. SIGMOD '10, ACM, New York, NY, USA (2010)

5. Gruenstein, A., McGraw, I., Badr, I.: The WAMI Toolkit for Developing, Deploying, and Evaluating Web-accessible Multimodal Interfaces. In: Proceedings of the 10th International Conference on Multimodal Interfaces. pp. 141–148. ICMI '08, ACM, New York, NY, USA (2008)
6. Limaye, G., Sarawagi, S., Chakrabarti, S.: Annotating and Searching Web Tables Using Entities, Types and Relationships. Proc. VLDB Endow. 3, 1338–1347 (September 2010)
7. Mapki: (Jul 2010), `http://mapki.com/`
8. Mulwad, V., Finin, T., Syed, Z., Joshi, A.: Using Linked Data to Interpret Tables. In: First International Workshop on Consuming Linked Data (COLD2010) (2010)
9. Pielot, M., Henze, N., Boll, S.: Supporting Map-based Wayfinding with Tactile Cues. In: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services. pp. 23:1–23:10. MobileHCI '09, ACM, New York, NY, USA (2009)
10. Porta, D., Sonntag, D., Neßelrath, R.: A Multimodal Mobile B2B Dialogue Interface on the iPhone. In: Proc. 4th Workshop on Speech in Mobile and Pervasive Environments (SiMPE) (2009)
11. Riva, C., Laitkorpi, M.: Designing Web-Based Mobile Services with REST. In: Nitto, E., Ripeanu, M. (eds.) Service-Oriented Computing - ICSOC 2007 Workshops, pp. 439–450. Springer-Verlag, Berlin, Heidelberg (2009)
12. Rowland, D., Flintham, M., Oppermann, L., Marshall, J., Chamberlain, A., Koleva, B., Benford, S., Perez, C.: Ubikequitous Computing: Designing Interactive Experiences for Cyclists. In: Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services. pp. 21:1–21:11. MobileHCI '09, ACM, New York, NY, USA (2009)
13. Sonntag, D., Porta, D., Setz, J.: HTTP/REST-based Meta Web Services in Mobile Application Frameworks. In: Proceedings of the 4nd International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM-10). pp. 170–175. IARIA/XPS (Xpert Publishing Services) (2010)
14. Venetis, P., Halevy, A., Madhavan, J., Paşca, M., Shen, W., Wu, F., Miao, G., Wu, C.: Recovering Semantics of Tables on the Web. Proc. VLDB Endow. 4, 528–538 (2011)