A Field Application of LOD

LOD extraction from Web and LOD search by Sensor

Takahiro Kawamura
Research & Development Center, Toshiba Corp.

Akihiko Ohsuga
Graduate School of Information Systems,
University of Electro-Communications, Japan

ABSTRACT

Semantic search is intrinsically suited for information retrieval in field research, where keyword search is burdensome for users. Also, exploitation of mobile and facility sensors is now prevailing, but applications are still vague. Therefore, LOD can serve as an intermediary interpreting the semantics of the users and their environmental information obtained by the sensors and connecting it to the collective intelligence on the net. In this paper, to build ecosystem of LOD used in the field, we propose a mechanism of LOD content generation and its field application. Firstly, we introduce an Android application, which searches a plant fitting the environmental conditions obtained by the sensors from Plant LOD. Then, we show that our LOD generation mechanism builds the Plant LOD from the Web with an accuracy of 85 to 97%.

Keywords

Sensor, LOD, AR, Plant, Field

1. INTRODUCTION

Semantic search is intrinsically suited for information retrieval in field research, where a trial-and-error approach to search is difficult because input is less convenient and the network tends to be slower than in the case of desktop search. It is burdensome for users in the field to research something while changing keywords and looking through a list of the results repeatedly. Therefore, search with SPARQL, which can specify the necessary semantics, would be useful in the field. Moreover, exploitation of mobile and facility sensors is now prevailing, but applications are still vague although sensor information is overflowing. Thus, LOD can serve as an intermediary interpreting the semantics of the users and their environmental information obtained by the sensors and connecting it to the collective intelligence on the net. LOD and SPARQL have the tremendous potential in the field. However, to build ecosystem of LOD used in the field, it requires at least the actual LOD content for the field, and its appealing application which consumes that LOD. In the talk of Tim Berners-Lee at TED2009 and 2010, it is intended that someone publishes data, and then the other one will use it, and create application mash-ups. But it would be better to show a use case with the data, in which the linking data is semantically utilized. Therefore, we would like to propose both of a mechanism of LOD content generation and its concrete application in this paper, which correspond to triplification and application mash-ups in the Call.

The remainder of this paper is organized as follows. Section 2 outlines related work, and then firstly we introduce a field application of LOD, where LOD is searched based on the sensor information on a smartphone in section 3. Next, section 4 describes a mechanism of LOD content generation, where LOD is extracted from the Web. Finally, section 5 presents conclusions and identifies future issues.

I-SEMANTICS 2012, 8th Int. Conf. on Semantic Systems, Sept. 5-7, 2012, Graz, Austria Copyright 2012 ACM 978-1-4503-1112-0 ···$10.00.

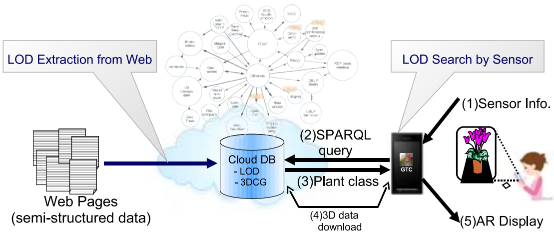

Figure 1: Proposed architecture of LOD use

2. RELATED WORK

First, we introduce two researches regarding architecture using sensors and semantics, and its application. The first one is Semantic Sensor Network (SSN), in which sensor data is annotated with semantic metadata mainly to support environmental monitoring and decision-making. SemSorGrid4Env[1] is applying it to flood emergency response planning. Our architecture (Fig. 1) is similar to SSN. However, instead of searching and reasoning within the corrected semantic sensor data, we assume the existence of LOD on the net, to which the sensor data is connected. In that sence, SENSEI[2] had almost the same purpose to integrate the physical with the digital world. But the project mainly addressed the scalability issue and the definition of services interfaces, and then LOD content was limited to a few types of data like geospatial.

The second one is about social sensor research, which integrates the existing social networking services and physical-presence awareness like RFID and twitter with GPS data to encourage users' collaboration and communication. Live Social Semantics (LSS)[3] applied it to some conferences and suggested new interests for the users. It resembles our architecture in that face-to-face contact events based on RFID are connected to the social information on the net. However, from the difference in its objective, which is a social or field support, the information flow is opposite. In our architecture, the sensor (client) side requests the LOD on the net, although in LSS the social information (DB) collects the sensor data.

Next, we introduce a research regarding LOD content generation. There are several ways and their combinations to generate LOD content. The first one is that an expert writes about a particular theme, e.g. data of Open Government. The second and third ones are user participatory creation, e.g. DBPedia and FreeBase, and crowdsourcing, e.g. use of Amazon Mechanical Turk. Both of them use the power of the masses (or collective intelligence), but are classified according to a business contract. The fourth one is conversion from the existing structured data like table, CSV and RDB, e.g. Life Science data, and then the last way we think is the (semi-)automatic generation of LOD from the Web. T. Mitchell presented studies on NELL (Never-Ending Language Learner) at ISWC09[4], which is a semantic machine learning system using the existing ontologies, where several learning methods are combined to reduce extraction errors. However, NELL is targeting the world, so the granularity of the properties is big and the number of them is limited. On the other hand, by restricting the domain of interest, it is possible for our mechanism to keep the variety of the properties and raise the accuracy of the extraction. The detailed description will be shown in section 4.

3. FIELD APPLICATION OF LOD

3.1 Problem Statement

Home gardens and green interiors have been receiving increased attention owing to the rise of environmental consciousness and growing interest in macrobiotics. However, the cultivation of greenery in a restricted urban space is not necessarily a simple matter. In particular, as the need to select greenery to fit the space is a challenge for those without gardening expertise, overgrowth or extinction may occur. In regard to both interior and exterior greenery, it is important to achieve an aesthetic balance between the greenery and the surroundings, but it is difficult for amateurs to imagine the future form of the mature greenery. Therefore, we considered it would be helpful if an 'agent' service offering gardening expertise were available on the user's mobile device. In this section, we describe our development of Green-Thumb Camera, which recommends a plant to fit the user's environmental conditions (sunlight, temperature, etc.) by using a smartphone's sensors. Moreover, by displaying its mature form as 3DCG using AR (augmented reality) techniques, the user can visually check if the plant matches the user's surroundings. Thus, a user without gardening expertise is able to select a plant to fit the space and achieve aesthetic balance with the surroundings.

Plant recommendation involves at least two problems. One problem concerns plant selection in accordance with several environmental conditions of the planting space. There are more than 300,000 plant species on the Earth. Also, their growth conditions involve a number of factors such

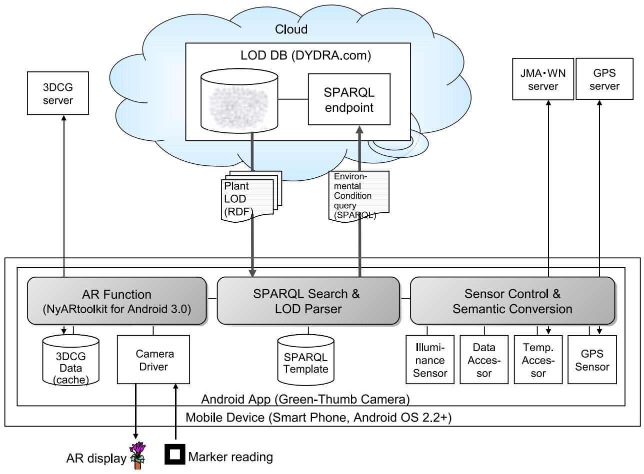

Figure 2: Green-Thumb Camera

as sunlight, temperature, humidity, soil (chemical nutrition, physical structure), wind and their chronological changes. Therefore, we have incorporated the essence of precision farming[5], in which those factors are carefully observed and analyzed, and crop yields are maximized through optimized cultivation. In our research, firstly, using the sensors on the smartphone, we determine the environmental factors listed in section 3.2.2, which we consider to be the major factors, and then try to select a plant based on those factors. Other factors, notably watering and fertilizing, are assumed to be sufficient [1].

Another problem concerns visualization of the future grown form. As well as the need to achieve aesthetic balance for both interior and exterior greenery, overgrowth is an issue. In fact, some kinds of plant cannot be easily exterminated. Therefore, we propose visualization of the grown form by AR to check it in advance.

3.2 Plant Recommendation Service

3.2.1 *Service flow of plant recommendation*

Firstly, the user puts an AR marker at the place where he/she wants to grow a plant. If the user looks at the marker through a camera view on the GTC App (Fig. 1), the app (1) obtains the environmental factors, (2) searches on LOD Cloud DB with SPARQL, and (3) receives some Plant classes that fit the environment. Then, the app (4) downloads 3DCG data for the plants, if necessary (the data once downloaded is stored in the local SD card), (5) overlays the 3DCG on the marker in the camera view. It also shows two tickers, one for the plant name and description below, and another for the retrieved sensor information on the top. If the user does not like the displayed plant, he/she can check the next possible plant by clicking 'prev' or 'next' button, or flicking the camera view. Furthermore, if the user clicks a center button, GTC shows a grown form of the plant. Fig. 2 shows the overview of this service.

Fig. 3 shows an experimental result of the plant recommendation. The test environment was as follows: Tokyo,

[1]A bioscience researcher whom we consulted confirmed that the factors listed in section 3.2.2 are sufficient to serve as the basis for plant recommendation to a considerable extent.

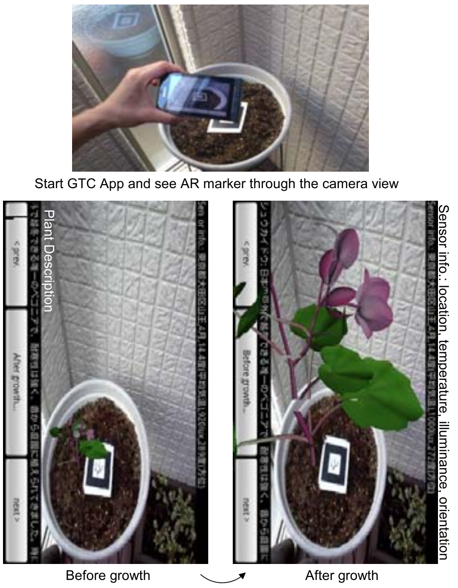

Figure 3: Result of plant recommendation

April, <3000 lux, approx. 15 °C. In this figure, 3DCG of a "Begonia" is displayed as a result. It is difficult to select a plant which can survive in a shade garden, so that we found that the service is helpful and working correctly. The GTC App is now open to the public, so anyone can download and try to use it [2]. In the near future, we are planning to conduct some evaluation by a group of potential users to determine it's effectiveness.

3.2.2 *Semantic conversion from sensor information to environmental factors*

This section describes the environmental factors, and how we convert raw data of the sensors to them.

Sunlight

This factor indicates the illuminance suitable for growing each plant and has several levels such as shade, light shade, sunny, full sun. To determine the current sunlight, we used a built-in illuminance sensor on the smartphone. After the application boots up, if the user brings the smartphone to the space where he/she envisages putting the plant and pushes the start button on the screen, the sunlight at the space is measured, and classified as the above levels.

Temperature

This factor indicates the range (min, max) of suitable temperature for a plant. To get the temperature, we referred to past monthly average temperatures for each prefecture from the Japan Meteorological Agency , using the current month and area (described below), instead of the current temperature.

Planting Season

The planting season means a suitable period (start, end) for starting to grow a plant (planting or sowing). To get the current month, we simply used the Calendar class provided by the Android OS. However, the season is affected by the geographical location (described below). Therefore, it is set one month later in the south area, and one month earlier in the north area. In the northernmost area, it is set two months earlier.

Planting Area

The planting area means a suitable area for growing a plant. To get the current area, we used the GPS function on the smartphone. Then, we classified the current location (latitude, longitude) for the 47 prefectures in Japan, and determined the provincial area.

3.2.3 *Recommendation Mechanism*

Firstly, we created a decision tree per plant because the reasons for recommendation are relatively easily analyzed from the tree structure. However, this approach obviously poses a difficulty in terms of scaling up since manual creation of training data is costly. Therefore, we prepared Plant LOD based on collective intelligence on the net and adopted an approach of selecting a plant by querying with SPARQL.

There are several DBs of plants targeting such fields as gene analysis and medical applications. However, their diverse usages make it practically impossible to unify the schemas. Furthermore, there are lots of gardening sites for hobbyists, and the practical experience they describe would also be useful. Therefore, instead of a Plant DB with a static schema, we adopted the approach of virtually organizing them using LOD on the cloud.

The SPARQL query includes the above-mentioned environmental factors obtained from the sensors in the FILTER evaluation, and is set to return the top three plants in the reverse order of the planting difficulty within the types of Plant class.

4. LOD CONTENT GENERATION

4.1 Overview of Plant LOD

The Plant LOD (Fig. 4) is RDF data, in which each plant is an instance of "Plant" class of DBpedia ontology. DBpedia has already defined 10,000+ plants as types of the Plant class and its subclasses such as "FloweringPlant", "Moss" and "Fern". In addition, we created 90 plants mainly for species native to Japan. Each plant of the Plant class has almost 300 Properties, but most of them are inherited from "Thing", "Species" and "Eukaryote". So we added 19 properties to represent necessary attributes for plant cultivation, some of which correspond to { name, country of origin, description, sunlight, temperature, planting season, blooming season, watering amount, planting difficulty }.

4.2 Semi-automatic generation of LOD

In order to collect the necessary plant information from the Web and correlate it to the DBPedia, we developed a semi-automatic mechanism to grow the existing LOD. But the plant names can be easily collected from a list on any gardening site and we have already defined the necessary properties based on our service requirements. Therefore, what we would like to collect in this case is the value of the property for each plant.

Process of our LOD generation is as follows. First of all, we make a keyword list, which includes an instance name

[2] http://www.ohsuga.is.uec.ac.jp/~kawamura/gtc.html (in Japanese)

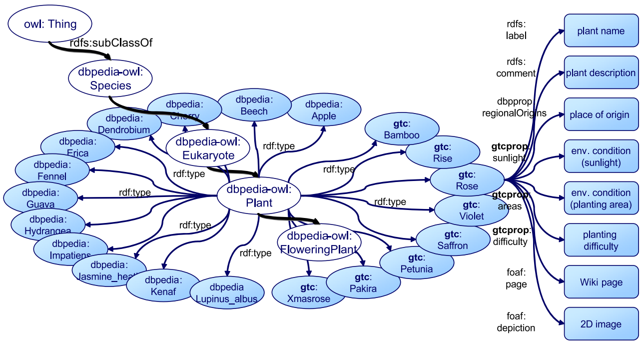

Figure 4: Overview of Plant LOD

(plant name) and a logical disjunction of property names and then search on Google.

As the boot strapping method, we first extract specific patterns from a web page based on some keys, which are the property names, and then we apply that patterns to other web pages to extract the values of the property. This method is mainly used for extraction of $< property, value >$ pairs from structured part of a page such as tables and lists.

However, we found there are many (amateur) gardening sites that explain the nature of the plant only in plain text. Therefore, we developed an extraction method using the dependency parsing, because a triple $< plantname, property, value >$ corresponds to $< s, v, o >$ respectively. It first follows modification relations in a sentence from a seed term, which is the plant name or the property name, and then extract the triple, or a triple $< -, property, value >$ in the case of no subject in the sentence ('-' is replaced with the plant name in the keyword list later).

We combine all the property values obtained above, and filter it if it matches to co-occurrence strings with the corresponding property names, where a set of the co-occurrence string are prepared in advance, e.g. the propery "temperature" obviously co-occurs with a string °C. Then, we form some clusters of the identical property values for each property name based on LCS (Longest Common Substring). Furthermore, for correction of errors, which may be an error of extraction and/or of the information source, we sum up the PageRanks of the source pages for each cluster to determine the best possible property value and the second-best. Finally, after a user determines a correct value from the proposed ones, CSV and RDF files are generated to each plant.

In either way, the key or seed of the boot strapping and the dependency parsing are retrieved from our predefined schema of Plant LOD in order to correlate to it.

4.3 Evaluation of LOD generation

4.3.1 Extraction accuracy

We applied this LOD generation mechanism to extract the values of the 13 properties for the 90 plants that we added. The result is shown in Table 1. If there are more than two clusters whose sum of the PageRanks are the same, we regarded them all as the first position. The accuracy is calculated in units of the cluster instead of each extracted value. In the case of 2-best, two clusters are compared with the correct value, and if either one of the two is correct, then it is regarded as correct. A cluster consists of the extracted

Table 1: Extraction accuracy

Accuracy (%)	1-best	2-best	1-best (bootstrap only)	1-best (dependency only)
Precision	85.2	97.4	88.6	85.2
Recall	76.9	87.2	46.2	76.9
Amount Ratio (%)	–	–	10.8	89.2

values for a property, which seem identical accroding to LCS, but the number of the values in a cluster may vary from more than 10 to 1.

The 2-best achieved an average precision of 97% and an average recall of 87%. So if we permit a user to make a binary choice, it is possible to present a correct answer in many cases. The accuracy of the automatic generation would not be 100% after all and a human checking is necessary. Therefore, the binary choice is a realistic option.

In detail, the boot strapping collects smaller amounts of values (11%), so the recall is substantially lower (46%) than the dependency parsing, but the precision is higher (89%). It is because data written in the tables can be correctly extracted, but lacks diversity of properties. The total accuracy is affected by the dependency parsing, because the biggest cluster of the PageRank is composed mainly of the values extracted by the dependency parsing. So we will need to weight the extracted values of the boot strapping to a certain degree.

5. CONCLUSION AND FUTURE WORK

Under the vision that LOD is suited for information retrieval in the field, we propose a mechanism of LOD content generation for a domain and its application to indicate the possible benefit of field use.

In the near future, we would like to apply this architecture of environmental sensing → semantic conversion → LOD Cloud (← collective intelligence) in other fields that would benefit from greater IT support. Now we are considering the provision of support for greening business, which addresses environmental concerns, and for agri-business in regard to the growing food problem.

6. REFERENCES

[1] A. Gray, et al.: "A Semantically Enabled Service Architecture for Mashups over Streaming and Stored Data", Proc. of 8th Extended Semantic Web Conference, 2011.

[2] M. Presser, P.M. Barnaghi, M. Eurich, C. Villalonga: "The SENSEI project", IEEE Communications Magazine, Vol. 47 Issue 4, 2009.

[3] M. Szomszor, C. Cattuto, W. V. Broeck, A. Barrat, H. Alani: "Semantics, sensors, and the social web", Proc. of 7th Extended Semantic Web Conference, 2010.

[4] T.M. Mitchell, J.Betteridge, A. Carlson, E.R. Hruschka Jr. and R.C. Wang: Populating the Semantic Web by Macro-Reading Internet Text (Invited Paper), Proc. of 8th International Semantic Web Conference, 2009.

[5] A. Srinivasan: Handbook of precision agriculture: principles and applications, Routledge, 2006.

[6] S. Brin: "Extracting patterns and relations from the world wide web", Proc. of WebDB Workshop at 6th International Conference on Extended Database Technology, 1998.