Query Answering using User Feedback and Context Gathering for Web of Data

Takahiro Kawamura
Graduate School of Information Systems
University of Electro-Communications
Tokyo, Japan
e-mail: kawamura@ohsuga.is.uec.ac.jp

Akihiko Ohsuga
Graduate School of Information Systems
University of Electro-Communications
Tokyo, Japan
e-mail: akihiko@ohsuga.is.uec.ac.jp

***Abstract*—The 'Web of Data' is a growing trend for the creation of innovative services on the web. Thus, a search engine for the data is becoming important for promoting data-intensive services. However, full-text search is not suitable for data fragments, and formal query languages are difficult for ordinary users. Therefore, we propose a query answering system in natural language over the 'Web of Data'. We focus on mapping of question sentences to open-schema data, and data acquisition, and then propose improvement of accuracy based on user feedback and acquisition of new data by user context information. We also present 'Flower Voice', which is an environmental application of the query answering system for assisting with users' fieldwork and confirm the effectiveness.**

Keywords*—*Web of Data; Open Data; Query Answering System; Field; Plant.

I. Introduction

The 'Web of Data' is attracting attention for the creation of innovative service businesses, mainly in the areas of government, bioscience, and smart X projects [3], [7], [9], [15]. To promote the application of the data in a greater number of consumer services, it would be helpful to have a search function for the data that can reveal what kinds of data are available on the web. Especially if the data on the web forms Linked Open Data (LOD) that is the collection of interrelated datasets described in a triple language like Resource Description Framework (RDF) in Extensible Markup Language (XML) format, full-text search is not suitable for data fragments in the linked datasets. Moreover, it is difficult for ordinary users to perform searches using SPARQL Protocol and RDF Query Language (SPARQL). We therefore propose a query answering system for matching triples extracted from the user query sentence to triples $< Subject, Verb, Object >$ in the RDF. This also serves as a registration mechanism for user-generated triples.

This paper is organized as follows. Section 2 describes problems with and approaches to realizing the query answering service to the LOD. Section 3 proposes an application for this software, Flower Voice, which is a smartphone tool for searching for information and for logging agricultural work. We then present several examples of related work in Section 4, and finally conclude with intended future work in Section 5.

II. Related Work

In research on the query answering (QA) systems and databases, many attempts have been made to automatically translate from natural language queries to formal languages like Structured Query Language (SQL) and SPARQL in order to help the understanding of ordinary users and even basebase (DB) experts. Research also exists on outputting the queries and results into natural language sentences [22], [11]. Although it is difficult to apply full-text search to data fragments, as we pointed out in Section 1, there has been research on converting a keyword list to a logical query [14], [23], [25].

In this section, we focus on Linked (Open) Data as a data structure and SPARQL as a query language, and classify research on QA systems, which translate natural sentences into queries into two categories based on whether a deep or shallow linguistic analysis is needed.

One system that requires deep linguistic analysis is ORAKEL [4], [5]. It first translates a natural sentence into a syntax tree using Lexicalized Tree Adjoining Grammars, and then converts it to F-logic or SPARQL. Although it is able to translate while retaining a high degree of expressiveness, it also requires the original sentence to be precise and regular. [27] considers a QA system together with the design of a target ontology mainly for event information, and features handling of temporality and N-ary during the syntax tree creation. It assigns the words of the sentence to slots in a constraint called a *semantic description* defined by the ontology, and finally converts the semantic description to SPARQL recursively. However, it requires advance knowledge of the ontology structure.

In terms of the voice-controlled QA system for users, however, these approaches are problematic for practical use due to voice recognition errors, syntax errors in the original sentences, and triplification errors. Furthermore, if the DB is open, the assumption that the ontology schema is already given is also questionable. Approaches that use shallow linguistic analysis thus aim for portability and schema independence from the DB. Our proposed system falls into this category.

FREyA [6] was originally developed as a natural language interface for ontology search. It has many similarities with our system like matching the words from the sentence with Resources and Properties by using a string similarity measure and synonyms from WordNet and improvement of accuracy based on user feedback. However, it performs conversion of the sentence to a logical form using ontology-based constraints (without consideration of the syntax of the original sentence unlike the semantic description), assuming completeness of the ontology used in the DB. By contrast, DEQA [17] takes an approach called Template-Based SPARQL Query Generator [26]. It takes prepared templates of SPARQL queries and converts the sentence to fill the slots in the template (not the ontology constraint). Like our system, DEQA also applies to a specific domain (real estate search), and exhibits a certain degree of accuracy. PowerAqua [18], [19], [20] also originated as a natural language interface for ontology search and has similarities with our system such as a simple conversion to

 ISBN: 978-1-61208-310-0

basic graph pattern called Query-Triples, matching of words from the sentence with Resources and Properties using a string similarity measure and synonyms from WordNet, and the use of user feedback. When used with the open data, PowerAqua also introduces heuristics according to the query context to prevent decreased throughput.

[21] serves as a useful reference for surveying QA systems. The system proposed in this paper is related to a number of works. However, it is distinguished by using a social approach, that is, improvement of accuracy and data acquisition through user participation by seamlessly combining the search query and registration statement. There are no similar work in terms of application to fieldwork (and Japanese sentences). Also, our system currently does not use the ontology, since our target source for the query is the LOD and we assume the open schema scenarios. The LOD schema is not regulated by any organization, and there may be several properties of the same meaning and a sudden addition of a new property. In addition, we assume searching over multiple LOD sets made by the different authors. Therefore, we do not rely on the ontology behind the LOD. However, the proper adaptation of the ontology is useful to interpret the semantics, and thus we will address this issue in the future.

Recently, well-known voice assistants such as Apple Siri and xBrainSoft Angie have been commercialized. Both offer high accuracy voice recognition functions and are good at certain typical tasks such as calling up handset capabilities and installed applications, which are easily identified from the query. In terms of the information search, these voice assistants correctly answer the question in cases that the information source is a well-structured web site such as Wikipedia. However, extracting the information from unstructured web sites often fails and they return the search engine results page (SERP), and thus the user needs to tap URLs from the list. Angie also provides a link to Facebook and a development kit. By comparison, our system focuses on the information search using the LOD as the knowledge source, and raises the accuracy using the user feedback.

Targeting smartphone applications in agriculture, Fujitsu Ltd. provides a recording system, in which the user can simply register the work type by buttons on the screen that have photos of cultivated plants. NEC Corp. also offers a machine-to-machine (M2M) service aimed at visualization of sensor information and support of farming diaries. Both systems address recording and visualization of work the same as Flower Voice, but our system contains a form of voice-controlled QA system for the open-schema data that takes a social approach by combining data recording with data viewing.

In terms of combination of the sensor and the semantics, the sensor data in semantic sensor network are annotated with semantic metadata to support for environmental monitoring and decision-making at the time of disaster. For example, SemSorGrid4Env [13] has been applied to flood emergency response planning. However, searching and reasoning is conducted on the collected semantic sensor data, whereas the sensor data in our system is connected to a broader range of the LOD DB. In social sensor research, social networking services and physical-presence awareness like Radio Frequency IDentification (RFID) and twitter with GPS data are integrated in order to encourage the collaboration and communication among users. For example, Live Social Semantics [24] was applied to academic conferences, and suggested new interests for attendees. Although the objectives are different, the architecture where face-to-face contact events obtained by RFID are connected to the social information is similar to our system. This will provide guidance for scaling up our system in the future.

III. Problems and Approaches to Open-Schema Data

In the classification of interactive systems, our QA service is in the same category as Siri, which is a DB-search QA system. However, Siri is more precisely a combination of a closed DB and open Web-search QA system, whereas our system is an ‘open’ DB-search QA system. Although the detailed architecture is described in the next section, the basic operation is to extract a triple such as subject, verb, and object from the query sentence by using morphological analysis and dependency parsing. Figure 1 shows a conversion from a dependency tree to a triple. Any query words (what, where, when, why, etc.) are then replaced with a variable and the LOD DB is searched. SPARQL is based on graph pattern matching, and this method corresponds to a basic graph pattern (one triple matching). At the data registration, if there is a Resource corresponding to the $Subject$ and a Property corresponding to the $Verb$ from the user statement, a triple, which has the $Object$ from the user statement as the Value is added to the DB.

Although DB-search QA systems without dialog control have a long history, there are at least the following two problems because the data schema is ‘open’.

A. Mapping of Query Sentence to LOD Schema

Although a mapping between the verb in the query sentence (in Japanese) and a Property in the LOD schema of the DB must be prepared in advance, both of them are unknown in the open-schema data (compared with a closed DB where the schema is given), so the score according to the mapping degree can not be predefined.

We therefore use a string similarity and a semantic similarity technique from the field of ontology alignment to map verbs to Properties, and attempt to improve the mapping based on user feedback. We first register a certain set of mappings {Verb (in Japanese), Property} as seeds in the Key-Value Store (KVS). If a verb is unregistered we then do the following:

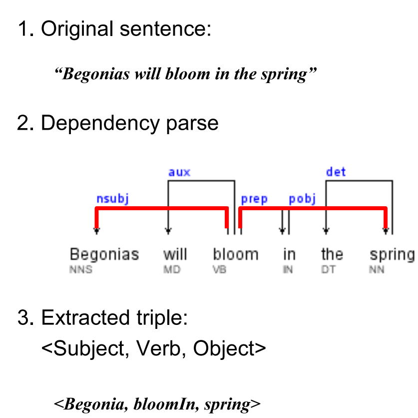

Figure 1: Conversion from dependency tree to triple.

(1) Expand the verb to its synonyms using Japanese WordNet ontology, and then calculate the Longest Common Substring (LCS) with the registered verbs to use as the similarity.
(2) Translate the new verb to English, and calculate the LCS of the English with the registered Properties.
(3) If we find a Resource that corresponds to a subject in the query sentence in the LOD, we then calculate the LCSs of the translated verbs with all the Properties belonging to the Resource, and create a ranking of possible mappings according to the combination of the above LCS values (Figure 2).
(4) The user feedback, which indicates which Property was actually viewed, is sent to the server, and the corresponding mapping of the new verb to the Property is registered in the KVS.
(5) Since the registered mappings are not necessarily correct, we recalculate the confidence value of the mapping based on the number of pieces of feedback, and update the ranking of the mapping to improve the N-best accuracy (refer to Section IV-D).

B. Acquisition and Expansion of LOD Data

Even for an open DB, it is not easy for an ordinary user to register new triples in the DB. We therefore provide an easy registration method that uses the same extraction mechanism as triples from statements.

We also provide an automatic registration method of the user context information to support of the data registration by the user. When the user registers a triple in the DB, the sensor data are automatically aggregated by using built-in sensors on the smartphone, and the context information related to the triple are inserted in the DB after the semantic conversion of the sensor data. Although Twitter is providing a function for attaching geographical information to tweets, this method is available with a greater variety of the context information. By using this method, the user can register not only the direct assertion, that is an object in the user statement, but also several background information at once. We describe examples of the sensor data and the corresponding context information in the next section. This is an approach to collect the necessary data from side effects of the user actions (the registration in this case), and corresponds to a typical method in the Human Computation mechanisms.

By contrast, we also attach the Twitter ID of the registrant

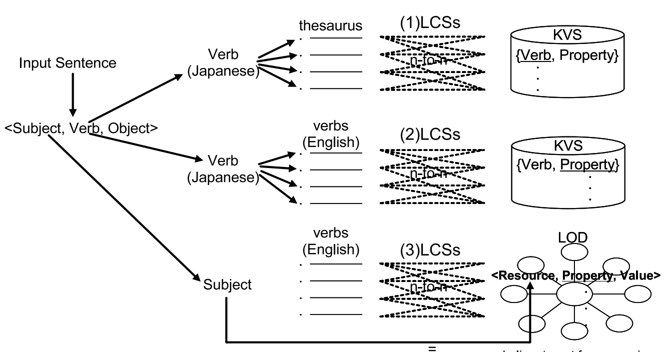

Figure 2: Calculation of LCS.

to the data as the Creator in order to raise the feeling of contribution in the user who share the significance of building the 'Web of Data'. This is another method in the Human Computation mechanisms. These efforts further promote the social user participatory approach.

We have also been developing a semi-automatic LOD extraction mechanism from web pages for generic and specialized information; this mechanism uses Conditional Random Fields (CRF) to extract triples from blogs and tweets. As [29] shows, it has achieved a certain degree of extraction accuracy.

IV. Development of Applications for Fieldwork Support

This section shows the implementation of our service and an applications. The applications of QA systems include Interactive Voice Response, guide systems for tourists and facilities, car navigation systems, and game characters. However, these all basically use closed DBs and would not be the best match for an open DB. In addition, our system does not currently incorporate dialog control such as Finite-State Transducers (FST), so that problem-solving tasks such as product support are also difficult. We thus focus on searching for information as described in the previous sections and introduce the following applications.

1) General Information Retrieval
DBpedia [9] already stores more than one billion triples, and there are 31 billion triples on the web, so part of the information people are browsing in Wikipedia can be retrieved from LOD.
2) Recording and Searching Information for Fieldwork
Since the system allows user registration of information, the information relevant to a specific domain can be recorded and searched, including for agricultural and gardening work, elevator maintenance, factory inspection, camping and climbing, evacuation, and travel.
3) Information Storage and Mining Coupled with Twitter
If we focus on the information sharing, it is possible that when a user tweets using a certain hash tag (#), the tweet is automatically converted to a triple and registered in the LOD DB. Similarly, when the user submits a query using a hash tag, the answer is mined from the LOD DB, which stores a large amount of past tweets. This would be useful for the recording and sharing of word of mouth information and life log information.

Although the above (1) is our purpose mentioned in the introduction, we introduce an application of our QA system from the second perspective in the following section to evaluate the system in a limited domain, which is "Flower Voice" to answer a query regarding the agricultural, gardening work like disease and pest, fertilization, maintenance, etc.

A. Flower Voice

Urban greening and agriculture have been receiving growing attention due 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. Beginners who have no gardening expertise have questions and get into trouble in several situations ranging from planting to harvesting. Although the user could employ

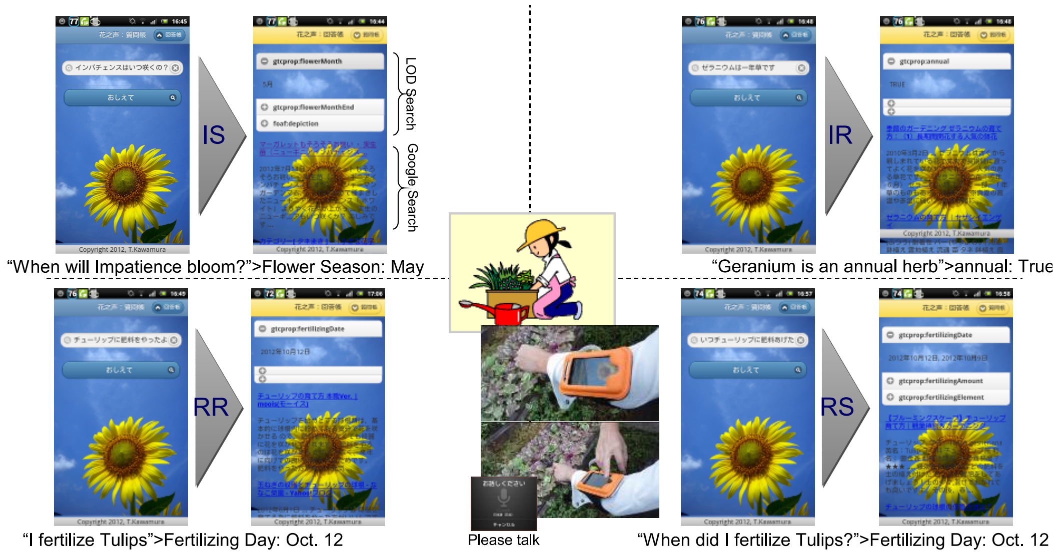

Figure 3: Overview of Flower Voice.

a professional gardening advisor to solve these problems, this would involve costs and may not be readily available in urban areas. Also this kind of work cannot be fully planned and the gardener needs to respond to the current status of the plants on site, since it highly depends on the surrounding environment. However, searching the Internet using a smartphone suffers from the disadvantages of inputting keywords and iteratively tapping and scrolling through SERP to find the answer. Therefore, we developed Flower Voice, which is a QA service for smartphones that answers questions regarding agricultural and gardening work. Moreover, we used a voice control, which is suitable for this work since users typically have dirty hands (and no need to be shy because no eyes and ears around). Furthermore, we provided a mechanism for registering the work of the user, since data logging is the basis of precision farming according to the Japanese Ministry of Agriculture. This is a tool for searching information and for logging by voice using smartphones for agricultural and gardening work. Figure 3 shows an overview of Flower Voice. It automatically classifies the speech intention (Question Type) of the user into the following four types (Answer Type is a literal, Uniform Resource Identifier (URI), or image).

1) Information Search (IS)
   Search for plant information in the LOD DB.
2) Information Registration (IR)
   Register new information for a plant that does not currently exist in the LOD DB or add information to an existing plant.
3) Record Registration (RR)
   Register and share records of daily work. Since data logging is important for the farming, it would be useful to add sensor information together with the registered record. However, the verbs that can be registered are limited to the predefined Properties in the DB (see the next section).
4) Record Search (RS)
   Search through records to remember previous work and view the work of other people.

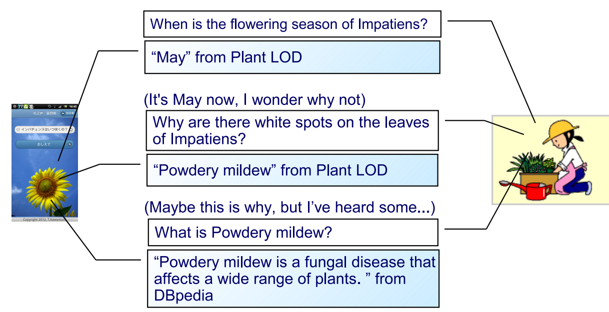

Figure 4: Chaining search.

The possible use cases are as follows.

1) Chaining Search: This is the case, in which Flower Voice continuously provides the pinpoint data that the user wants to know on site during the work (Figure 4).

2) Use of User Participation: This is the case where the user uses the registration mechanism to share a piece of data they learned about. This would be useful, for example, in environmental surveys (Figure 5 above) where users cooperatively investigate and report the specific environmental items such as rare species of plants that the users discovered, and building a knowledge community (Figure 5 below). The registered data are annotated with the Twitter ID of the registrant.

B. Plant LOD

The LOD used by Flower Voice is called Plant LOD, and consists of more than 10,000 Resources (species) under the Plant Class in DBpedia and 104 Japanese Resources that we have added. We have also added 37 Properties related to plant cultivation to the existing 300 Properties. In terms of the LOD Schemas for registering records, we prepared Properties mainly for recording dates of flowering, fertilizing, and harvesting. Figure 6 illustrates Plant LOD, which is an extension of the LOD used by Green-Thumb Camera [16], which was developed for introducing plants (greening design).

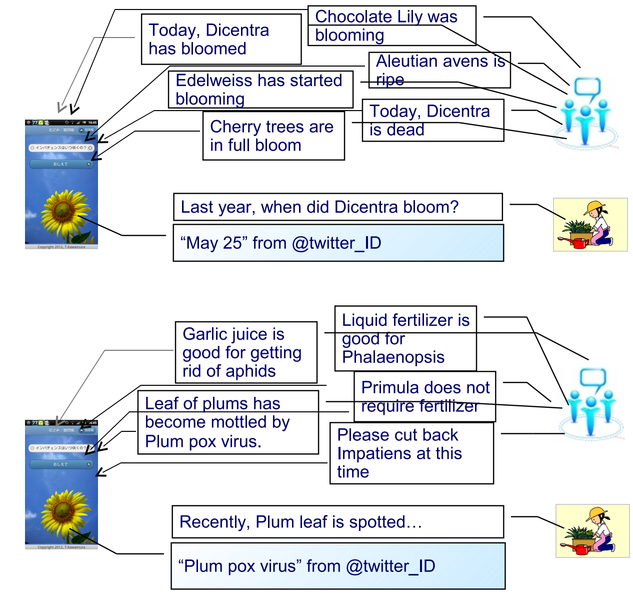

Figure 5: Use for environmental survey (above) and knowledge community (below).

Plant LOD is now stored and publicly available at Dydra.com.

C. System Architecture

Figure 7 shows the architecture of Flower Voice. The user can input a query sentence by Google voice recognition or keyboard. The system then accesses the Yahoo! API for Japanese morphological analysis to extract a triple using the built-in dependency parser, and generates a SPARQL query by filling in slots in a query template. The similar process also works for English sentences, although the morphological analyzer and dependency parser must be changed to, for example, Berkeley Parser [2]. The search results are received in XML format. After searching the {Verb, Property} mappings registered in Google Big Table and accessing the Microsoft Translator API and Japanese WordNet Ontology provided by the National Institute of Information and Communications Technology (NICT), the LCS values for each mapping are calculated as described in Section 2. The order of matching is firstly matching the $Subject$ against Resources by tracing 'sameAs' and 'wikiPageRedirects' links, and then searching for $Verb$ matches with the Properties of the Resources. A list of possible answers is then created from the pairs of Properties and Values with the highest LCS values. The number of answers in the list is set to three due to constraints on the client UI. The results of a Google search are also shown below in the client to clarify the advantages and limitations of the QA service by comparison. User feedback is obtained by opening and closing a collapsible area in the client, which gives a detailed look at the Value of the Property (but only the first click). During searches, feedback updates the confidence value of a registered mapping {Verb, Property} or registers a new mapping. During registration, the feedback has the role of indicating which of three Properties to which the $Object(Value)$ should be registered. The client UI displays the results. Text-to-speech has not been implemented yet.

In terms of the computational performance, this service is currently running on 1 CPU with 55.1 MBytes memory of Google App Engine 1.8.4, where 1 CPU corresponds to 1.0–1.2 GHz 2007 Opteron. Then, it needs almost 1 (sec) for retrieving a plant data from the LOD, but once loaded the data, it takes 0.05–0.3 (sec) for answering a query. However, it is difficult to compare the performance with other services, and thus evaluate the data accuracy and acquisition in the following sections.

TABLE I: MAPPING OF SENSOR AND CONTEXT INFORMATION.

Sensors	Context Info. that can be obtained
Clock	Date, Time
GPS	Location, Nearby POI
(Combination of the above two)	Weather, Temperature, Humidity
Illuminance	Space{Indoor, Outdoor}
Acceleration	Status{Moving, Stop}, Walking Time&Distance

The automatic registration method of the user context information is realized by the acquisition of sensor data and the semantic conversion based on the LOD Schema. The sensor data are obtained by JavaScript running on the smartphone, except for Osaifu-Keitai that is FeliCa (a specification of Near Field Communication) mobile payment. Table I shows examples of the sensor data and the corresponding context information. Note that although the clock and Osaifu-Keitai are not the sensors, these are included in the table for showing the mapping with the context information. Furthermore, Points of Interest (POI) and Weather are obtained by accessing Yahoo! Open Local Platform and Japan Meteorological Agency based on the Global Positioning System (GPS) information. The POIs specifies location names (buildings, companies, stations) around the location.

We prepared the LOD schemas (Properties) corresponding to the above context information, and once the sensor information is retrieved, we convert it to the property value with the designated data types like literal and interger that are predefined by the schemas. For example, when a user registers a triple describing "a flower has blossomed", the sensor data for the location is converted to literal, one for the temperature is conveted to integer, and one for the space is translated to Indoor or Outdoor, respectively. Then, the context information such as **gtcprop:flowerAddress** (location), **gtcprop:flowerDateHighTemp** (highest temperature of the day), **gtcprop:flowerDateLowTemp** (lowest temperature of the day), **gtcprop:flowerSpace** (space of the flower) are automatically registered in the LOD DB.

We show the combinations of Properties registered by the user and the additional context information obtained by the sensors in Figure 8. In this figure, **gtcprop:flowerDate–Weather** means that the weather is registered with a flowering date. The links of the property and the context information can be easily changed according to the purpose of application. Flower Voice currently does not use the context information related to the user actions like number of steps, walking distance, walking time, etc. Therefore, there are unlinked contexts in the figure.

We have also added an advanced function for changing the LOD DB that is searched by the user input to a SPARQL endpoint as entered in an input field of the client UI, although

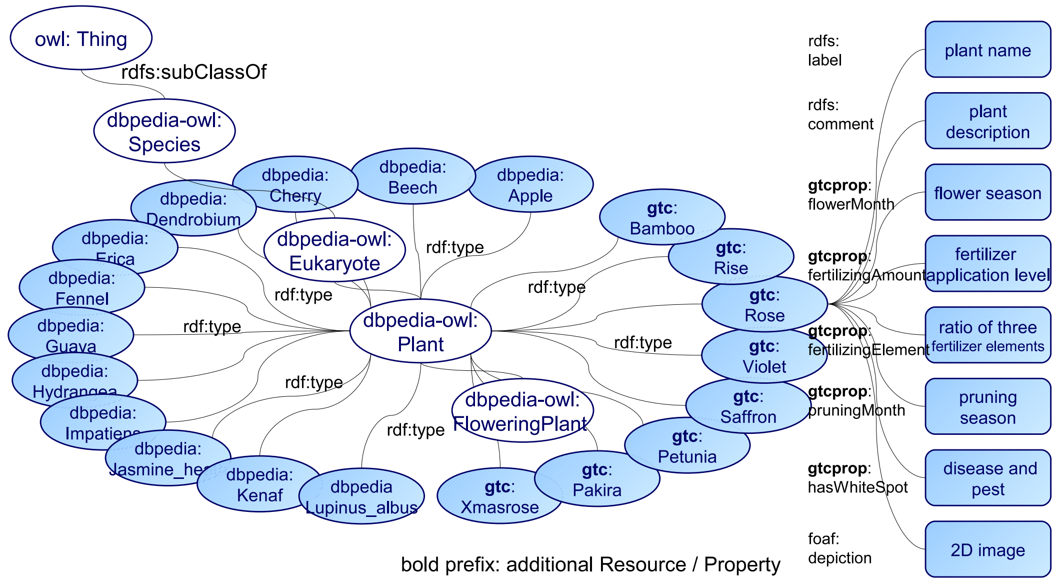

Figure 6: Plant LOD.

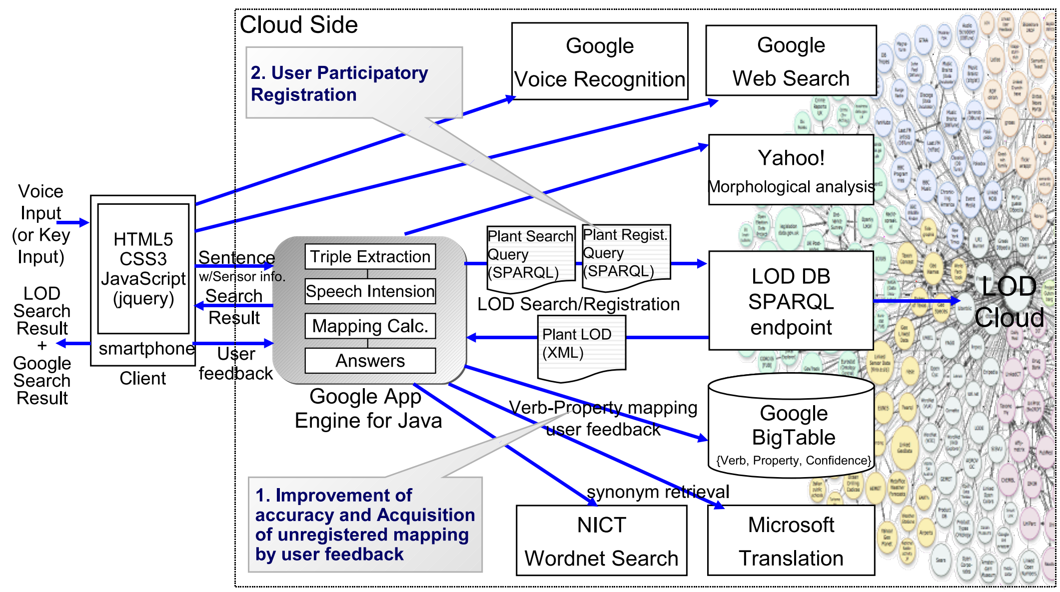

Figure 7: Mashup architecture.

the change is limited to searches. This is not compatible with all servers because the query is based on predefined templates and the results are received in XML format. Some servers also require attention to latency. Endpoints that have been confirmed include DBpedia Japanese [10], Data City SABAE [8], Yokohama Art LOD [28], etc. Users can also manually register {Verb, Property} mappings. If the Property that a user wants does not appear in the three answers, the user can input a {Verb, Property} mapping in an input field. The mapping is then registered in the KVS and will be searched by the next query. Although this function targets users who have some expertize dealing with LOD, we are expecting to discover unanticipated use cases when the system is open to users.

Flower Voice is available from our website (in Japanese) [12], and almost 500 users have used it with at least one query so far (Flower Voice won a Judges' Special Award in the LOD Challenge Japan 2012).

D. Evaluation of Accuracy Improvement

We conducted experiments on the current system to confirm the search accuracy, and how the accuracy is improved by the user feedback mechanism described in Section 2. Note that if a sentence is composed of more than two triples, it must be queried as separate single sentences. The intention of the speech, such as searching or registration, is classified by the existence of question words and the use of postpositional words, not by intonation. Sentences need to be literally described regardless of whether they are affirmative or interrogative.

 ISBN: 978-1-61208-310-0

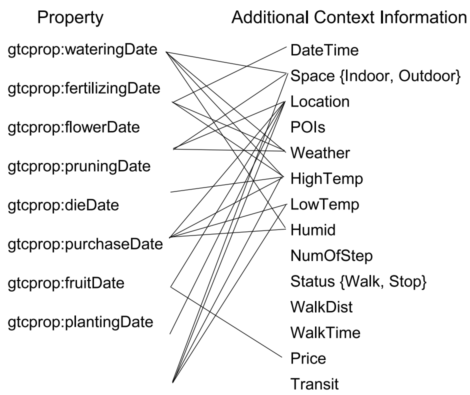

Figure 8: Properties and corresponding contexts.

TABLE II: ACCURACY OF SEARCH.

	False			True	
	no Res.	no Prop.	triplifica-tion error	1-best	3-best
1st Set (ave)	18.2%	0%	9.1%	54.5%	72.7%
2nd Set (ave)				72.7%	72.7%

In the experiment, we asked several experienced gardeners to select frequently asked questions from their daily work, and collected 99 query sentences (and the preferred answers). Although there were no duplicate sentences, sentences having the same meaning at the semantic level were included. We then randomly constructed 9 test sets consisting of 11 sentences each. We first evaluate one test set randomly selected, and give the correct feedback, which means registering {Verb, Property} mappings and updating the confidence value for one of the three answers for each query. We then proceed on to the next set. After evaluating the second test set, we clear the effects of the user feedback and repeated the above again from the first set. The difference of the accuracy between the first and the second set corresponds to the improvement by the user feedback. The results are shown in Table II. We assume that query sentences are correctly entered, since in practice Google Voice Recognition returns the possible results of the recognition, and the users can select the correct sentence in a dialog, or start over from speech.

In the table, “no Res.” means that there was no corresponding Resource (plant) in the Plant DB, and “no Prop.” means no Property corresponding to the Verb in the query sentence. “triplification error” indicates failure to extract a triple from the query sentence in case of a long complex question, etc. N-best accuracy is calculated by the following equation:

$$N - best\ precision = \frac{1}{|D_q|} \sum_{1 \leq k \leq N} r_k \qquad (1)$$

,where $|Dq|$ is the number of correct answers for query q, and r_k is an indicator function equaling 1 if the item at rank k is correct, zero otherwise. In the case of 3-best, the three answers are compared with the correct answer, and if any one of them is correct, then the result is regarded as correct.

TABLE III: EFFECTIVENESS OF ADDITIONAL CONTEXT.

False		True		
no Prop.	triplifica-tion error	success	num. of additional context	num. of useful context
0%	9.1%	90.9%	9.3 triples per registration	3.4 triples per registration

We found that approximately 20% of the queries were for unregistered plants, and the prepared Properties covered all of the queries. The current extraction mechanism is rule-based, and approximately 10% of the queries were not analyzed correctly. Although the queries are in a controlled natural language since the queries need to be literally described as single sentences, we found that 90% of questions are allowed in our system. We are planning to extend the rules and use CRF [29] for further improvement.

The N-best accuracy can be increased by providing more data such as Resources and Properties in the Plant LOD and {Verb, Property} mappings, and so the base accuracy of the first set is not particularly important. However, by comparing the results for the first set with the second one, we can confirm that the improvement of the accuracy was affected by the user feedback (note that the fact that 1-best accuracy equals 3-best accuracy means all the correct answers are in the first position, that is, they are among the first three positions).

We expect that the number of acquired {Verb, Property} mappings will form a saturation curve according to the number of trials that saturates to a domain-dependent value. In this domain, we found that an average of 0.09 new mappings were acquired per trial (query) from an initial 201 mappings in the DB. More detailed analysis will contribute to the bootstrap issue for applications in other domains.

E. *Evaluation on Data Acquisition*

We also conducted experiments on the current system to confirm the effectiveness of the context acquisition. In the experiment, we first collected 44 sentences for the registration from the experienced gardeners, and then registered them in the DB. We do not consider voice recognition errors as well as the previous experiment. We also assume that the user feedbacks that indicate Properties for registering the context information are correctly entered. The results are shown in Table III.

In the table, “no Prop.” means no Property corresponding to the Verb in the sentence. “triplification error” indicates failure to extract a triple from the query sentence. However, if there was no corresponding Resource (plant) in the Plant DB during the registration, the Resource is automatically created, and so “no Res.” does not happen in this experiment. Furthermore, “num. of additional context” means how many triples for the context information on average are automatically added with a triple that is successfully registered. Note that all the context information shown in Figure 8 are not necessarily obtained in practice because of the status and timing of the sensors. “num. of useful context” means the number of triples the experienced gardeners considered useful among all the additional context information. The followings are examples of the useful context information.

wateringDate–Location, HighTemp, Space: By this combination, useful data to analyze correlation with the watering period to the circumstances and seasons would be collected.

flowerDate, fruitDate, dieDate–Address, Weather, HighTemp, LowTemp, Space: By these combinations, usefull data regarding the process from flowering and fruiting to dying depending on weather change in each area would be collected.

pruningDate, flowerDate, fruitDate–Address, HighTemp, LowTemp: Correlation with flowering and fruiting to pruning can be investigated based on these data.

hasWhiteSpot–Humid: The risk of developing red spiders would be anticipated by drying in the planting space.

As a result, by automatically adding the context information as the side effects of the user registration, we confirmed that the useful triples have been increased 3.4 times more as the result. If these data are described in RDF and shared in the Cloud DB, then people who have several viewpoints can easily analyze from their own aspects.

V. Conclusion and Future Work

In this paper, we proposed a query answering service, which uses the LOD as a knowledge source to facilitate the spread of the data-intensive services. We then developed and evaluated an application for assisting with fieldwork. It also features the Human Computation mechanisms, namely, the improvement of accuracy based on user feedback and the acquisition of new data by user participation.

We have realized the method to register the context information converted from the sensor data in order to increase the new data in this paper. However, as lessons learnt from the application, we should also consider to use the context information for the searches. This means the refinement of the search results using subgraph matching based on user context. It would be useful to automatically select the necessary information based on the current and past situation of the user without explaining every detail. Also, we need to conduct the evaluation of accuracy with the LOD size, in general scalability of the system, and discuss the impact of the values gathered in the sense of how well does the system scale. In the future, we also intend to collect customer feedback on this application, and to apply the system to domains other than agriculture.

References

[1] Ars Technica, “”Siri, does anyone still use you?” Yes, says survey,” http://arstechnica.com/apple/2012/03/siri-does-anyone-still-use-you-yes-says-survey/[accessed: 2013-09-06].

[2] Berkeley Parser, https://code.google.com/p/berkeleyparser/[accessed: 2013-09-06].

[3] Bloomberg Businessweek Technology, “Q&A with Tim Berners-Lee,” http://www.businessweek.com/stories/2007-04-09/q-and-a-with-tim-berners-leebusinessweek-business-news-stock-\market-and-financial-advice[accessed: 2013-09-06].

[4] P. Cimiano, “ORAKEL: A Natural Language Interface to an F-Logic Knowledge Base,” Proc. of 9th International Conference on Applications of Natural Language to Information Systems (NLDB), Jun. 2004, pp. 401-406.

[5] P. Cimiano, P. Haase, J. Heizmann, and M. Mantel, “Orakel: A portable natural language interface to knowledge bases,” Technical Report, University of Karlsruhe, March 2007, pp. 1-77.

[6] D. Damljanovic, M. Agatonovic, and H. Cunningham, “FREyA: an Interactive Way of Querying Linked Data using Natural Language,” Proc. of 1st Workshop on Question Answering over Linked Data (QALD-1), May 2011, pp. 125-138.

[7] DATA.gov, http://www.data.gov/[accessed: 2013-09-06].

[8] Data City SABAE, http://lod.ac/sabae/sparql[accessed: 2013-09-06].

[9] DBpedia, http://dbpedia.org[accessed: 2013-09-06].

[10] DBpedia Japanese, http://ja.dbpedia.org/sparql[accessed: 2013-09-06].

[11] B. Ell, D. Vrandecic, and E. Simperl, “SPARTIQULATION: Verbalizing SPARQL Queries,” Proc. of Interacting with Linked Data (ILD), May 2012, pp. 50-60.

[12] Flower Voice (in Japanese), http://www.ohsuga.is.uec.ac.jp/\~{}kawamura/fv.html[accessed: 2013-09-06].

[13] R. Garcia-Castro et al., “A Semantically Enabled Service Architecture for Mashups over Streaming and Stored Data,” Proc. of 8th Extended Semantic Web Conference (ESWC), May 2011, pp. 300-314.

[14] P. Haase, D. Herzig, M. Musen, and D. T. Tran, “Semantic Wiki Search,” Proc. of 6th European Semantic Web Conference (ESWC), May 2009, pp. 445-460.

[15] T. Heath, “The Web of Data,” Proc. of 9th Summer School on Ontology Engineering and the Semantic Web (SSSW), July 2012.

[16] T. Kawamura and A. Ohsuga, “Toward an ecosystem of LOD in the field: LOD content generation and its consuming service,” Proc. of 11th International Semantic Web Conference (ISWC), Nov. 2012, pp. 98-113.

[17] J. Lehmann et al., “DEQA: Deep Web Extraction for Question Answering,” Proc. of 11th International Semantic Web Conference (ISWC), Nov. 2012, pp. 131-147

[18] V. Lopez, E. Motta, and V. Uren, “PowerAqua: Fishing the Semantic Web,” Proc. of 3rd European Semantic Web Conference (ESWC), May 2006, pp. 393-410.

[19] V. Lopez, M. Sabou, V. Uren, and E. Motta, “Cross-Ontology Question Answering on the Semantic Web - an initial evaluation,” Proc. of 5th International Conference on Knowledge Capture (K-CAP), Sep. 2009, pp. 17-24.

[20] V. Lopez et al., “Scaling Up Question-Answering to Linked Data,” Proc. of 17th International Conference on Knowledge engineering and management by the masses (EKAW), Oct. 2010, pp. 193-210.

[21] V. Lopez, V. Uren, M. Sabou, and E. Motta, “Is question answering fit for the semantic web?,” A survey. Semantic Web J. Vol. 2, No. 2, July 2011, pp. 125-155.

[22] A. Simitsis and Y. E. Ioannidis, “DBMSs Should Talk Back Too,” Proc. of 4th biennial Conference on Innovative Data Systems Research (CIDR), Jan. 2009.

[23] S. Shekarpour et al., “Keyword-driven SPARQL Query Generation Leveraging Background Knowledge,” Proc. of International Conference on Web Intelligence (WI), Aug. 2011, pp. 203-210.

[24] M. Szomszor, C. Cattuto, W. V. Broeck, A. Barrat, and H. Alani, “Semantics, sensors, and the social web: The live social semantics experiments,” Proc. of 7th Extended Semantic Web Conference (ESWC), May 2010, pp. 196-210.

[25] D. T. Tran, H. Wang, and P. Haase, “Hermes: Data Web search on a pay-as-you-go integration infrastructure,” J. of Web Semantics Vol.7, No.3, Sep. 2009, pp. 189-203.

[26] C. Unger et al., “Template-based question answering over RDF data,” Proc. of 21st International Conference on World Wide Web Conference (WWW), April 2012, pp. 639-648.

[27] M. Wendt, M. Gerlach, and H. Duewiger, “Linguistic Modeling of Linked Open Data for Question Answering,” Proc. of Interacting with Linked Data (ILD), May 2012, pp. 75-86.

[28] Yokohama ART Search, http://archive.yafjp.org/test/inspection.php[accessed: 2013-09-06].

[29] T. M. Nguyen, T. Kawamura, Y. Tahara, and A. Ohsuga: “Building a Timeline Network for Evacuation in Earthquake Disaster”, Proc. of AAAI 2012 Workshop on Semantic Cities, July 2012, pp. 15-20.