CO411: Individual Project I – Semester1 Report

NLP and Semantic web/data technologies in Advancement of Business Intelligence

Abayarathna A.H.M.S.P (E/09/001)

Ilangakoon I.A.M.M.K (E/09/118)

Kuruppu K.A.D.I.M (E/09/198)

http://www.ce.pdn.ac.lk
Department of Computer Engineering
Faculty of Engineering
University of Peradeniya
Peradeniya 20400
Sri Lanka

Abstract

Business Intelligence is the future of efficient management and improves profitability in today's industry. Semantic web technologies provide much flexible knowledge gathering and processing platform on data. Data can be gathered and processed using various available technologies to find relationships base on reason. This knowledge can be stored and processed using RDF/OWL technologies and can quire by languages like SPARQL with available technology frameworks. Processing large amounts of unstructured data can be quite computation intensive process. This challenge can be overcome by parallelism and using techniques like Google Map Reduce.

Further by including Natural Language Processing the information analysis can be greatly eased out for the end user of the system. This will help the end users to make quick decisions which can lead to profitability in an organization.

Contents

Chapter 1

Introduction and Motivation

By having a business intelligence system in a business organization can assist people in the organization to make decision on the fly.

Business intelligence (BI) can be defined as the process of finding, gathering, aggregating, and analyzing information for decision making .Semantic technologies of the type advocated by Semantic Web are being applied for BI in the context of the Project. We are developing a new generation of BI tools and modules based on semantic-based knowledge and natural language processing (NLP) technology to mitigate the efforts involved in analyzing information.
.
Data can be gathered and processed using various available technologies to find relationships base on reason. This knowledge can be stored and processed using RDF/OWL [5] technologies and can quire by languages like SPARQL [4] with available technology frameworks. Processing large amounts of unstructured data can be quite computation intensive process. This challenge can be overcome by parallelism and using techniques like Google Map Reduce.

Further by including Natural Language Processing the information analysis can be greatly eased out for the end user of the system. This will help the end users to make quick decisions which can lead to profitability in an organization.

Chapter 2

2.1 Related Work

One of the researches was done to Add semantics to Business Intelligence.[1] Despite the importance of analytical tools to organizations, they still lack the inference power needed to solve the requests of decision makers in a flexible way. Their approach aims at integrating business semantics into analytical tools by providing semantic descriptions of exploratory functionalities and available services.

The researches in [3] presented the A Framework for Business Intelligence Application using Ontological Classification. Every business needs knowledge about their competitors to survive better. One of the information repositories is web. Retrieving Specific information from the web is challenging. An Ontological model is developed to capture specific information by using web semantics. From the Ontology model, the relations between the data are mined using decision tree. From all these a new framework is developed for Business Intelligence.

[6] Skyscanner is a leading global travel search site, providing instant online comparisons for millions of flights on over a thousand airlines, as well as car hire and hotels. This web site uses semantic web technologies to give the results for given queries.

Business Intelligence and search Framework Any product requires reporting, business intelligence (BI) and search functionalities. Having many ISV customers, Eurocenter has seen the requirement for an integral framework on product development in this domain.
To cater to the demands of BI and Search, Eurocenter has made a proactive investment on building a framework which is highly customizable according to customer needs.

2.2 Literature Survey

As a part of our project we did a literature survey to get an idea about the Technologies involved in the project. The Semantic web consists of mainly three technologies (Standards),

- **RDF (Resource Description Framework)**: The data modeling language for the Semantic Web. All Semantic Web information is stored and represented in the RDF.

- **SPARQL (SPARQL Protocol and RDF Query Language)**: The query language of the Semantic Web. It is specifically designed to query data across various systems.

- **OWL (Web Ontology Language)** The schema language, or knowledge representation (KR) language, of the Semantic Web. OWL enables to define concepts composably so that these concepts can be reused as much and as often as possible. Composability means that each concept is carefully defined so that it can be selected and assembled in various combinations with other concepts as needed for many different applications and purposes.

Although there other standards mentioned in some of the literatures in Semantic web these are the main three technologies.

2.2.1 RDF (Resource Description Framework)

RDF (Resource Description Framework) is one of the three foundational Semantic Web technologies.

In particular, RDF is the data model of the Semantic Web. That means that all data in Semantic Web technologies is represented as RDF. Semantic Web data is stored in RDF. Semantic Web data (typically using SPARQL) is queried from RDF data. Therefore Semantic Web data is represented in RDF. Semantic Web data is shared using RDF.

Hence, RDF is the foundation of the Semantic Web and what provides its innate flexibility. All data in the Semantic Web is represented in RDF, including schema describing RDF data. RDF is not like the tabular data model of relational databases. Nor is it like the trees of the XML world. Instead, RDF is a graph.

In particular, it's a labeled, directed graph. I don't mean "graph" as in "charts and graphs" but rather as in "dots and lines." Therefore RDF can be taken as a bunch of nodes (the dots) connected to each other by edges (the lines) where both the nodes and edges have labels.

There are three kinds of nodes in an RDF directed graph:

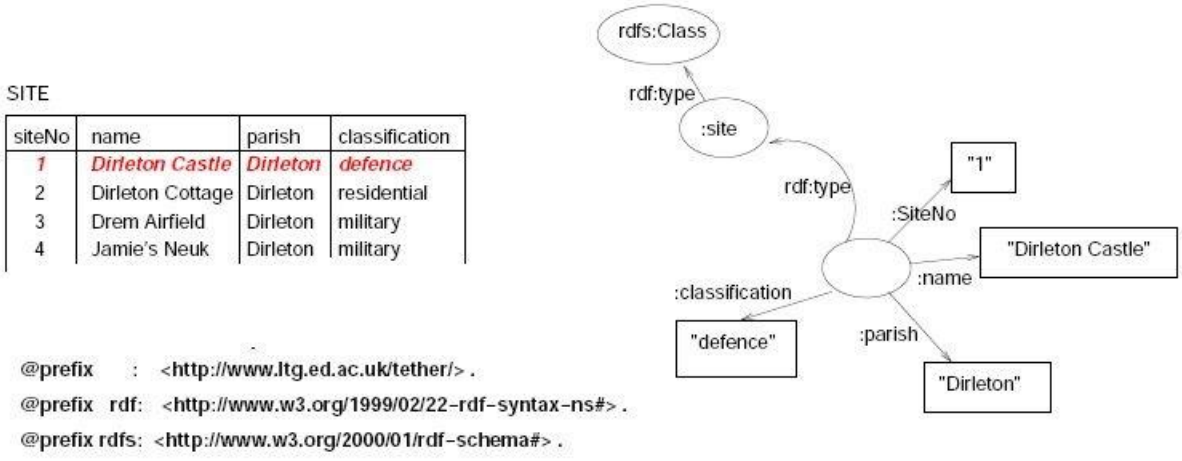
- **Resource nodes.** A resource is anything that can have things said about it. It's easy to think of a resource as a thing vs. a value. In a visual representation, resources are represented by ovals.

- **Literal nodes.** The term literal is a fancy word for value. In a visual representation, literals are represented by rectangles.

- **Blank nodes.** A blank node is a resource without a URI (Uniform Resource Identifier which denote resources).

Edges can go from any resource to any other resource, or to any literal, with the only restriction being that edges can't go from a literal to anything at all.

Following figure shows translation from relational table tuple to RDF graph.



Translation of a relational table tuple to RDF, using bnodes.

2.2.2 SPARQL (SPARQL Protocol and RDF Query Language)

SPARQL (pronounced "sparkle") is the query language for the Semantic Web. Along with RDF and OWL, it is one of the three core technologies of the Semantic Web

SPARQL is a recursive acronym, which stands for SPARQL Protocol and RDF Query Language. SPARQL consists of two parts: query language and protocol. The query part of that is pretty straightforward. SQL is used to query relational data. XQuery is used to query XML data. SPARQL is used to query RDF data. Despite this similarity, SPARQL differs in that it was designed to operate over disconnected sources over a network in addition to a local database.
In particular, the SPARQL protocol allows transmitting SPARQL queries and results between a client and a SPARQL engine via HTTP. We can take advantage of that fact to query live, public SPARQL endpoints, as we'll see later in this tutorial. A SPARQL endpoint is simply a server that exposes its data via the SPARQL protocol.

We'll cover the importance of the SPARQL protocol later in the lesson, after introducing some more basic SPARQL concepts.

The following SPARQL query has all the major components from SPARQL:

```
PREFIX      foaf:    <http://xmlns.com/foaf/0.1/>
SELECT      ?name
FROM <http://example.com/dataset.rdf>
WHERE       {
        ?x      foaf:name       ?name                                   .
}
ORDER BY ?name
```

Let's look at each component in turn.

The **PREFIX** keyword describes prefix declarations for abbreviating URIs. Without a prefix, would have to use the entire URI in the query (<http://xmlns.com/foaf/0.1/name>). Create a prefix by using a string (foaf) to reference a part of the URI (<http://xmlns.com/foaf/0.1/>). When use the abbreviation (foaf:name), it appends the string after the colon (:) to the URI that is referenced by the prefix string.

The **SELECT** keyword is the most popular of the 4 possible return clauses (more on the others later). If used SQL, SELECT serves very much the same function in SPARQL, which is simply to return data matching some conditions. In particular, SELECT queries return data represented in a simple table, where each matching result is a row, and each column is the value for a specific variable.
Using SPARQL query above in which SELECT ?name, the result would be a table with one column and as many rows as match the query. The variable ?x is not returned.

The **FROM** keyword defines the RDF dataset which is being queried. There is an optional clause, FROM NAMED, which is used to query a named graph.

The **WHERE** clause specifies the query graph pattern to be matched. This is the heart of the query. A graph pattern, as mentioned above, is, in essence, RDF with variables.

Finally, **ORDER BY** is one of the several possible solution modifiers, which are used to rearrange the query results. Other solution modifiers are LIMIT and OFFSET.

Here is the SPARQL query for following SQL Query

SPARQL Query

```
SELECT DISTINCT ?name ?email WHERE {
  ?person rdf:type foaf:Person.
  ?person foaf:name ?name ;
          foaf:mbox ?email
}
LIMIT 10
```

SQL Query

```
mysql> select FirstName, LastName, email from persons;
+-----------+-------------+-----------------------------+
| FirstName | LastName    | email                       |
+-----------+-------------+-----------------------------+
| Yolanda   | Gil         | gil@isi.edu                 |
| Varun     | Ratnakar    | varunr@isi.edu              |
| Jim       | Blythe      | blythe@isi.edu              |
| Andreas   | Eberhart    | eberhart@i-u.de             |
| Borys     | Omelayenko  | borys@cs.vu.nl              |
| Andy      | Seaborne    | andy.seaborne@hpl.hp.com    |
| Alberto   | Reggiori    | areggiori@webweaving.org    |
| Sonia     | Bergamaschi | bergamaschi.sonia@unimo.it  |
| Francesco | Guerra      | guerra.francesco@unimo.it   |
| Christian | Bizer       | chris@bizer.de              |
+-----------+-------------+-----------------------------+
10 rows in set (0.00 sec)
```

2.2.3 OWL (Web Ontology Language)

OWL (or **W**eb **O**ntology **L**anguage) is the ontology (think "schema") language of the Semantic Web. It is one of the core Semantic Web standards and must be familiar with, along with RDF and SPARQL.

Its two primary uses are:

1. Fast and flexible data modeling
2. Efficient automated reasoning

OWL is a modeling language Although OWL is a modeling language in the classical sense, it has many advantages compared to the modeling languages that came before it.

OWL is Expressive, Flexible and Efficient. OWL's expressiveness, flexibility, and efficiency make it an ideal modeling language for creating web ontologies that represent exceptionally complex and refined ideas about data

Chapter 3

Project Objective

The main objective of our project is to do a comprehensive analysis of semantic web and data technologies in making structured data more meaningful for an organization and also find tools and their capabilities in delivering the above requirements. Then use NLP in generating queries on semantic data space to create working software.

Chapter 4

Proposed Solution

In this Project we are going to develop business Intelligence software using semantic web and Natural Language Processing. By having a business intelligence system in a business organization can assist people in the organization to make decision on the fly. In this project we are going to use Agile Methodology for developing the software. Also we have planned to maintain a blog about the work carried by each group members. First of all, we are going to study the technologies we can use in this project (RDF , OWL , SPARQL etc.).

Then we are going to use a dummy relational database as the input data for the project. Then this database is converted to Resource Description Framework (RDF). In order to this we will have to find a library/tool to convert the relational database to RDF. If not we have to develop a library to do this work. When we are able to convert the relational database to RDF we can extend this process to gather data from other sources such as web resources, email to RDF.

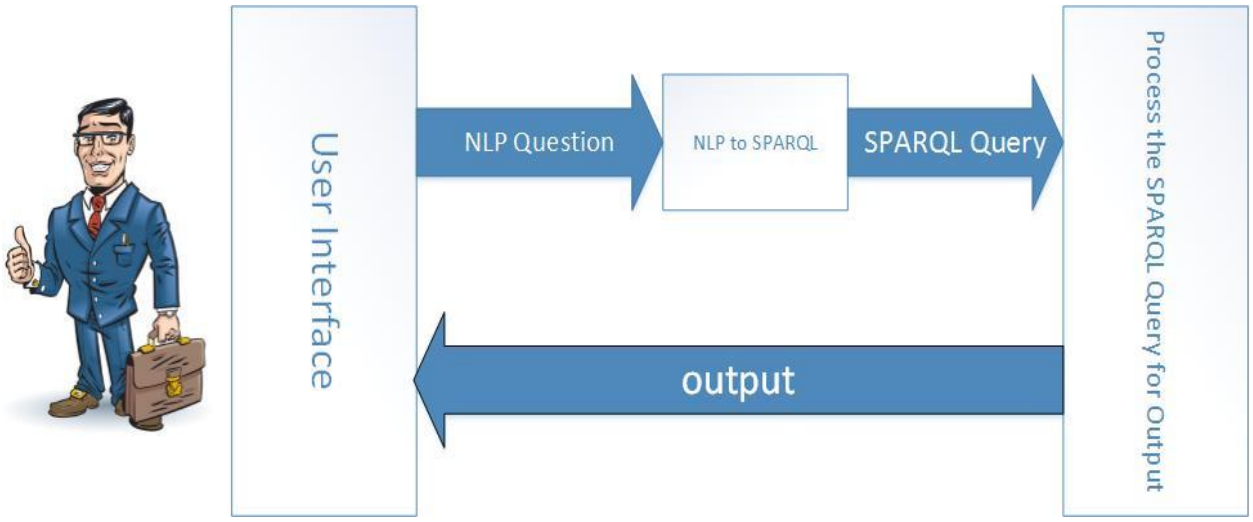
After creating the RDF file we are going to use SPARQL queries to get data from RDF file. In order to do that we will have to study about the SPARQL and find libraries to query RDF file using SPARQL in Java.

But the business people don't know SPARQL. So we are going to use Natural Language Processing (NLP) to convert Natural language inputs to relevant SPARQL queries and get the output.

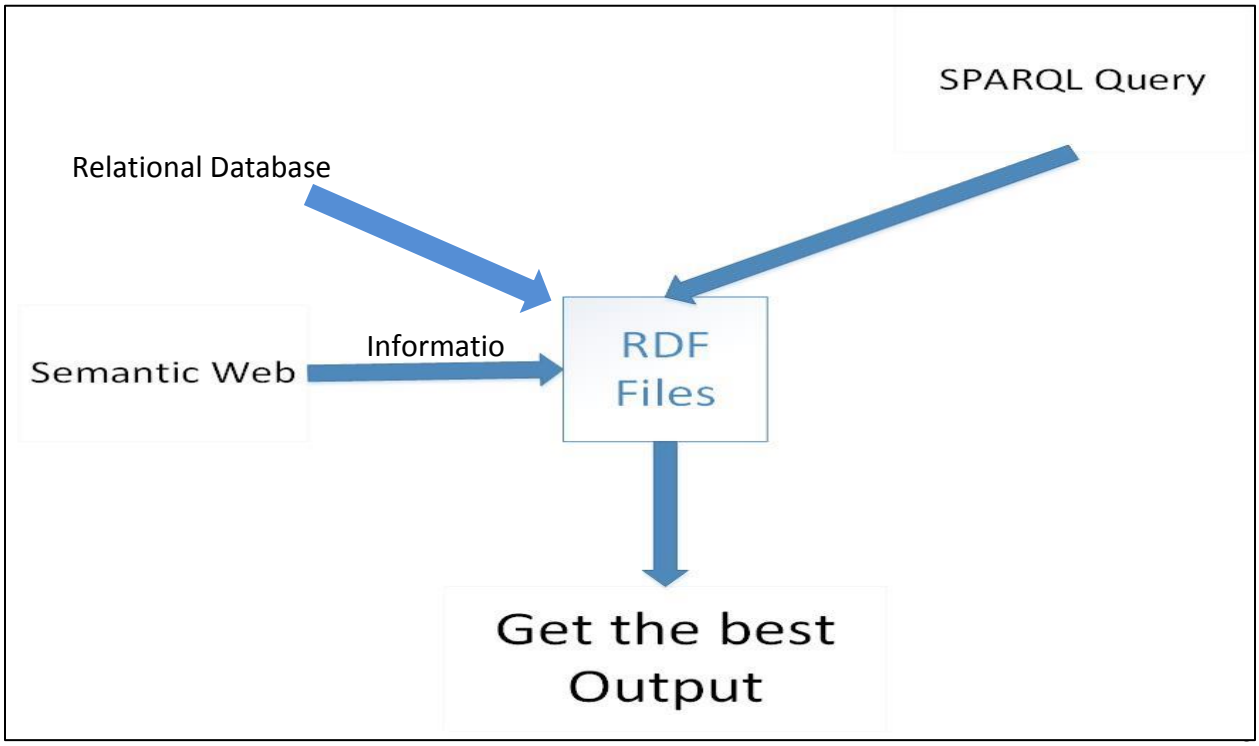
.

Chapter 5

System Design and Justification



Process the SPARQL Query for the best output

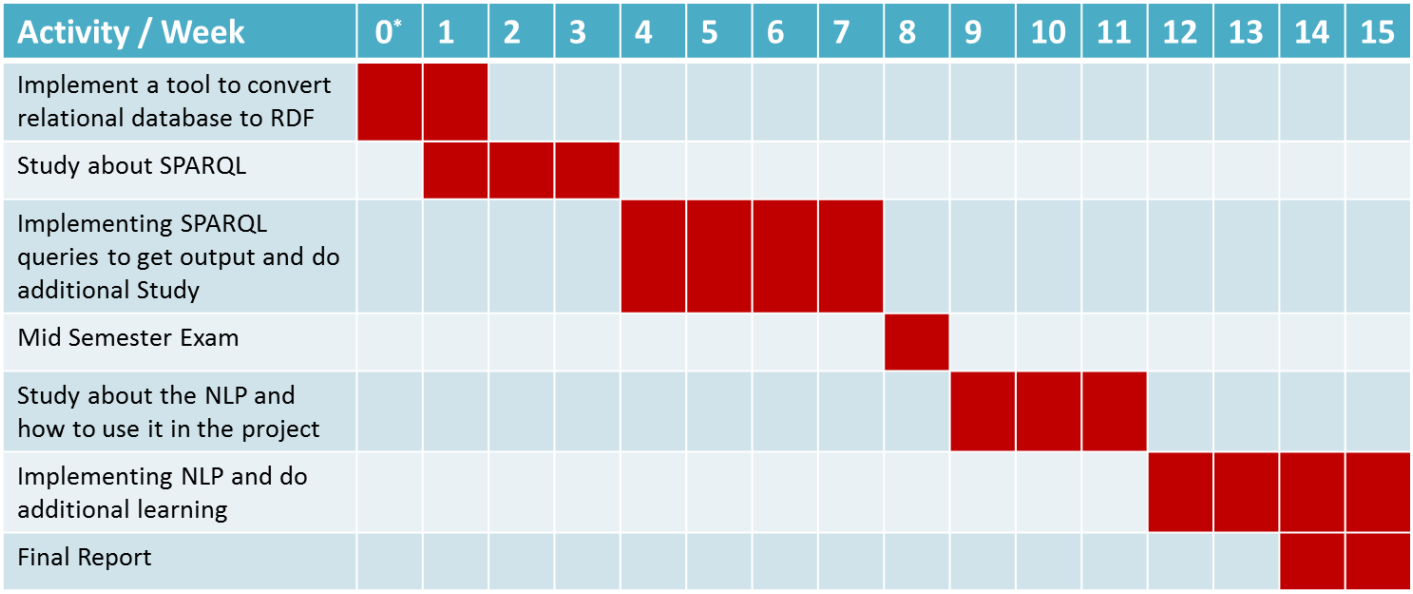


The main objective of the project is to get information from semantic web or Relational Database using NLP (Natural Language Processing). In order to do that when the question is given in Natural Language by the user, first it should be converted to the SPARQL queries using NLP. Then this query should be used to query out the RDF file which is acquired from semantic web of created from Relational Database. Then the output for the SPARQL Query should be shown in the user interface in way that the user can understand easily.

Using the above system design we can see that the main objective of the project can be accomplished.

Chapter 6

Work Plan



*** Vacation Week**

Reference

[1] Denilson Sell , Liliana Cabral , Enrico Motta , John Domingue and Roberto Pacheco Stela Group, Universidade Federal de Santa Catarina, Brazil Knowledge Media Institute, The Open University, Milton Keynes, UK INE, Universidade Federal de Santa Catarina, BrazilAdding Semantics to Business Intelligence.

[2] http://www.cambridgesemantics.com/semantic-university/

[3] A. Martin , D.Maladhy , Dr . V . Prasanna Venkatesan: A Framework for Business Intelligence Application using Ontological Classification, 2011 IJEST

[4] Foundation of Semantic web technologies, Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph

[5] Brickley, D., & Guha, R. V. (Eds.). (2004). RDF vocabulary description language 1.0: RDF schema. Retrieved from http://www.w3.org/ TR/ rdf-schema/

[6] http://www.skyscanner.net/