Diagnosis Aider

CSEN 915 Semi-Structured Data and the Web

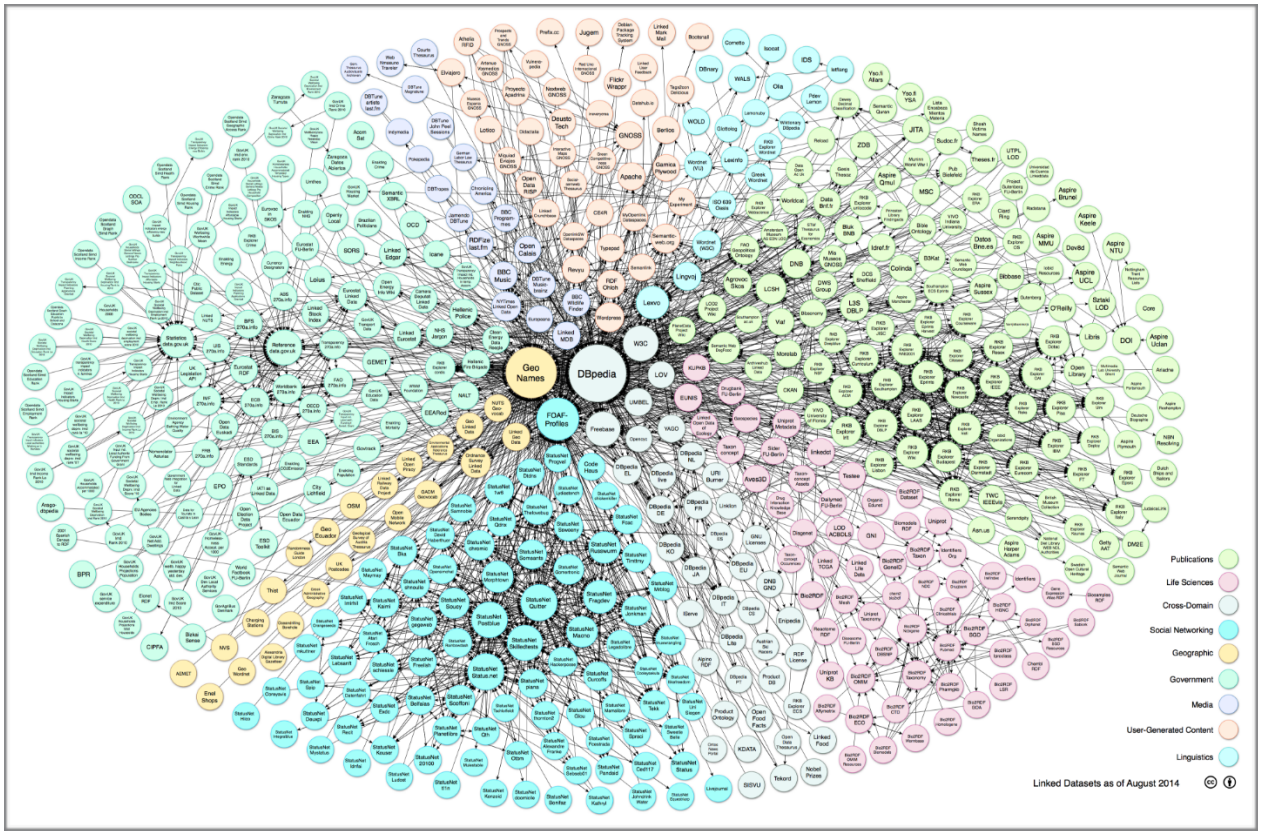

Figure 1: An illustration showing the datasets published in Linked Data format and how different data sets are interlinked together.
source: http://lod-cloud.net/

Team members:

Madeleine Aziz

Hend Serry

Lin Kassem

1. Introduction

The term 'semantic web' was first coined by Tim Berners Lee in 2001[1], were he had the vision that data on the internet is very valuable and with unlimited potential. Tim Berners Lee proposed that if the data was written in a standard format across the web and with relations stated between the data , the linked-data format [2], this will enable scientists from different backgrounds to collaborate together, make discoveries and detect patterns in the data. Thus empowering scientists to solve major challenges; like curing cancer or Alzheimer's. He also suggested that this linked-data standard format will make the data more understandable for the machines as well. Thus, opening new roads and possibilities in the fields of artificial intelligence and machine learning.

Currently there are huge efforts towards making people, organizations and governments publish their data in the linked-data standard format. DBPedia [3] is one of the leading projects that aim to collect data from the web and put them in a the linked-data structured and standardized format. There are several semantic web applications that make use of linked-data. These semantic web applications are being used by companies in various fields, such as media, medical research and even by oil companies [4].

In this report we will propose and introduce a semantic web application called 'Diagnosis Aider'. In section 2, the idea of the semantic web application is explained. Along with a scenario illustrating how the application can be used. Moreover, the technologies used during the implementation phase of the application is explained. Finally, in section 3 the report is concluded and future work and proposals are mentioned.

---

[1] Berners-Lee, Tim; James Hendler; Ora Lassila (May 17, 2001). "The Semantic Web". Scientific American Magazine. http://www.scientificamerican.com/article/the-semantic-web/

[2] Berners-Lee, Tim; Heath, Tom; Bizer, Christian Bizer. Linked Data - The Story So Far . http://tomheath.com/papers/bizer-heath-berners-lee-ijswis-linked-data.pdf

[3] http://wiki.dbpedia.org/

[4] http://www.cambridgesemantics.com/semantic-university/example-semantic-web-applications

2. Diagnosis Aider

The aim of 'Diagnosis Aider' system is to help doctors during the phase of diagnosis of patients seeking medical care. 'Diagnosis Aider' will empower doctors with the ability to use and investigate linked data between the patient's family medical history, patient's current disease symptoms along with a data set of medical diseases and their associated information. Thus, allowing the doctors to diagnose diseases correctly and in a shorter period. Ultimately affecting the health of patients and speeding up their recovery.

During the implementation of 'Diagnosis Aider', we have made the assumption that it will be used in a health care system that stores the data of the patient's family relations and the patient's family medical history in a standard linked-data format (or ontologically based data format).

1.1 *Scenario*

In this section we will illustrate a simple scenario to show how the 'Diagnosis Aider' system can be used.

The scenario starts with a patient suffering from particular symptoms coming to the doctor seeking for a medical diagnosis. The doctor will take the patient's medical id and enter it in the system. The system will check if there are records for the patient or not. If the patient is found, the doctor is asked to enter the symptoms that the patient has.

The system will output a result set of all the possible diseases that the patient might be suffering from. The system assumes that the probability that a patient is suffering from one of the diseases in the result set is [1/ (number of diseases in the result set)].

Then for every disease in this set, the family history of the patient will be checked to see if there is an existing family member who is already suffering from the same disease. If a family member was found to be suffering from the same disease, the probability of the patient likelihood to be suffering from the same disease is incremented.

After scanning the data of all the family members of the patient, the system will output the results to the doctor in a clear and readable format. Thus, allowing the doctor to base his diagnosis not only on the patient's symptoms but taking consideration of the family history as well.

1.2 C*onsole Demo*

In this section we will illustrate a working demo of the system. The current demo is run through the console and is with only a basic user interface.

For this demo we used manually entered data set to represent a family and the diseases they have. The data set along with the relations and properties is illustrated below, in Figure 2.

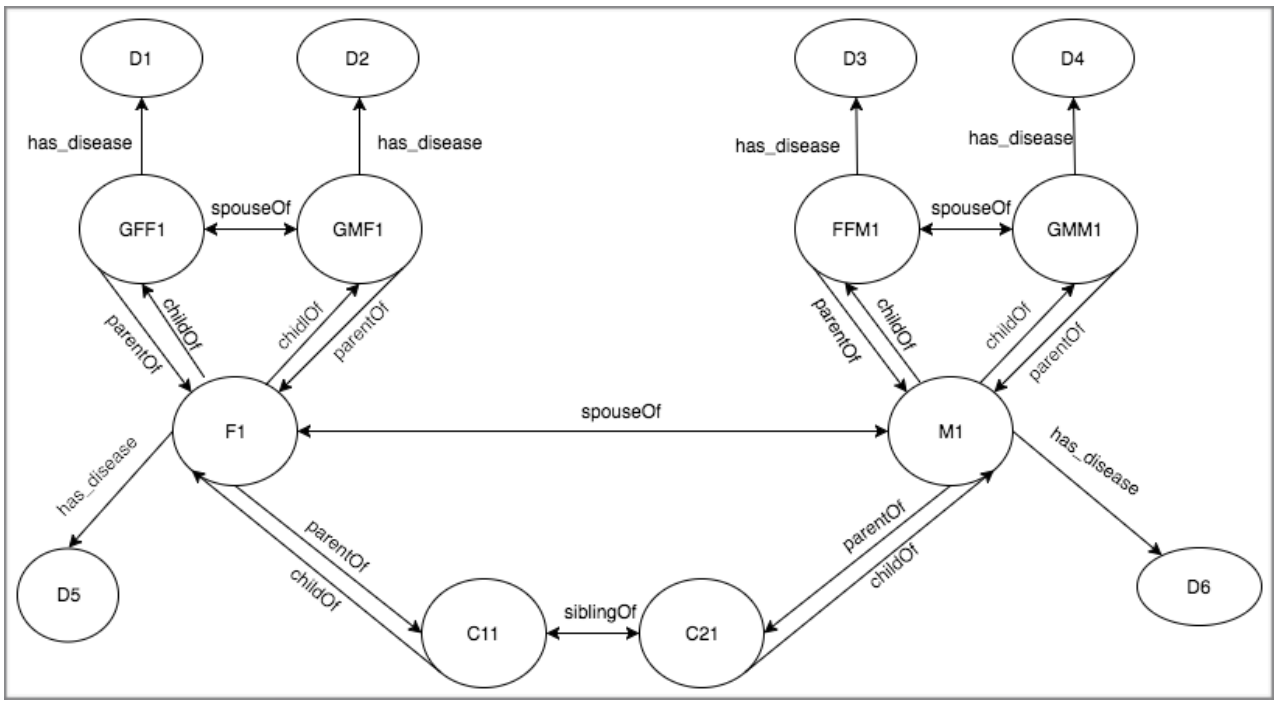

Figure 2: Graphical Representation of the data used in the console demo

```
-> Start of Console Demo

Welcome to *Diagnosis Aider*
Please enter your ID
c11

Patient found. Please enter your symptoms separated by ','
then press enter when you are done.
s1, s7, s12, s18, headache, bone-ache, back pain

Symptoms read.

Processing!
```

```
===============
Results Ready:
===============
Disease Name: d1
Disease URI: <http://purl.bioontology.org/ontology/DOID/D1>
Probability of having this disease according to inheritance
factor: 1.25
Family members/ancestors with the same disease:
    <http://www.met.guc.edu.eg/#GFF1>
-------------------------------------------------
Disease Name: d3
Disease URI: <http://purl.bioontology.org/ontology/DOID/D3>
Probability of having this disease according to inheritance
factor: 1.25
Family members/ancestors with the same disease:
    <http://www.met.guc.edu.eg/#GFM1>
-------------------------------------------------
Disease Name: d4
Disease URI: <http://purl.bioontology.org/ontology/DOID/D4>
Probability of having this disease according to inheritance
factor: 1.25
Family members/ancestors with the same disease:
    <http://www.met.guc.edu.eg/#GMM1>
-------------------------------------------------
Disease Name: d6
Disease URI: <http://purl.bioontology.org/ontology/DOID/D6>
Probability of having this disease according to inheritance
factor: 1.25
Family members/ancestors with the same disease:
    <http://www.met.guc.edu.eg/#M1>
-------------------------
-> End of Console Demo
```

Technologies used

In this section we will illustrate the libraries and ontologies used to build 'Diagnosis Aider'.

FOAF [5] ontology was used to represent family individuals and their properties. The RELATIONSHIP [6] ontology was used to model the relation between family members, e.g. John is *fatherOf* Mary. The RELATIONSHIP ontology is built on top of the foaf ontology, in other words it compliments the FOAF ontology. The Human Disease [7] ontology was used to represent diseases and their properties.

The API provided by Apache Jena java based framework [8] was used to interact with, extract information from and read the above mentioned ontologies. Apache Jena API provides the possibility to build our own model (a container like structure) and fill this model with the ontologies, properties and statements that we want and build. This model can then be queried and altered using the SPARQL [9] query engine.

The technology to build the web service version of this system is still under research. Java Play framework [10] was considered for it's modern and clean MVC [11] based structure, however we are facing difficulties in integrating the Apache Jena API with it. Another consideration is using the JSP [12] java pages technology, however with the huge advancements in the web JSP java pages is considered to be an old technology with very little support.

---

[5] http://www.foaf-project.org/

[6] http://vocab.org/relationship/

[7] https://bioportal.bioontology.org/ontologies/DOID

[8] https://jena.apache.org/

[9] https://www.w3.org/TR/rdf-sparql-query/

[10] https://www.playframework.com/

[11] http://blog.iandavis.com/2008/12/what-are-the-benefits-of-mvc/

[12] http://docs.oracle.com/javaee/5/tutorial/doc/bnagy.html

1.3 *Java Code*

In this section we will illustrate the classes and the java methods used to build the system. There are two main classes, DA.java and Disease.java.

Methods in DA.java

- public void populateModel():
  Populates the model used through out the system with the appropriate ontologies, classes, individuals, properties & statements.
- public void printRdfToFile():
  Prints the RDF representation of the model to an output.rdf file.
- public String convertIdToURI(String ns, String id):
  Converts an id to a URI.
- public boolean patientExists(String patientURI):
  Checks if the input patient URI exists in the model or not.
- public String convertToValuesFormat(String symptoms):
  Converts a comma separated input string to the format of SPARQL 'VALUES' which is { "value1" "value2"}.
- public ArrayList<Disease> getPossibleDiseases(String symptoms):
  Given some symptoms, all diseases with these symptoms are found and returned.
- InitializeDiseaseProbability(ArrayList<Disease> diseases):
  Given an ArrayList of Diseases, the probability attribute of the every disease in this array list is set to 1/the number of diseases in the array list.
- public void recursiveFamilyCheck(String patientURI, ArrayList<Disease> possibleDiseases):
  Recursively checks the parents of the patient URI and checks if they have any of the diseases in the possibleDiseases array list. Accordingly every disease in the possibleDiseases arrayList is updated.
- public void hasDisease(String personURI, ArrayList<Disease> diseases):
  Given a person URI and an array list of diseases, the method checks if this person has this disease or not. Accordingly every disease in the possibleDiseases arrayList is updated.
- public void printResults(ArrayList<Disease> possibleDiseases):
  Loops over all the Disease objects in the possibleDiseases array and prints the relative properties & attributes of this disease.

Methods in Disease.java

This class is used to represent the Disease object. Where a Disease object has a *name*, *diseaseURI*, *probability* which represents the probability of having this disease and an *ArrayList strings* representing the people who suffer from this disease. The class is composed of a constructor and setters and getters for the class attributes.

3. Conclusion & Future Work

We have implemented a semantic web application called 'Diagnosis Aider' that aims in providing the doctors with meaningful data about the patients family disease history. Moreover the system helps the doctors in linking the diseases with their symptoms.

Currently the system is seeded with information about the diseases and information about family trees and histories manually. Future versions of the system should allow the integration of already existing datasets (e.g. uploading a file containing an RDF representation of the family relations and their disease history [13]).

Moreover, in the current version we calculate the probability of having a disease in a non-scientifically based way (e.g. we increment by1 for every family member having the same disease). In the future, the probability calculation should be based on scientific calculations and evidences.

Also practical testing for the system should take place, and accordingly error margins should be estimated. During testing we could use the records of already diagnosed patients with scientific evidence that they were correctly diagnosed. For example we know that this patient has cancer because he was already diagnosed by a doctor and we have lab reports confirming that the patient indeed has cancer. We could use this patient as a test for our system. Where we will enter the patient family history and query his symptoms and check if the system will output cancer as one of the possible diseases or not and accordingly we can calculate error margins of the system.

Further more, while matching the symptoms agains the diseases we should eliminate any general symptoms. This is to limit the search space and produce more accurate results and thus better diagnosis.

[13] http://lod-cloud.net/