MedSearch: A Semantic Search Engine to find diseases, its symptoms and treatment.

[Hassan Alhuzali]

[halhuzal@indiana.edu]
School of Informatics & Computing
Indiana University

[Archana Singh]

[arcsingh@indiana.edu]
School of Informatics & Computing
Indiana University

Abstract

There has been an upsurge in efforts, in recent times, to align healthcare with semantic web. Proponents of healthcare and biomedical research are starting to acknowledge the benefits of sharing data across different domains and are finding ways to reap the benefits of semantic web. This paper highlights the work done in order to create MedSearch, a semantic search system that uses Jena Reasoner and SPARQL to display list of symptoms and treatment available for the list of diseases available in the dataset. We describe the process of (1) creating our own dataset with list of diseases, its symptoms and treatment (2) building the OWL/RDF ontology using Protégé (3) building a Jena Reasoner and use SPARQL query to display results and finally (4) use Parametrized query to display results based on user input.

Introduction

Tim Berners-Lee et al. defines semantic web as "extension of the current one (web), in which information is given well-defined meaning, better enabling computers and people to work in cooperation" [i]. Semantic web is transforming web right now, and paving way for the creation of Web 3.0, which is an even more connected web of network, where entities are able to collaborate and share information at a much higher rate, thus enabling huge transfer of data, concept and ideas. Healthcare and Life Sciences industry can benefit majorly from this movement. There is a lot of data right now floating around us, but there are very few outlets where users can receive credible or useful information. There are websites like WebMD, Mayo Clinic's website, Medicinenet that have a plethora of disease or drug related information for users, but the sheer volume of information could be intimidating for some users. Cheung et. al. (2009) when describing a semantic search system for healthcare say that it should, "make well-informed decisions that may lead to important scientific breakthroughs. For this to be achieved, diverse types of data about drugs, patients, diseases, proteins, cells, pathways and so on must be effectively integrated." [ii] This lack of an informative, accurate yet simple search system on the web, was motivational and acted as a starting point for the development of MedSearch. The objective was to implement a system that stores information about diverse types of disease, right from common cold and flu to globally relevant diseases like Ebola and Zika Fever, and integrate it effectively so it can lead to users making well informed decisions [ii].

To achieve the objectives mentioned we first created a data set of accurate and reliable information regarding a list of 15 diseases that included its common symptom and treatment. Following that we utilized Protégé, an open source ontology editor, to create an OWL/RDF ontology. The ontology has 3 classes Disease, Symptom and Treatment, and included 3 object properties association, hasSymptom and treatWith. The ontology also had object property assertions for inferences. Next, we used Jena Reasoner to reason with our ontology and gather results based on inferences. Then we used SPARQL query to retrieve information stored in our RDF. Finally, we implemented parameterized query using Java, where a user can get results based on a query term. The query term is searched against our RDF and appropriate results are then displayed.

Related Work

Mohammed. O, Benlamri. R, and Fong, S (2012) published a paper regarding Building a Diseases-Symptoms Ontology for Medical Diagnosis. This ontology was designed by mapping between a disease and its symptoms. The way that the authors built such an ontology was through using a disease ontology (DOID) and a symptom ontology (SYMP) to align both together, for instance, a disease could have one or multiple symptoms. In addition, Protégé was the tool that was utilized to implement this ontology, representing the set of disease classes as the domain of the property, while the set of symptoms as the range. The proposed alignment method has been repeated for 11 inter-related diseases and as a result this method could be used for any number of disease to build a larger version. Overall, this disease-symptom ontology is published online on the authors' university site so that interested researchers in this field could benefit from the ontology and also contribute to it. [iii]

A vital point for a semantic search system is to apply both Querying and reasoning, which are an aspect of semantic web technologies. As a result, we are interested in using OWL and SPARQL as both support Querying and reasoning. (Martin, Moulin, & Sbodio, 2010) discuss that the relation of SPARQL to OWL is well defined and well-understood. SPARQL, as it described further by (Martin, Moulin, & Sbodio, 2010), also contributed to give high degree of flexibility and of meaningfulness. [iv]

Finally, Hadzic, & Chang (2005) described that each disease is uniquely identified by its disease's type, symptom, and treatment. As a result, this support our approach of building disease symptom treatment ontology semantic search where a disease serves as a unique form for finding both symptom and treatment belonging to that given disease. [v]

Data Collection

We decided on creating our own data set for the purposes of creating the semantic search system. Although there are multiple publicly available datasets like OpenDDX and HuDiNe, they are very extensive and don't contain records in the correct format for our query (sometimes due to HIPPA restrictions). Additionally, the decision to create our own dataset was triggered by the sentiment to have an open dataset that could be built upon and easily updated, depending on future work. To start off, we searched the web extensively to create a list of diseases. The idea was to create a list of some commonly occurring ailments like Flu, some other widely searched for diseases like Leukemia and other diseases that have surfaced in the past few years like Ebola and Zika fever, so our dataset is as inclusive as it could be. We narrowed down the list to 15 diseases which are *Flu, Chronic Kidney Disease, Fibromyalgia, Hepatitis, Leukemia, Rheumatoid Arthritis, Sleep Apnea, Food Allergy, Asthma, Epilepsy, Diabetes, Tuberculosis, Zika fever, Ebola and Traumatic Brain Injury*. After finalizing the list of diseases the most common list of symptoms was created followed by the list of common treatments to cure diseases in our list. Our list of treatments includes various drugs or home treatment options for various diseases in the list.

Method

The ontology for MedSearch, disease-symptom semantic search system, was created and a semantic based rule was designed to respond to the corresponding user query. The ontology consisted of three classes, namely disease, symptom, and treatment. Furthermore, the ontology also contained three object properties (hasSymptom, associate, and treatWith). For each class, there were 15 instances that were created belonging to each class. Table 1 displays the examples of each class across the whole dataset.

Instance	Name of disease	Name of symptom	Name of treatment
1	Chronic Kidney Disease	Poor appetite	Paricalcitol
2	Fibromyalgia	Trouble sleeping	Physical exercise
3	Hepatitis	loss of appetite	Avoid alcohol
4	Leukemia	Persistent weakness and fatigue	Blood transfusion
5	Rheumatoid Arthritis	Fever	Stretching
6	Sleep Apnea	Headaches in the morning	Weight loss
7	Flu	cough	Peramivir
8	Food Allergy	Skin rash/swelling	Zyrtec
9	asthma	difficulty breathing	Methylprednisolone
10	Epilepsy	muscle spasms	Primidone
11	Diabetes	increased thirst	Insulin aspart
12	Tuberculosis	chest pain	Rifabutin
13	Zika fever	joint pain	Benadryl
14	Ebola	muscle pain	blood transfusion
15	Traumatic Brain Injury	amnesia	Mannitol by injection

Table 1: Dataset instances

Figure 1 is a snapshot of Protégé that displays our ontology and few of the instances we created.

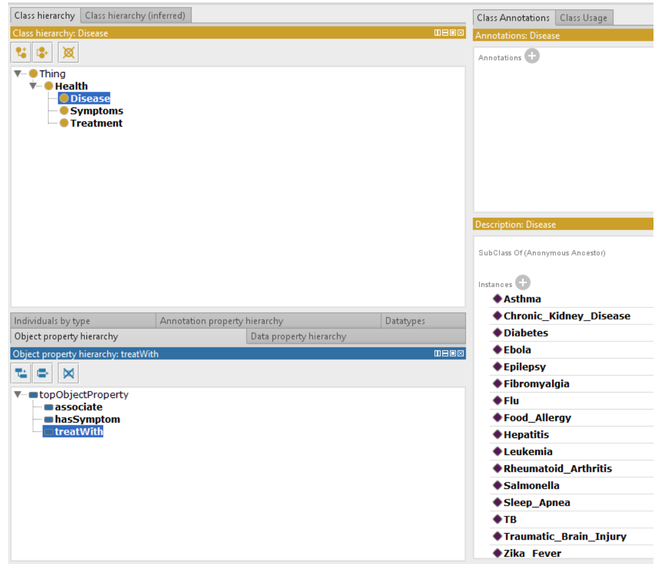

Figure 1: Ontology in Protégé

In addition, object property assertions for both disease and symptom classes were created, while the treatment class object assertions were not modified, so the reasoner would be able to infer the relationship between the disease class and treatment class. The type of rule that we have implemented is a transitive functionality, which can be illustrated as A has a relation with B, B has a relation with C, so ultimately A has a relation with C. Figure 2 shows the association that was implemented.

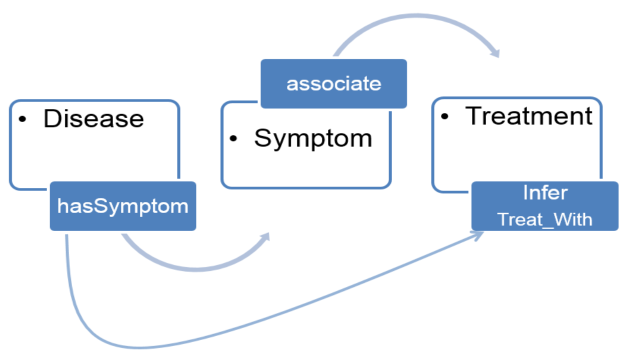

Figure 2: Associations between class and properties

The implementation of Jena Reasoner was also based on similar principles of transitive functionality and inference. In similar fashion, the reasoner will infer this transitive relationship between the disease class and symptom class. To test the functionality and build further upon our reasoner, the team utilized SPARQL to implement some queries. First and foremost, a query was executed that displayed a table with list of diseases, its symptoms and treatment associated with it. This association was built upon the inferences by Jena reasoner. Since the system was able to infer treatment associated to a particular disease and display it accurately, our reasoner was working as we expected. Finally, the team decided to include a function to perform parameterized query so users can perform search based on disease name. This added functionality elevated the semantic search system we had by introducing a feature that made it more user friendly. Users can input the name of the disease into command prompt and retrieve its respective symptom and treatment. This process was implemented by executing a Java code that requested for user input, stored it in a string and searched through our ontology, to retrieve the information it's looking for and present it to the user.

Result

The team was successful in implementing the reasoner and queries including the parameterized query. The initial query to display the list of diseases and its associated symptoms and treatment worked great and the reasoner was able to successfully infer relationships and display treatment associated with the particular disease. Figure 3 is the result we obtained by executing query 1. That shows a successful implementation of our reasoner and query.

disease	symptom	treatment
:TB	:Chest_pain	:Rifabutin
:Hepatitis	:Fever	:Stretching
:Food_Allergy	:Skin_rash_swelling	:Zyrtec
:Diabetes	:Increased_thirst	:Insulin_aspart
:Epilepsy	:muscle_spasms	:Primidone
:Ebola	:Muscle_pain	:Blood_transfusion
:Rheumatoid_Arthritis	:Fever	:Stretching
:Leukemia	:Fatigue	:Blood_transfusion
:Zika_Fever	:Joint_pain	:Benadryl
:Fibromyalgia	:Trouble_sleeping	:Physical_exercise
:Flu	:cough	:Peramivir
:Chronic_Kidney_Disease	:Poor_appetite	:Paricalcitol
:Asthma	:difficulty_breathing	:Methylprednisolone
:Sleep_Apnea	:Headache	:Weight_loss
:Traumatic_Brain_Injury	:Amnesia	:Mannitol_by_injection

Figure3: Query 1 results

Our execution for parameterized query was also successful. A user is able to retrieve results based on their input. For example, when searching for the disease ‘Leukemia’ the system is able to retrieve and display the symptom as ‘Fatigue’ and treatment as ‘Blood Transfusion’. This was a breakthrough and definitely helped us elevate our system to the next level, as this was probably the most difficult aspect of the system to implement. Figure 4 shows some examples of the query in action. The user input is not case sensitive and the system will display a blank result when the user queries a disease that not in the dataset.

```
(Canopy 64bit) C:\Jena>java user_input
Semantic search system
Enter a disease: Leukemia
-----------------------------------------------
| symptom         | treatment                 |
===============================================
| health:Fatigue  | health:Blood_transfusion  |
-----------------------------------------------
```

```
(Canopy 64bit) C:\Jena>java user_input
Semantic search system
Enter a disease: flu
-----------------------------------
| symptom       | treatment        |
===================================
| health:cough  | health:Peramivir |
-----------------------------------
```

```
(Canopy 64bit) C:\Jena>java user_input
Semantic search system
Enter a disease: Zika fever
--------------------------------------
| symptom            | treatment      |
======================================
| health:Joint_pain  | health:Benadryl |
--------------------------------------
```

```
(Canopy 64bit) C:\Jena>java user_input
Semantic search system
Enter a disease: hepatitis
-------------------------------------
| symptom       | treatment          |
=====================================
| health:Fever  | health:Stretching  |
-------------------------------------
```

Figure4: Parameterized Query results

Conclusion and Future Work

Through the process of implementing MedSearch, a semantic search system, the team was able to acquire great insights and knowledge of semantic web, RDF/OWL Ontologies, Jena Reasoner and SPARQL query. The team was able to successfully attain objectives set forth and was able to build a dataset that could be utilized to build MedSearch. The implementation of

Reasoner was conducted successfully and parameterized query integrated seamlessly into MedSearch. But this is not the end for MedSearch. There are a bunch of activities that the team hopes to carry on in order to implement MedSearch on a broader and elevated scale. First and foremost, there is a need to expand the dataset, from current list of 15 diseases to probably many more, as well as include other classes like Prescription information for drugs etc. It would also be ideal if this could somehow be implemented by automating the expansion, instead of manual entry that was performed for MedSearch. The team also aims to benefit from Linked Open Data and integrate other medical databases, in order to retrieve more information and display to the users. Finally, the team plans to create a website as the front end for the search engine, so it's even more user friendly.

MedSearch was built with the idea to make it easier for any user to find symptoms and treatment associated with a disease through utilizing Semantic Web. At this point the system does a great job of attaining this objective and much more. With the implementation of ideas for future work, MedSearch will be able to perform even better than it is right now, and hopefully will be able to transform the health sciences industry presence on the web.

References

[i] Berners-Lee, Tim, James Hendler, and Ora Lassila. "The semantic web." *Scientific american* 284.5 (2001): 28-37.

[ii] Cheung, Kei-Hoi, et al. "Semantic Web for Health Care and Life Sciences: a review of the state of the art." *Briefings in bioinformatics* 10.2 (2009): 111-113.

[iii] Mohammed, O., Benlamri, R., & Fong, S. (2012). Building a diseases symptoms ontology for medical diagnosis: An integrative approach. *The First International Conference on Future Generation Communication Technologies* (pp. 104 - 108). London: IEEE.

[iv] Martin, D.L., Moulin, C., & Sbodio, M.L.. (2010). Discovering Semantic Web services using SPARQL and intelligent agents. *J. Web Sem., 8*, 310-328.

[v] Hadzic, M., & Chang, E. (2005). Ontology-based Multi-agent Systems Support Human Disease Study and Control. *SOAS*, *135*, 129-141.

Contribution of Each Team Member

Archana

- Research dataset on the web
- Create dataset in csv format and assist with Ontology creation
- Generate a workable code for reading the owl/rdf file, creating the reasoner and running query
- Create user input parameterized query
- Create presentation
- Research for and creation of final project paper

Hassan

- Research dataset on the web
- Create Ontology and assist with data set creation in csv format
- Generate a workable code for reading the owl/rdf file, creating the reasoner and running query
- Create additional queries to go along with parameterized query
- Create presentation
- Research for and creation of final project paper