

A Tool for Symptom Querying and Diagnosis

Problem Statement:

Web services such as WebMD's symptom checker are helpful modern tools that allows anyone on the web to type in their symptoms and see what diseases match their inputs. Although they have been criticized as overly broad with users self-diagnosing themselves with rare diseases, they can be helpful tools to see what the possible diseases are for different symptom sets. Our project aims to replicate some of this functionality in a disease ontology capturing the 50 most common diseases patients have.

As a simpler version of more complicated disease ontologies used in clinical practice and in research groups, this project is less powerful but more understandable. It can be used for instructional purposes to better visualize related diseases, to expose students to some of the most common diseases facing patients, and to observe an easy to understand and modify disease ontology.

Technical Needs:

For technical feasibility, our model aims to capture just the most common diseases patients face (we aim for the 50 most common). Although this is only a small fraction of all known diseases (such as everything found on the Disease Ontology database), modeling a handful of the most common disease covers a large number of patients.

The steps to complete this project include: (1) compile a list of the 50 most common diseases and associated information, including symptoms and treatment options. We can then (2) compile the information into an OWL-base ontology for clarity. Next, we need to (3) specify how a user is to interact with the system to query for information on different diseases and on how to check any symptoms against what exists in the ontology. As the user gives more data to the program, it will ask different questions to see narrow down a differential diagnosis and present different diseases matching those symptoms. User interaction will The ontology will be built in Protege. Interaction with the user will be executed in Python 2.7 with a Tkinter GUI.

Team Structure:

Andy Vu - Data gathering of different diseases, handle GUI and user interaction
Brian Liu - Digital construction of ontology, poster presentation