Project Final Report

Brian Liu

March 16, 2018

1 Abstract

1.1 Background

In summary, this is a simpler version of a symptom checker that uses a watered-down disease ontology.

1.2 Methodological Approach

Although the original plan was to use OWL, Protégé, and Python with a Tkinter GUI, the final product only uses Python for **classification and investigative deduction**[1].

1.3 Results

Overall, the program performs one main function well:

- Given a small user-inputted list of symptoms, it asks the user more and more questions about the disease's symptoms until the singular disease is deduced.

1.4 Discussion

Overall, the basic functionality and logic are intact, but these are the top two features to be added for the most immediate improvements for the future:

- Call Google's medical knowledge graph API, or Google's Knowledge Graph API[2], to more thoroughly and comprehensively populate the ontology of the most common and most commonly searched diseases.
- Compile information into an OWL-based ontology and link the current Python interface's functionality with the OWL API.

[1] Disclaimer: classification was listed as a problem solving method in the slides that seemed to fit the problem here, but its use here may not be completely precise, which is why I also use "investigative deduction" to be more descriptive.

[2] `https://www.google.com/intl/en_us/insidesearch/features/search/knowledge.html`

2 Background/Motivation

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 the following instructional purposes:

- to better visualize related diseases
- to expose students to some of the most common diseases facing patients
- to observe an easy to understand and modify disease ontology

In summary, here are the main background questions answered:

- **Why and for whom is it a problem?** Any person who needs to deduce a disease based on a set of symptoms can do so with classifiers such as WebMD, or this smaller, related ontology and classifier.
- **What other attempts have been made to solve this problem, and how?** The main two are:
  - **WebMD**. WebMD likely uses a classifier similar to but much more powerful than the one in this project, as well as a more comprehensive and informational ontology than the one here.
  - **Google's Knowledge Graph**. The second of two main inspirations and motivations of this project, Google's Knowledge Graph retrieves its information from many sources, including the CIA World Factbook, Wikidata, and Wikipedia. [3] Based on results used in research for the ontology in this project, Google's Knowledge Graph likely also uses various medical literature and websites, including high-quality websites, medical professionals, government agencies, medical partnerships, search results, and medical illustrators.[4]

[3] https://googleblog.blogspot.no/2012/05/introducing-knowledge-graph-things-not.html

[4] https://support.google.com/websearch/answer/2364942?p=medical_conditions&visit_id=1-636568519382507184-3737693266&rd=1

3 Methods

3.1 Ontology

The model of concepts, relationships, and axioms that is a conceptualization of the application domain follows:

- The **application domain** in this project is the 26 (originally 50) most common and most commonly searched diseases.
- The **concept** in use in this project is that while some diseases have symptoms in common, given a detailed enough ontology and user information about the symptoms of the disease, the list of possible diseases can be reduced to just one disease.
- The **relationships** in this project are that diseases have symptoms and treatments. Symptoms have categories such as bodily areas (including gastrointestinal, muscular, and skin), while treatments have different types (such as supportive care, medication, and surgery).
- The **axiom** in use is that diseases are unique and have symptoms unique to them, meaning that given enough information (a detailed enough ontology and enough user information about the disease), a list of diseases can be narrowed down to one.

3.2 Problem Solving Methods

The main problem solving method is that of **investigative deduction**. The procedure is straightforward:

1. Acquire knowledge and data (an ontology) about the domain—in this case, diseases and their symptoms.
2. Acquire user input about the disease, such as an initial list and through subsequent questioning.
3. Based on the additional information, eliminate unlikely diseases or diseases that do not match the given symptoms.
4. Repeat steps 2 and 3 until the list of diseases contains one disease.

3.3 Evaluation

To evaluate the project, the four main categories of evaluation were used:

- **Formative evaluation**: The goal of a formative evaluation is to determine how the system can be *better* built. The results of this formative evaluation are mainly listed in the discussion/future works section, but include future utilization of Google Knowledge Graph API, future utilization of OWL API, and ontology refinement.

- **Summative evaluation**: The goal of a summative evaluation is to find out what can be concluded about a system that *has been* built. The results of this summative evaluation are that although the logic of the evaluator/-classifier is almost completely well-formulated, the ontology itself contains a little more than half of those fifty originally intended to be covered.
- **Objectivistic evaluation**: The goal of a objectivistic evaluation is to find out what can be measured and compared about a system—"quantitative," or "hard" research. The result of this objectivistic evaluation is that for the diseases in the current version of the ontology, the deduction process is almost guaranteed (of course, even more testing could be done) to produce the correct result (given accurate user input).
- **Subjectivistic evaluation**: The goal of a subjectivistic evaluation is to find out what can be inferred from observation of a system—"qualitative," or "soft" research. The result of this is that the lack of a pretty graphical user interface and therefore heavy reliance on the use of a Unix terminal makes this inaccessible to most people.

4 Results

As stated earlier, overall, the program performs one main function well:

- Given a small user-inputted list of symptoms, it asks the user more and more questions about the disease's symptoms until the singular disease is deduced.

Figure 1 contains a full sample run of the program.
Originally, the tools planned to use to model the ontology were:

- OWL
- Protégé frames
- Python UML Diagram (possibly)

The tools **originally** planned to use to access the ontology were:

- OWL API
- Python Data Structure

In the end, Python was the main tool used to **model, modify, access, and classify** the ontology. Similarly, the original plan was to use the following problem solving method tools,

- OWL reasoner
- Python program

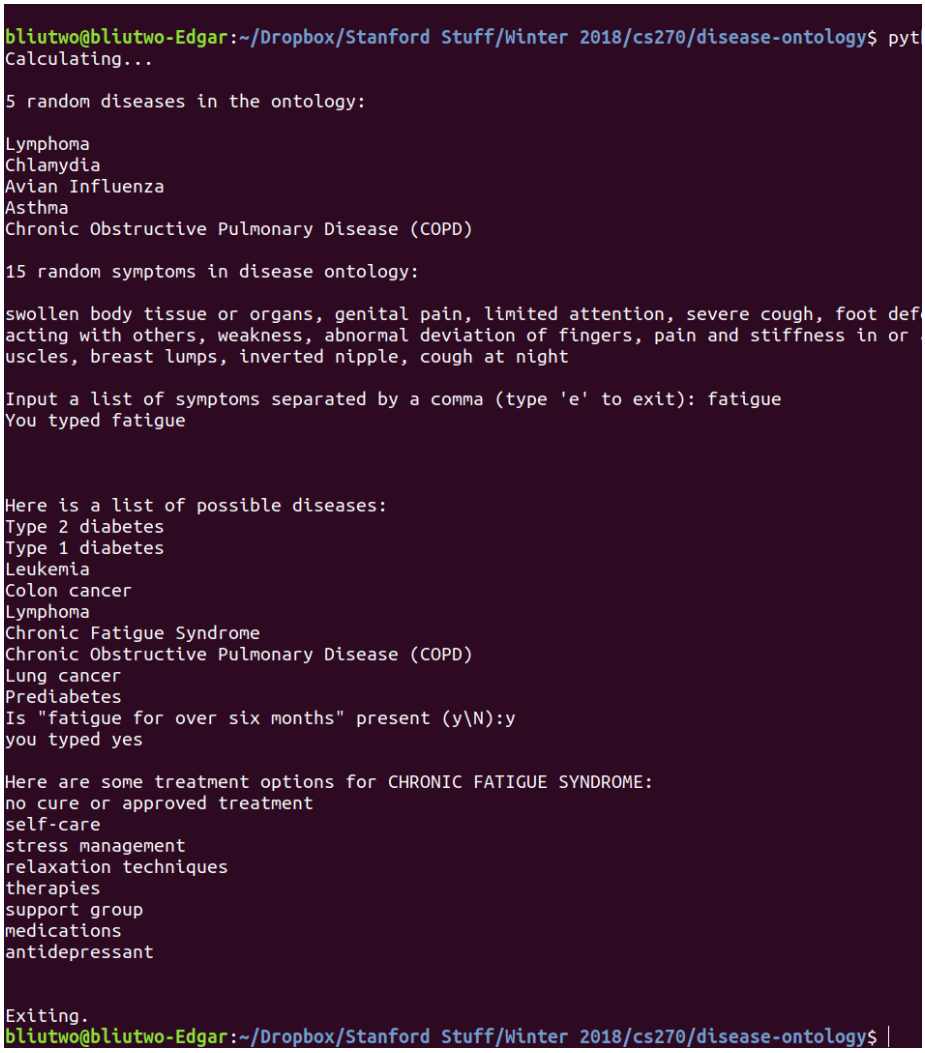

Figure 1: A full run of the program.

but ultimately, Python and its main data abstractions—**lists and dictionaries**—were the main problem solving method tools used.
One bug as of the time of submission is that the classifier may ask the same question as asked earlier—this is simply due to my recursive function not also passing past information into the function, which can be easily fixed with more development time.

5 Discussion/Future Work

There are a main advantage and disadvantage of my approach:

- **Advantage:** Given a well-structured, complete ontology with accurate information, formatted in a specific way, the classifier accurately distinguishes between similar diseases using basic rules of logic and deduction.
- **Disadvantage:** The ontology needs to be those several things: well-structured, complete, accurate, and formatted in a particular way, many of which are not at all guaranteed.

As outlined earlier, overall, the basic functionality and logic are intact, but there are several features that could be added for the most immediate and impactful improvements for the **future**:

- **Call Google's medical knowledge graph API, or Google's Knowledge Graph API[5], to more thoroughly and comprehensively populate the ontology of the most common and most commonly searched diseases.** This is a direct application of the goal to completely finish researching the entire list of the most common and most commonly searched diseases for a more complete and comprehensive ontology.
- **Compile information into an OWL-based ontology and link the current Python interface's functionality with the OWL API.** OWL already has functionality for logic and reasoning on a set of rules, and adding that functionality to the already existing functionality in this project done in Python would add more features and clarity.
- **Add functionality to query about specific diseases.** This feature was originally going to be implemented, but due to time and staff constraints, as of time of submission of the project, the feature is not included.
- **Refine ontology such that synonyms can be accounted for.** One big problem in researching symptoms and treatments of diseases using Google's Knowledge graph is that many of the same symptoms and even treatments were stated in several different ways (e.g. "joint pain" vs. "pain in joints.").

[5] `https://www.google.com/intl/en_us/insidesearch/features/search/knowledge.html`

- **Add more subclasses and structure to symptoms and treatment.** The structure of the actual text file containing the list of diseases with its symptoms and treatments actually differentiates and has subclass types. For example, many forms of treatment for arthritis have classes of "self-care"[6], "therapies"[7], and "medications" [8], whose instances are listed in the corresponding footnotes.

6 References

Most references are the URLs in the footnotes.

References

[1] Centers for Disease Control and Prevention: Diseases and Conditions, Popular Health Topics, `https://www.cdc.gov/diseasesconditions/index.html`

7 Division of Labor

Originally there were supposed to be four project members total, but eventually that number dropped to two, and when *that* project partner stopped responding to texts and emails, the division of labor devolved into the following:

- Brian Liu
  - Handle user interaction
  - Poster
  - Data gathering of different diseases
  - Digital construction of ontology
  - Basically everything

Another way to format it based on the project guidelines:

- Biomedical Researcher/Clinician: **Brian Liu**
- Computer Scientist: **Brian Liu**
- Data Wrangler: **Brian Liu**
- Project Presenter: **Brian Liu**
- Poster Designer: **Brian Liu**

[6] "Self-care" has instances of heating pad, physical exercise, weight loss, tai chi, yoga, cold compress, and ice pack.

[7] "Therapies" has instances of hydrotherapy, stretching, massage, and acupuncture

[8] "Medications" have instances of nonsteroidal non-inflammatory drug, steroid, narcotic, and immunosuppressive drug.

8 Appendix/Program submission

- All files:
  - `https://github.com/bliutwo/disease-ontology`
- Project poster:
  - `https://docs.google.com/presentation/d/1DlCD23tfI4yuineN9ywYr_iEL9K2zVEAfjdFLR6MhRM/edit?usp=sharing`
  - Note: Originally, the pages were to be taped together to form one large poster.
- Five-minute video presentation with demo:
  - `https://youtu.be/o1GThDb0A3g`