Storing Taxonomies in Graphs

Team 1 Membership	Email	UID
Daxton Furniss	Daxton.Furniss@Utah.edu	U1009927
Scott Wardle	U1484669@Utah.edu	U1484669
Tanner Frahm	U1117078@Utah.edu	U1117978
Nathan Losee	Nathan.Losee@Utah.edu	U1094661
Sakshi Singh	U1418696@Utah.edu	U1418696

Project Repository – https://github.com/Nathan-Losee/BMI-6016_Group_1

Background and Motivation

Our project is motivated by an interest in enhancing the comprehension and utility of biological and clinical taxonomies, which are fundamental in various biologically relevant settings. Members of our team have diverse backgrounds, such as ARUP and basic science research laboratories. We are motivated by various experiences with taxonomies like ICD and other biological and clinical EMR codes. We have a collective desire to explore and develop methodologies for preparing these taxonomies for graphical representation.

The complexity and relevance of biological and clinical taxonomies in our respective fields have highlighted the potential benefits of graphical representations, which could significantly improve understanding of biological systems. By focusing on developing a data wrangling preparation method for this taxonomy set, we aim not only to facilitate a deeper understanding of these systems in our current contexts but also to lay a foundation that could be applied to future taxonomy sets. The project seeks to prepare data for a novel graph technology, offering a novel approach to navigating and interpreting the intricate relationships inherent in biomedical taxonomies.

Project Objectives

Our project is centered around the primary objective of transforming taxonomy data, traditionally stored in spreadsheet formats, into a form that is compatible with graphing technologies such as Neo4j. This entails developing a comprehensive understanding of the manipulations required to not only transfer the data into a graphical format but also to accurately represent the hierarchical and categorical relationships inherent within the data.

The learning objectives guiding our efforts include mastering the preparation of datasets for graphical representation, demonstrating the relationships between hierarchical data points, developing a system or pipeline for processing taxonomy data for visual representation, and enhancing our skills in manipulating string and character data for quantitative analysis and comparison.

The potential applications of successfully achieving these objectives are vast and varied. For educational purposes, visualizing taxonomy data could improve the way new trainees in basic science laboratories comprehend the complex relationships between biological organisms. Additionally, the ability to visually communicate hierarchical data could significantly improve the understanding of novel technologies or research results among students, colleagues, and other lab members.

Beyond the immediate educational and communicative benefits, establishing a standard technique for wrangling taxonomy data could have profound implications for visualizing Electronic Medical Records (EMR) classification systems. Such a technique could facilitate the onboarding of new employees, enhance the comparison of hospital systems or units, and ultimately contribute to more effective and efficient healthcare delivery.

Data

We will be collecting data from a terminology database for a subset of the Unified Medical Language System (UMLS), specifically focusing on a SNOMED CT dataset. We will first find and select this dataset using the National Library of Medicine's UMLS Metathesaurus Browser function, and download a .txt or .csv file for a SNOMED CT US taxonomy to utilize for our analysis.

This file will then be used for upload to neo4j or similar graphing technology to display in a graphical format to display the hierarchical relationships.

Links for data sources:

- NLM UMLS Homepage:
  - https://www.nlm.nih.gov/research/umls/index.html
- SNOMED CT Browser:
  - https://browser.ihtsdotools.org/?perspective=full&conceptId1=404684003&edition=MAIN/SNOMEDCT-US/2023-09-01&release=&languages=en

Data Processing

Depending on the dataset(s) that are ultimately chosen for this project, it is likely that we will be working with categorical data. With this type of data, the data cleanup decisions that will need to be made might include what to do about high cardinality. When there are so many unique values under a given data structure, with each occurring infrequently, it makes analysis computationally intensive and challenging to find useful insights for the results. We may need to combine or throw away some of the more sparsely populated categories to make them more manageable. We will surely want to do some frequencies for each of the categories in the data sets.

We may also need to evaluate inconsistent categorization for some of the data sets, especially if there is a need to merge two or more datasets together across a primary or foreign key. If there are misspellings or inconsistent capitalization in some categories, that would have to be corrected first so we can standardize the categories.

As we will be creating hierarchical tree structures with this data, there are a number of considerations that will need to be addressed to make the structure clean and well-organized:

- The categories with significant observational quantities will need to be represented on the structure. Others without significant quantities should be eliminated or combined so the graphs do not appear too cluttered.
- With limited space on the trees to represent the categories due to the size of the bubbles or boxes that will be used, we may need to shorten category names or use codes with a corollary table lookup to help the user understand the structure and what it represents.

Design

The project dictates that we should be able to graphically show the hierarchical relationships between various SNOMED codes. As such, we will use graphing software to display our data. It will look similar to a typical family tree, with the highest term and code being at the top and having the “children” fields and specific codes at the subsequent levels.

We will most likely subset our data with one main field, such as ‘Disease,’ and then show its relationships to other SNOMED codes and their relationships to other codes, the most common relationship being ‘as of,’ but other types will also be displayed. An alternative prototype would be a circle diagram with ‘Disease,’ and its SNOMED code in the middle and having subsequent circle layers expand out with the more specific codes and terms to the lowest level we have in our dataset.

The relationships between the terms could potentially be made in different colors to be more specific with how the terms relate to each other. The codes and their relationships can be modeled in many ways, but we will want the highest, most general term at the top or center as a skeleton. All terms and codes stemming from it either surrounding or below with lines or connections either attached or color-coded. This will allow us to satisfy the requirements of showing the hierarchy of these terms and how they relate.

Must-Have Features

- **Flexible Data Importation:** We should be able to support various data formats for importing taxonomy data, including .txt and .csv, directly from sources like the UMLS Metathesaurus browser
- **Data Cleaning Automation:** We should have some sort of built-in functionality to automatically (or semi-automatically) detect and correct common data issues.
    - Misspelling and duplicate entry correction
    - Normalization (capitalization, punctuation, etc.)
- **Hierarchical Visualization:** Overall, we should be able to convert data to an understandable graphical format through graphing software. It should be able to run without errors that inhibit any output
- **Scalability:** The data wrangling method should be able to scale up and handle larger datasets than our small, class project example
- **Customizable Graph Layouts:** The data should be wrangled in a manner that it can be adapted to more than one graphical layout
- **Annotation and Notes:** The data cleaning code should include clear comments that allows future potential users to identify why code was selected and how it alters the dataset.

Optional Features

- **Performance Optimization:** Optimize the performance and quantify the speed and time to complete the operations
- **Quality Assurance Testing:** It would be nice to develop some method to test our initial data against the produced graphical representation, perhaps by compiling existing taxonomy representations.
- **Data Enrichment:** Ideally, we could link smaller taxonomies already graphed and cleaned to larger ones and speed up the cleaning by utilizing similar contained fields.

Project Schedule

- **Feb 5th - Feb 12th:** All group members are to fill out their assigned sections; Nathan to upload the proposal document to GitHub and submit
- **Feb 12th - Feb 19th:** Everyone meets with class instructors to receive feedback on the proposed project design. Make adjustments as suggested. Team 1 will collectively review the data and do preliminary wrangling and analysis to ensure we have enough good, quality data to begin our project as soon as it's green-lit.
- **Feb 19th - Feb 26th:** Nathan will load .txt data files into CSV files and will send them to Scott, Daxton, Tanner, and Sakshi for preliminary Data Wrangling and Data Analysis to ensure data quality.
- **Feb 26th - Mar 4th:** Team 1 will meet and discuss findings from preliminary data quality work. They will update the project details per the feedback received earlier and findings from data wrangling. Nathan will upload the document to GitHub and submit an assignment to Canvas on March 1st.
- **March 4th - March 11th:** Team 1 will review the feedback and update the project plans as needed. Nathan will take the updates, post them to GitHub, and submit them.
- **March 11th - March 18th:** Sakshi and Scott to prepare and peer review the intermediate work presentation for presentation. The presentation is to be given on the 18th by Team 1 collectively.
- **March 18th - March 25th:** Preston and Daxton will update the project scope and presentation. Once completed, Nathan will peer review the documents, upload them to GitHub, and submit the assignment.
- **March 25th - April 1st:** Team 1 will meet to discuss the final steps needed for project completion. Any updates will be made.
- **April 1st - April 8th:** Team 1 will meet with Instructors for further feedback and instruction.
- **April 8th - April 15th:** Team 1 will meet and discuss the feedback. Make plans for Sakshi and Daxton to update the paper with Nathan and Scott for peer review.
- **April 15th - April 22nd:** Team 1 members will continue the project until its completion. Create plans and run through how the final presentation will go.
- **April 22nd - April 29th:** Team 1 will prepare for and give their final presentation on April 29th.
- **April 29th - May 6th:** Team 1 will review any feedback given from their final presentation. Scott, Daxton, and Preston will tweak anything that's left. Sakshi and Nathan will peer review, and Nathan will do the final upload to GitHub and submit the final project on or before May 6th.