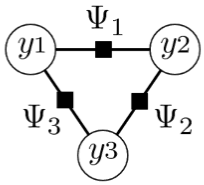
Open Gov
July 25th 2017

Documentation of Natural Language Processing

My goal for my portion of the assignment for our open-gov html parser was to create a universal parser that would work on almost every type of website and take that data and create uniform tables which could easily be queried by users of Open-Gov. Unfortunately creating a universal parser turned out to be much more difficult than was previously thought, and it revealed itself to be a project that would take much longer than a few weeks. However the task ended up being a great learning experience and also clearly illustrated the importance of adhering to the scrum guidelines. The essence of what I was trying to do was take any html code, use the Directed Object Model to turn the html into a collection of objects and then use a machine learning concept known as conditional random fields that would allow me to parse any set of objects. The way conditional random fields allow universal html parsing is by taking each html object and considering it in the context of the objects which surround it.

$$p(\mathbf{y}) = \frac{1}{Z} \prod_{a=1}^{A} \Psi_a(\mathbf{y}_a),$$

What the above equation does is it takes the entire set of objects and from size 1 to the size of the set of objects and creates every possible combination of the objects as long as the objects are connected nodes in the graph. In this manner every single possible combination of objects is analyzed, and the probability of a portion of the html code being a table can be determined, and if this meets the threshold of 70% likelihood, this threshold was decided to account for errors in the parsing while also preventing a large number of false positives.




An example of a Directed Object Model with factors phi describing the relations of the nodes of the graph

And if this set of objects has been determined to be a table, the objects are then passed into our database for easy querying. However, the most challenging part to implement of the natural language processor is the initial training of the algorithm, where the algorithm is initially trained by a human to be able to recognize whether something is a table or not. This involves thousands of analyzed sets of html objects, and the human telling the algorithm whether something is a table or not, and the algorithm changing its factors phi to as these results are given to the algorithm, over time fine tuning the algorithm for use in general cases.

My hope is that a future group in the software engineering class will be able to pick up the reigns where my group left off and possibly create a universal parser that can scrape numerous government websites and aggregate the information in an easily accessed database that will allow for queries from a specific website, and extend that to allow for searches for particular types of information such as gdp per capita for a particular year and receive numerous results from many government sites easily.