Project Documentation

Overview

I created a stand-alone crawler tool that takes a university website from the user and automatically crawls to gather directory and faculty webpages. The overview of the steps performed are:

- Utilize train and test data to train the classification model
- Start crawling for directory and faculty URLs

Implementation

The preprocessing of the data is done in the model.py file. Looking through the list of valid URLs, I noticed that words such as directory, faculty, faculty-staff, faculty-directory, people, people-page, staff, teaching-faculty, about-people, members, all-faculty-staff, faculty-and-staff, faculty-list, professors, and faculty-profiles were common. Therefore, I used these words as hints in deciding whether a given URL is a valid URL or not. I also decomposed all the URLs into parts where the individual parts can be analyzed to identify the URLs correctly. I utilized the Support Vector Classifier for classification. I used both positive and negative examples for training and testing purposes.

Usage

To begin with, install Chrome web driver so that the dynamic JavaScript content can be crawled using selenium.
Then, the following libraries need to be installed:

- Selenium
- NumPy
- Sklearn
- Bs4 (beautifulsoup)

Then follow the following steps:

- Open the main.ipynb file using jupyter notebook
- Insert the university URL that you want to start crawling
- Run the file to start the crawler
- View the results in directory_urls.txt and faculty_urls.txt