Final Project Report

1. Function Overview

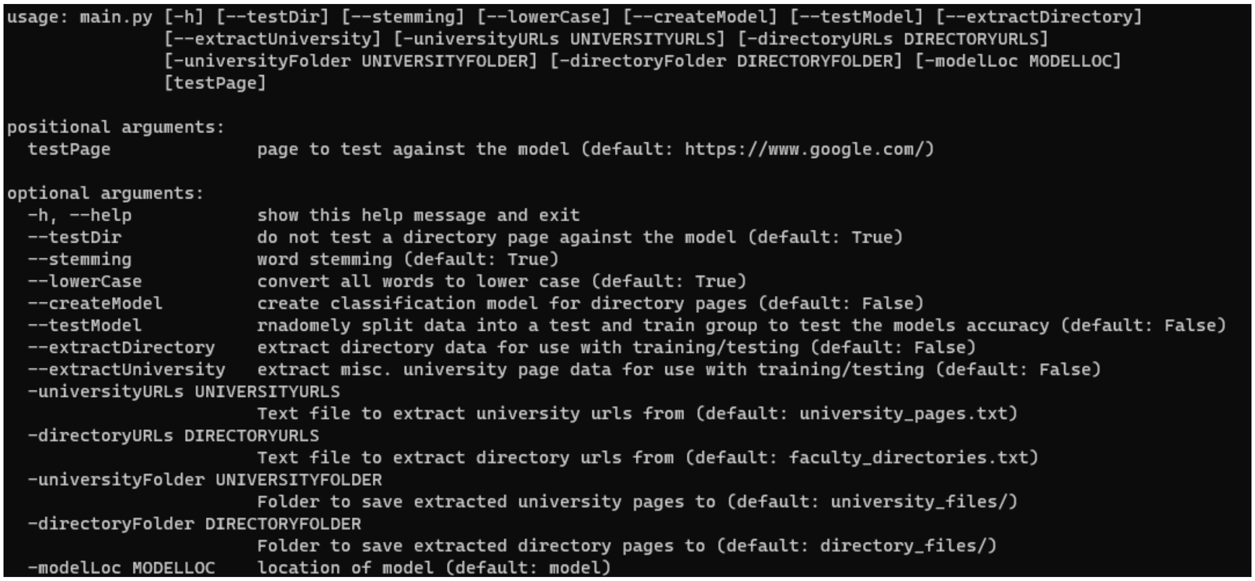
Directory page classifier with cli to control preparing the data for the model, creating the model, and testing against the model.

2. Implementation Detail

Built in Python using the bs4, requests, and nltk outside libraries. First to prepare data for the model, webpages are pulled from a list of directory and non-directory webpages. The html text content from these websites is then cleaned, scrapped, and saved in a directory for use later. The text content in these files is then tokenized and modified using the nltk library to be fit to create a model with. The modified terms are then passed through to create a bigram bag of words model, which is then saved to a file for use later classifying directory and non-directory pages. With the model created, a webpage can be entered into the cli, which will make a determination based on the model whether it thinks it is a directory or non-directory page.

3. Usage Documentation

All function of the program occurs through the cli. For the program to work python3 and its standard libraries are required alongside bs4, requests, and nltk. To run the program, the user should navigate to the repository and type `python3 main.py`. This will run the program at all of its default settings, utilizing the pre-built model and checking it against http://www.google.com, so the program will return ‘Classified as NOT a Directory Page’. If the program is run against a directory page such as `python3 main.py https://cs.illinois.edu/about/people/all-faculty` it should return ‘Classified as a Directory Page’. To view the other options the program has, the user can type in ` python3 main.py -h` which will return:



```
usage: main.py [-h] [--testDir] [--stemming] [--lowerCase] [--createModel] [--testModel] [--extractDirectory]
               [--extractUniversity] [-universityURLs UNIVERSITYURLS] [-directoryURLs DIRECTORYURLS]
               [-universityFolder UNIVERSITYFOLDER] [-directoryFolder DIRECTORYFOLDER] [-modelLoc MODELLOC]
               [testPage]

positional arguments:
  testPage              page to test against the model (default: https://www.google.com/)

optional arguments:
  -h, --help            show this help message and exit
  --testDir             do not test a directory page against the model (default: True)
  --stemming            word stemming (default: True)
  --lowerCase           convert all words to lower case (default: True)
  --createModel         create classification model for directory pages (default: False)
  --testModel           rnadomely split data into a test and train group to test the models accuracy (default: False)
  --extractDirectory    extract directory data for use with training/testing (default: False)
  --extractUniversity   extract misc. university page data for use with training/testing (default: False)
  -universityURLs UNIVERSITYURLS
                        Text file to extract university urls from (default: university_pages.txt)
  -directoryURLs DIRECTORYURLS
                        Text file to extract directory urls from (default: faculty_directories.txt)
  -universityFolder UNIVERSITYFOLDER
                        Folder to save extracted university pages to (default: university_files/)
  -directoryFolder DIRECTORYFOLDER
                        Folder to save extracted directory pages to (default: directory_files/)
  -modelLoc MODELLOC    location of model (default: model)
```

This shows all options of the program including those related to extracting the initial data and creating the model.

4. Video Overview

https://www.youtube.com/watch?v=KabKW-5aXY8