**Question 4**

In order to run this file, just set up a maven project with Tika and OpenNLP NER dependencies. Download all the NER bin files and change the location accordingly in the program. Also make changes to the path containing the CCA files.

The output of the program is a json file containing the request URL, request content-type, response URL, response content type and the Named Entities and types of Named Entities (Location, Date, organization etc.) found in the page.

**Question 7**

In order to run this file, just set up a maven project with Tika dependencies. Also make changes to the path containing the CCA files.

The output of the program is a CSV file containing the language found for different content types and also the number of document for that language and content type.

**Question 8:**

- Get ner.json file from Assignment 2 result

- Run json\_to\_csv.py to generate count\_words.csv

- Use WorldCloud.html and count\_words.csv to view WorldCloud d3 graph

**Question 9:**

- Change “from” variable in java/Test.java to data folder path

- Run java/Test.java and get output files

- Run python/max\_joint\_agreement.py on output files to produce max\_joint.json

- Use NER.html and max\_joint.json to view NER maximum joint agreement

**Question 10:**

- Modify paths in java/ParserExtraction.java to the correct data folder paths. (Follow comments in file)

- Run java/ParserExtraction.java and get output files

- Run python/analysis\_measurements.py to generate output tsv files

- Use Spectrum and tsv files to view Spectrum d3 graphs