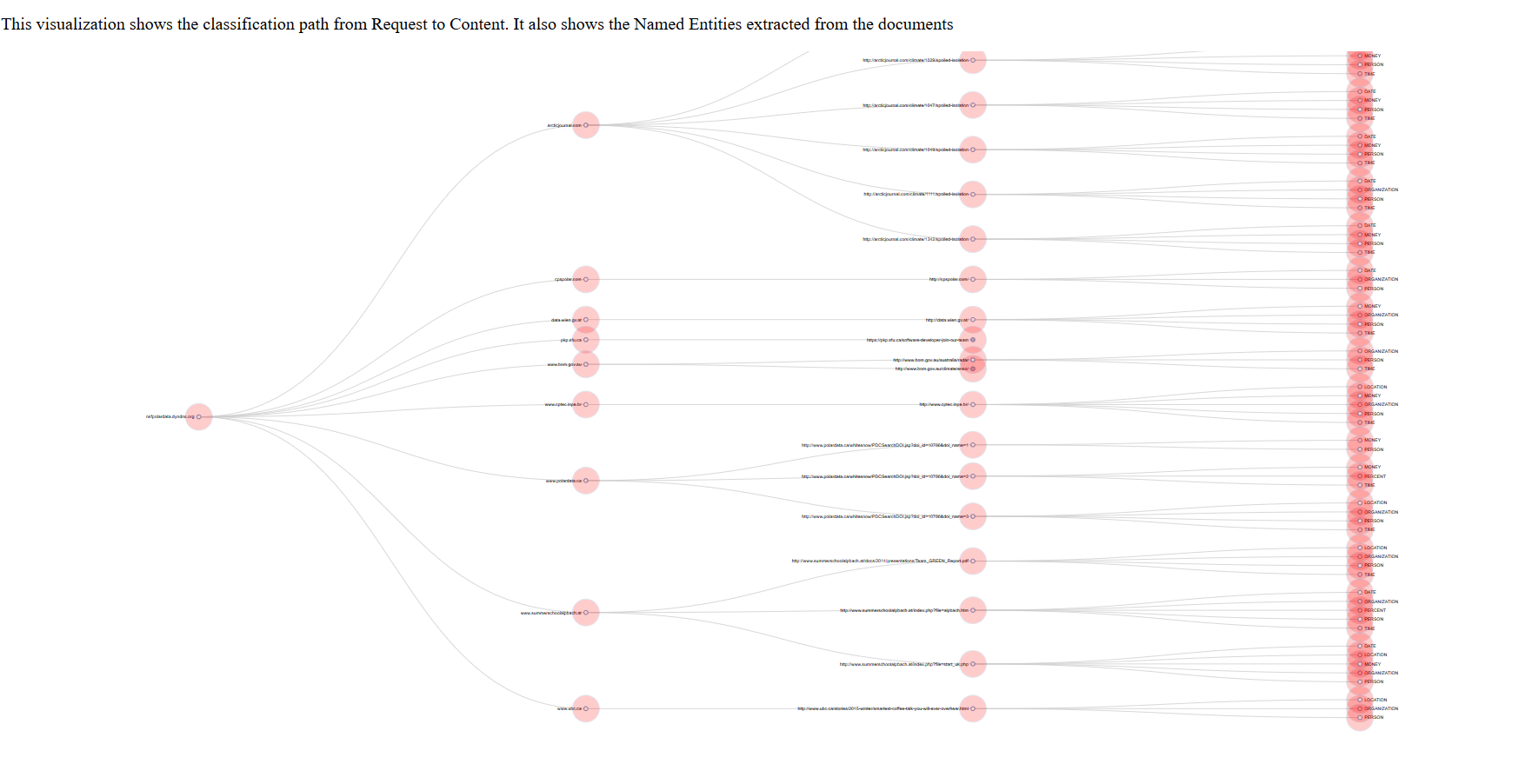
**ReadMe – Assignment 3**

**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.

**D3 –**

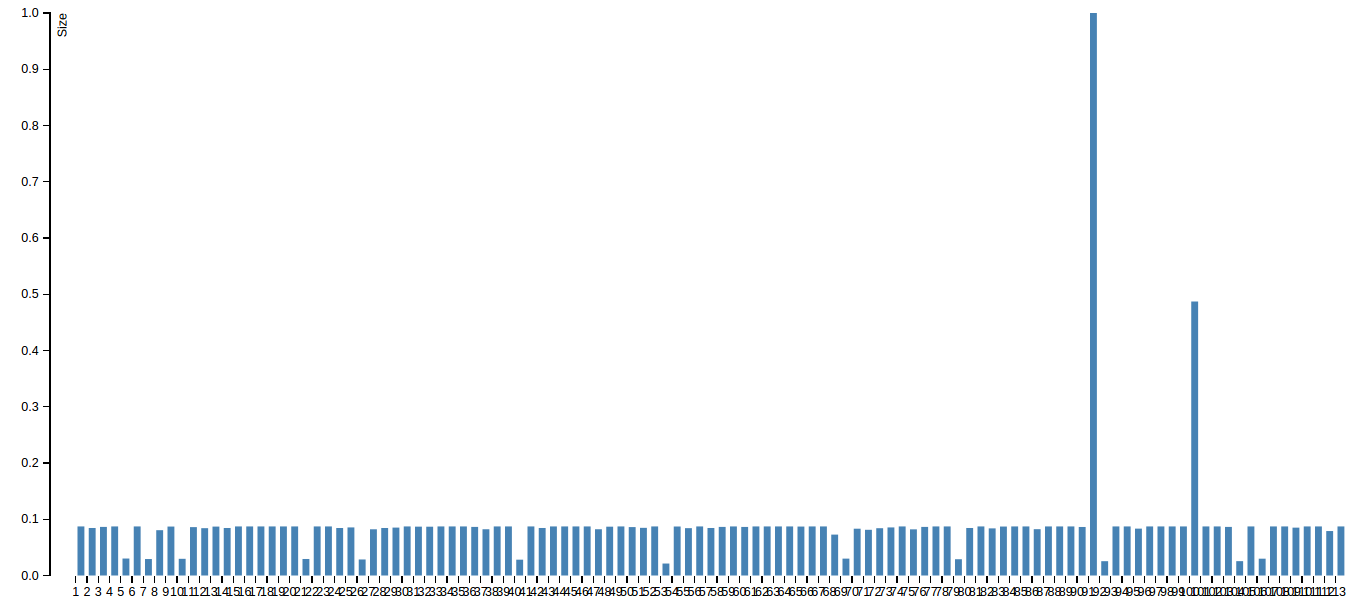


The parent node is the host from where the crawler starts. The next nodes are the request URLS from where the crawler makes the request. Each request URL has many response URLS which contain the types of Named Entities that were extracted from the files.

**Question 5**

There is a script for calculating file size and one script to normalize it after taking ratio with solr index. They are cal\_filesize\_diversity.py and normalize.py. The folder also contains data files for 6 MIME types.

**D3 -**

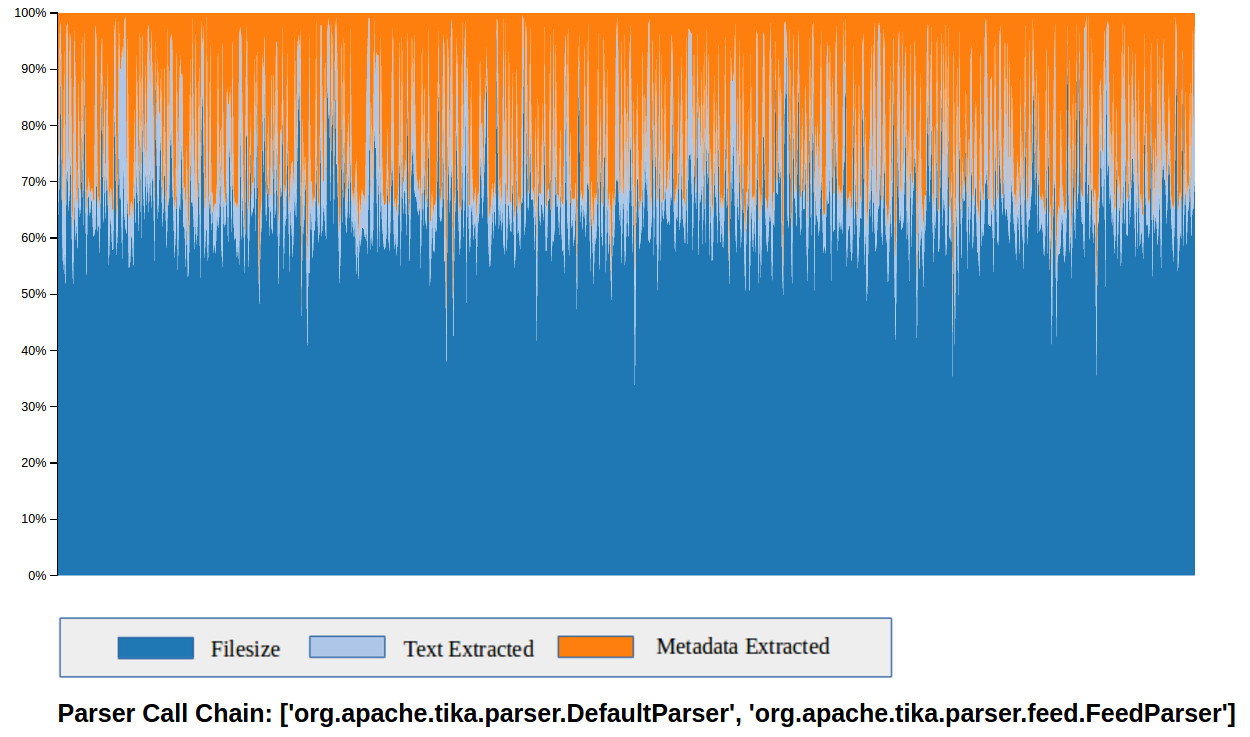


The d3 shows the ratio of the solar index to the original file size diversity for each MIME type.

**Question 6**

Extract\_txt\_met.py extracts relevant fields for further steps like parser call chain, size of extracted text and size of metadata extracted. The folder contains html files for d3 along with the data files.

**D3 -**



This d3 shows the relative percentages of actual file size, size of extracted text and size of metadata as an area chart. You can find all d3 visualization for each parser call chain and 6 MIME types in our submission folder file\_diversity and parser\_call\_chain.

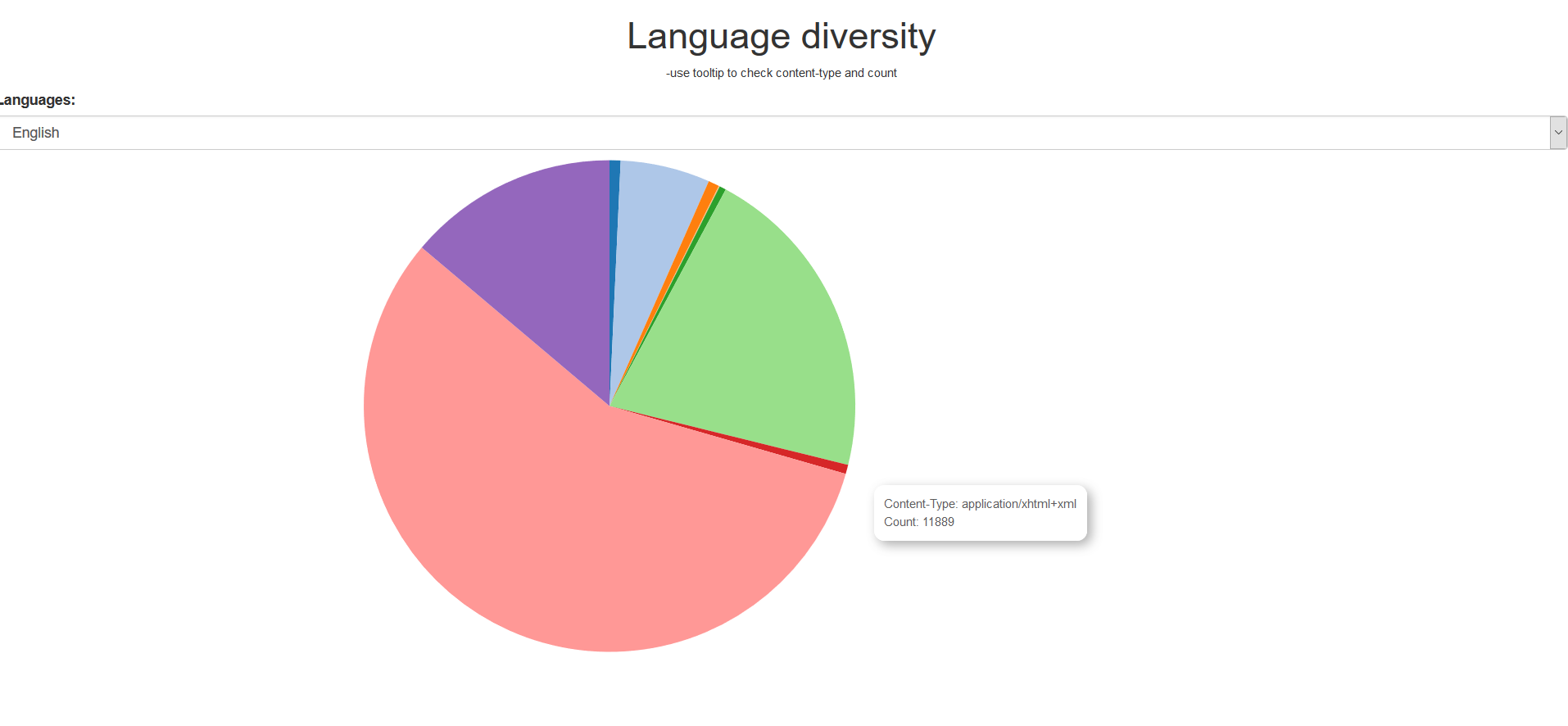
This d3 contains the plot for both amount to text retrieved per MIME type and amount of Metadata retrieved per MIME type

**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.

**D3 -**



This d3 shows the pie chart for each of the 25 languages detected. The pie chart represents the content type for which the particular language was detected and the number of files for that content type containing the selected language.

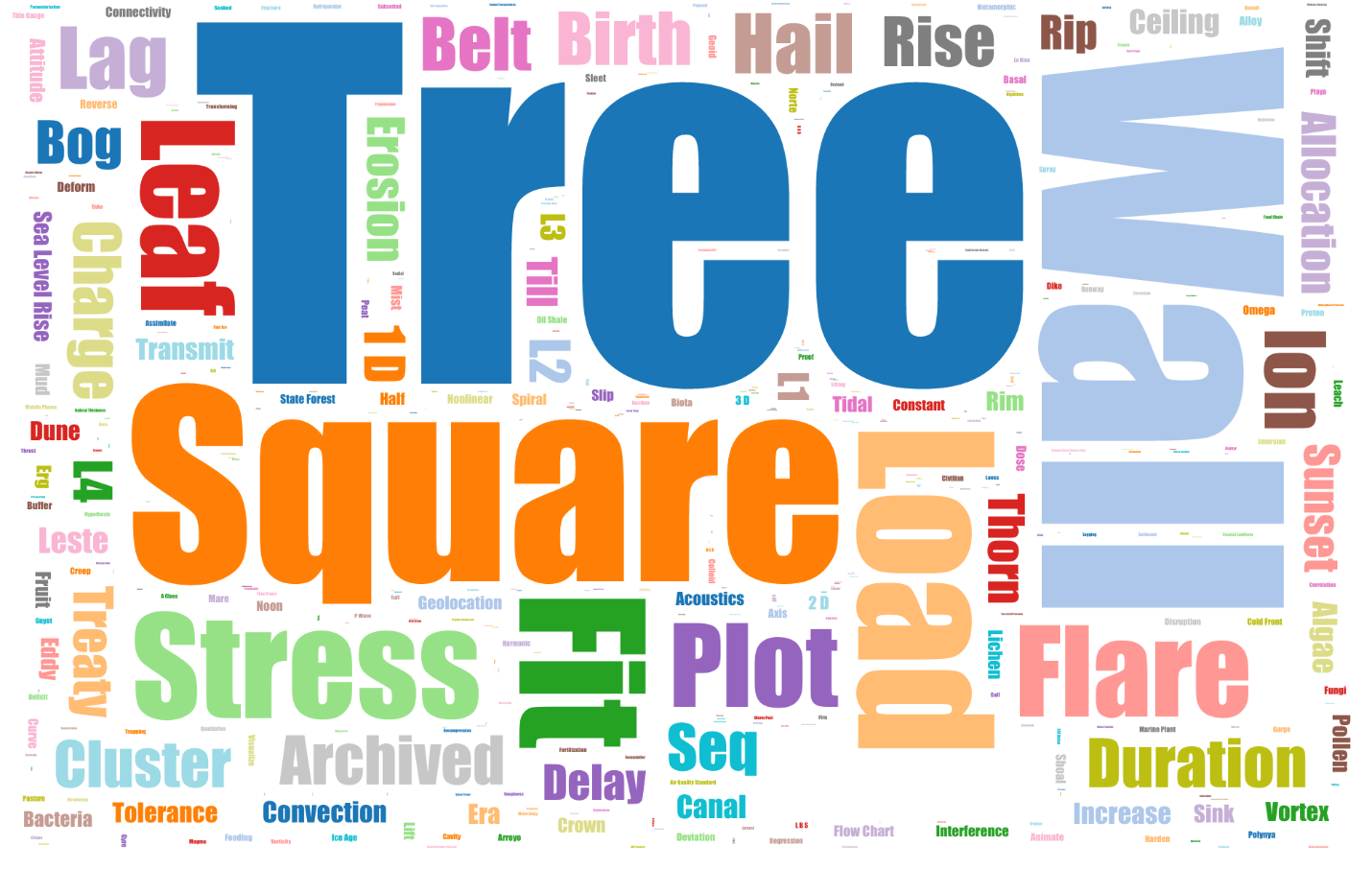
**Question 8:**

- 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

**D3 –**



The d3 shows the world cloud containing the metadata extracted from the SWEET ontology and the measurements from Assignment 2.

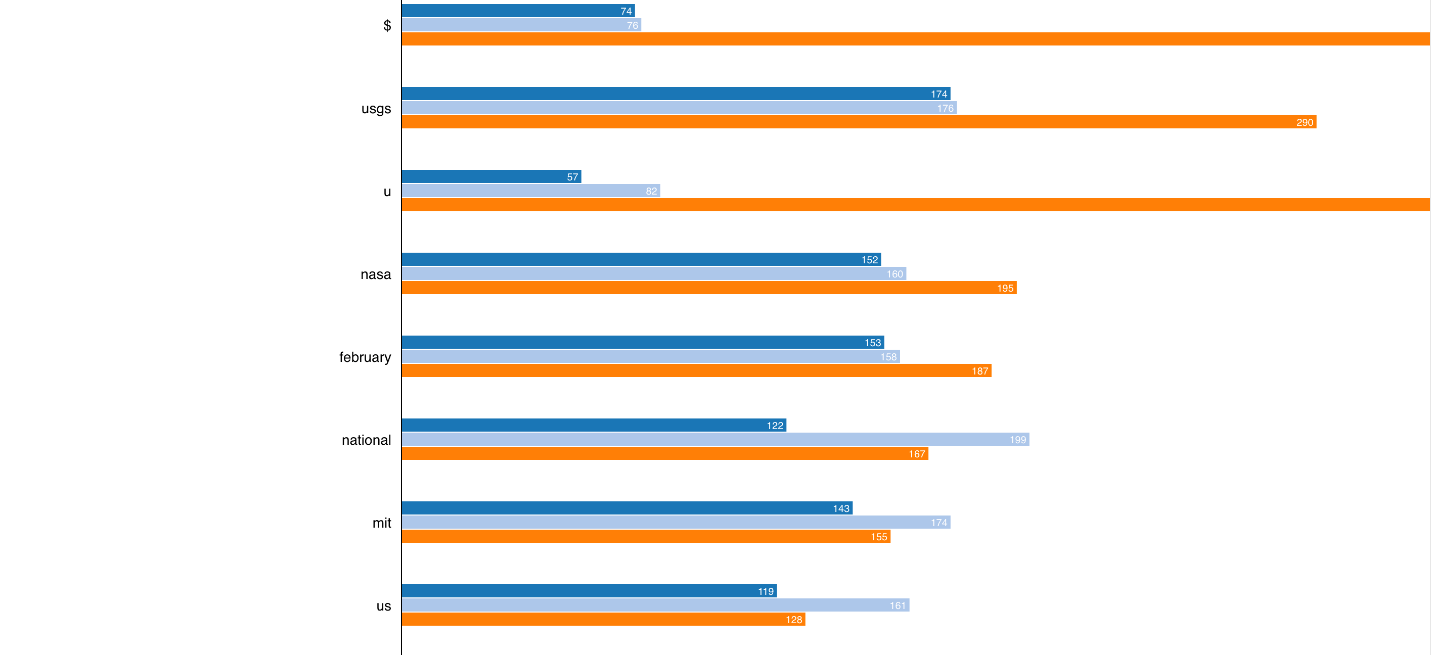
**Question 9:**

- 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

**D3 -** 

This d3 shows the frequency of occurrence of terms in OpenNLP, CoreNLP and NLTK for the maximal joint agreement.

**Question 10:**

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

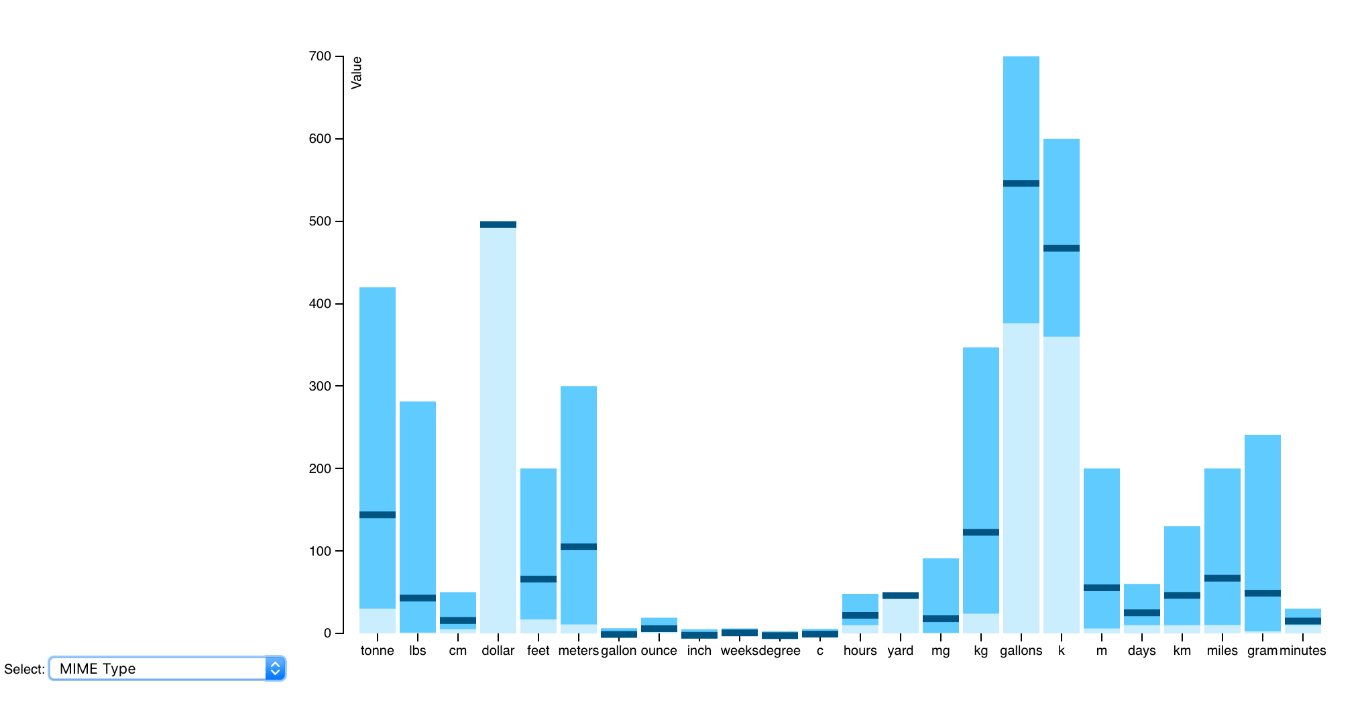
- Run java/ParserExtraction.java and get output files

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

- Use Spectrum and tsv files to view Spectrum d3 graphs

- We have included the screenshots of the visualizations required for the spectrum of measurements as .png files in the folder.

**D3 –**



This shows the graph for the spectrum of measurements. The dark blue band in the middle represents the average.