**Tokenization Process**

**Author**: Timothy Naudet

**End Result**:

A json file containing the error ID, the text, the tokens produces by teqhcniques involving spacy, nltk, bagOwords, Doc2Vec, keras, and stokenizer, as well as the percentages that each tokenizer overlaps with the spacy tokens.

**List of executable scripts:**

1.make\_github\_tokens.py

2.compare\_token\_metrics.py

3.plot\_token\_counts.py

4.tfidf.py

**Approach**:

The process includes called 1.make\_github\_tokens.py to access the “spack-monitor-nlp” repository, use the errors-\*.json raw urls, and collect the text from each entry in the files. This script then returns two json files named “\_stoken\_\*\_tokens.json”. An import anecdote is that regex expressions were implemented to capture the file paths provided in the error messages and stored within the tokenization component of each json entry. These files are then used in 2.compare\_token\_metrics.py.

2.compare\_token\_metrics.py uses the glob package to open each of the aforementioned stoken json files and calculate the percentage of overlap of each tokenization technique with the spacy tokenization technique. This percentage is a basic comparison:

The metric is calculated as a percentage of the number of

non-spacy tokens in the non-spacy technique:

((number\_spacy\_tokens – number\_not\_in\_spacy)/number\_spacy\_tokens) \* 100

These metrics are the provide for each non-spacy tokenization technique in the singular outbound json file.

3.plot\_token\_counts plots two seaborn bar-charts in one image. These list the 30 most common tokens for all tokens as well as the 30 most common file paths (separated using regex).

4.tfidf prepare a few deeper analyses of the “error-?.json” text files.