Sabarish Chockalingam

MSiA 414 – Text Analytics HW1

Package Text Preprocessing Analysis

NLTK (Natural Language Toolkit) and spacy are popular libraries for Natural Language Processing (NLP). NLTK was created in 2001 and spacy was created more recently in 2015. NLTK has a wide variety of algorithms to use, whereas spacy focuses on one (supposedly the best). Spacy may be more user friendly in some cases as it seems to be specialized for the industry and more tailored to industry users. Below is discussion of results and thoughts after I tested the text preprocessing features from the two libraries for myself.

The default tokenizing in NLTK seemed to be more accurate. Spacy also included whitespace in the tokenization. This lead to the text file with all of spacy's tokens to be larger than that of nltk. This is could likely be fixed by creating a custom tokenizer in spacy, but that is a difficult task for beginners. Stemming with nltk seemed inaccurate. For example, resource was changed to 'resourc' and foundation was changed to 'foundat'.

NLTK stemming seems to use rudimentary rules of only removing suffixes such as 'ed' and 'ion' and does not have a higher intelligent procedure (using lemmatization may be more accurate). However, this method did work correctly for quite a bit of words. For example, contradiction was changed correctly to contradict. spacy does not contain built in stemming function as it relies only on lemmatization. NLTK has had more time to gain features and stemming may not be as prevalent in the present. Spacy developers may have deemed it unnecessary to add a stemming feature.

POS tagging in NLTK works surprisingly well. It can identify names (such as Darwin) as proper nouns. 'written' was also correctly identified as a past participle. A more complete run through is needed to assess actual accuracy (all words have to be classified manually first for comparison) but from a superficial overview, tagging works quite well. It is also worth noting, some tags maybe easier to categorize than others such as CC - coordinating conjunction (and, or, etc.) and IN – preposition (for, by, etc.). POS tagging performed similarly with spacy. The tags were however easier to understand as they were fully written out (i.e. NOUN, PROPN, ADVERB, SYM (symbol), etc.)

Parallelization was more effective with NLTK. There maybe more processes than tokenizing and POS tagging occurring under the hood when using spacy (it is difficult to split the two task as it occurs as a built-in pipeline). This likely causes it to take more time than NLTK. Parallelizing only saves about 2 min with Spacy but saves about 7 min with NLTK for this newsgroup corpus.

Timing was similar for tokenizing and pos tagging for NLTK and spacy. Parallelizing was most effective with NLTK and had little return with spacy.

	NLTK	Spacy
Tokenizing Time	0h:01m:32.850105s	0h:01m:32.047774s
Stemming Time	0h:02m:39.883217s	No stemmer
POS Tagging Time	0h:09m:09.057957s	0h:09m:09.283922s
Parallelized Tokenizing and POS	0h:04m:13.825640s	0h:08m:29.550801s

*spacy had a character limit for tokenizing. Therefore tokenizing had to be done for each corpus file individually. NLTK did not require this extra step.