Text Analytics HW1

1. nltk vs. spacy

In this homework, I’ve used nltk and spacy for text analytics. We could apply tokenization, stemming and POS tagging on the full corpus using nltk, but spacy utilizes lemmatization to look for the root of each word. Comparing running time, I found the performance of nltk is similar to spacy in terms of word tokenization and POS tagging, but nltk outperformed in sentence tokenization. The performance was not improved that much for spacy and even slower for nltk in parallelization. And I’ve noticed that spacy has built-in function of multithreading.

Running time in seconds

	NLTK	Spacy
1. Word Tokenization	2.68	2.81
2. Sentence Tokenization	0.95	2.83
3. Stemming	4.68	N/A
4. POS Tagging	12.35	12.38
Parallelization (1+2+4)	44.2	14.25

In production, spacy is the fastest NLP framework, however in my case, nltk and spacy are similar in performance, maybe because I only implemented on relatively small dataset due to limited memory. Spacy only supports 7 languages, whereas nltk is much more applicable.

Please see source code here:
https://github.com/MSIA/zzm7646_msia_text_analytics_2020/blob/hw1/HW1.py

2. Regular expression

Please see source code and result here:
https://github.com/MSIA/zzm7646_msia_text_analytics_2020/blob/hw1/HW1.html