Assignment 6 (Draft)

due 12/3/2020

Like some earlier assignments, this assignment revisits the theme of analyzing text, but this time we will use a Python package to do most of the heavy lifting. Package `nltk` provides a range of capabilities for working with natural-language text, for example tools to decompose text into its constituent parts (words or sentences) and so on. The objective here is not only to produce code that accomplished the task set out below, but to use this as an exercise to familiarize ourselves with `nltk`.

The objective is to write a function named `analyze(book_name)` based on `nltk` that takes the name of a novel (in our case a particular novel accessible though `nltk`), to analyze it and print a summary akin to the following:

```
Analysis of ‘austen-sense.txt’
# chars = 673022
# words = 141576
# sentences = 4999
Longest word = ‘disinterestedness’
Longest sentence = ‘I am sure. . .Marianne ,’ (303 words)
Vocab size =  4089
Largest stem family ’respect’:
    {’respectability’, ’respects’, ’respected’, ’respecting’, ’
    respectful’, ’respect’, ’respective’, ’respectably’,
    ’respectfully’, ’respectable’}
```

The output should include the following information gleaned from the novel:

1. the number of characters, words and sentences [1];
2. a longest word (or one of same if there is a tie);
3. a summary of a longest sentence (by word length) to include the first and last few words of the sentence;
4. the size of the vocabulary (detailed below) and
5. the largest stem family (detailed below) appearing in the novel.

When calculating the vocabulary size, we seek the number of *distinct* words. For this calculation we consider only words (ignoring capitalization) and not numbers and punctuation marks and so on. Moreover however many times a word appears in the test, it contributes at most one to the vocabulary count. Furthermore we conflate all words sharing the same stem i.e treat them as the being the same when counting vocabulary. By stem we mean the core/root of the word that it shares with closely related words. For example, singular and

[1] In `nltk`-speak the concept of “word” (more accurately “token”) encompasses not just English-language words but also numbers, punctuation marks and so on. Also its concept “sentence” is a little loose. For the purpose of counting word and sentences here, we will accept `nltk`’s word- and sentence-decomposition logic. For the vocabulary count later on we will consider only English-language words.

plural nouns “cat”, “cats” have the same stem “cat” and related forms of the verb “laugh”, such as “laugh”, “laughs”, “laughed”, “laughing” and so on all share the stem “laugh”. The `nltk` package contains stemming machinery. See below.

. . . .

Notes

1. The `www.nltk.org` webpage contains a wealth of information about the `nltk` package. Skim this to get a sense of its capabilities, but don’t get bogged down. The art here is to find the stuff you need, while glossing over the stuff you don’t. There is an online version of a book linked to on the `nltk` home page that may be useful.

2. The package incorporates several corpora (selections of text), including the material drawn from the Gutenberg corpus `nltk.corpus.gutenberg` that contains, among others, the text of Jane Austen’s *Sense and Sensibility* in a file named `‘austen-sense.txt’`. Poke around the `nltk` site to figure out how to get access to the raw text and the words and sentences contained therein.

3. There are several stemmers available in `nltk`, but we will use the Porter stemmer. This is available as `nltk.stem.porter`.