Due Monday, March 11, 2019

Project 1 - Cleaning and Tokenization

Level 1

Write a program that can clean a file and count the words in it. It should run using the following command in the terminal:

./clean_and_count_tokens.py <input_file> <output_file>

The input file to clean is in xml, which contain tags encased in <>. These tags should not be counted. All other words are counted.

A word may only contain the following:

- Capital letters
- Lowercase letters
- the straight apostrophe '
- internal periods (with alphabetic characters on either side)

Words should count towards the tally regardless of capitalization. Ex: 'Is' and 'is' should both count as instances of 'is'

Write the results to a file, one word per line with the count IN ORDER from most common to least common. Ties should be in alphabetical order. There should be a single tab between the word and the count.

A sample input/output is included in the folder. Please take a look!

Please run your code on the included Wikipedia-LexicalAnalysis.xml, and call the output file lexical_analysis_out.txt

You may only import: sys, re (or regex)

Your submission should include the following 2 files:

1. clean_and_count_tokens.py
2. lexical_analysis_out.txt

Depending on how you organize your code, you may have more files than this.

Due Monday, March 11, 2019

Level 2

In addition to the above file, write a program that uses nltk's word_tokenizer and porter stemmer after cleaning out the tags, but before counting the tokens. This program should run using the following command in the terminal:

./nltk_clean_and_count_stems.py <input_file> <output_file>

For more information about NLTK's tokenizing/stemming, check out Chapter 3 of the NLTK book: http://www.nltk.org/book/ch03.html

Check out sample_stemmed_out.txt for an example output.

You may import: sys, re (or regex), nltk (for this file only)

Your submission should include:

1. clean_and_count_tokens.py
2. nltk_clean_and_count_stems.py
3. lexical_analysis_out.txt
4. lexical_analysis_nltk_stemmed_out.txt

Depending on how you organize your code, you may have more files than this.

Level 3

In addition to both of the above files, write your own Porter Stemmer. See https://tartarus.org/martin/PorterStemmer/def.txt for the original paper that describes the algorithm in detail. Your program should remove the tags, tokenize the text, and run it through your porter stemmer before counting the tokens. It should run using the following command in the terminal:

./my_clean_and_count_stems.py <input_file> <output_file>

You may import: sys, re (or regex). You may not use NLTK for any of these steps.

Your full submission should include:

1. clean_and_count_tokens.py
2. nltk_clean_and_count_stems.py
3. my_clean_and_count_stems.py
4. lexical_analysis_out.txt
5. lexical_analysis_nltk_stemmed_out.txt
6. lexical_analysis_stemmed_out.txt

Depending on how you organize your code, you may have more files than this.