Computational Linguistics
Fall 2018

Practical 5

This is a graded individual homework. Discussion is encouraged within and outside the class, but **please write down your solution yourself. Copying code or letting others copy your code will result in both assignments being discarded at the very least.**

Open the file 'Lovers_on_Aran_messed.txt'. You'll find that the poem is messed up (by the previous TA) with many unwanted space/tab characters, and "empty" lines which contain no words (but there might be some invisible space/tab/newline characters!!!). Write code to:

(1) **(20%)**Read in the file. Remove all "empty" lines. Remove all superfluous space/tab characters. Add a space character **between a punctuation mark and surrounding words**, but **do not** leave any space character after a punctuation mark if it's at the end of a line. For example:
'Came glinting, sifting from the Americans.'
–> 'Came glinting , sifting from the Americans .'
Write the cleaned poem to a new file called 'Lovers_on_Aran.txt'. Make sure that there is a newline character (\n) at the end of **each line**.

(2) **(20%)**Read in the cleaned file you just created. Count the number of words (including punctuations) of the file. Try to find every word/**punctuation mark** which either begins with a capital letter, or contains fewer than 3 characters.
Write the result to a new file called 'count_and_find.txt' (the first line is the count, followed by one target word each line). Make sure that there is a newline character (\n) at the end of **each line**.

(3) **(50%)**Read in the cleaned file you just created. Convert every word to lowercase. Use the code you wrote in the last practical session, compute the probability of each line of text (including title and the author name) under the **trigram** assumption. **Include punctuation marks in your probability calculation.**
Write the result to a new file called 'trigram_prob.txt' in the following format: each line contains a line id (starting from 0) and the corresponding probability (to three decimal places), separated by a : and a space character. Sort the lines in a decreasing order by probability (for lines that have the same probability, **sort them increasingly by id**). For example:

```
3: 0.139
5: 0.139
0: 0.075
…
```

Make sure that there is a newline character (\n) at the end of **each line**.

What you need to submit:

- Your code (named 'prac5.py', written in Python 3, with appropriate comments)
- Three output files: Lovers_on_Aran.txt, count_and_find.txt, trigram_prob.txt

Please **drag and drop** the files in the submission box for Practical 5 on GradeScope, and **do not zip** anything.

What you'll be graded on :

- 90%: Correctness of output files
- 10%: Code style (clarity, readability, and elegance)