COMP 206 Assignment 4

Python programming exercise
Due Dec. 5, 11:59PM

1 Pedagogical objectives

Learn basic Python programming skills in domains where Python is effective, and especially as applied to problems that would be difficult in C. Get exposed to input, output and basic data structures. See that you can do something useful and substantive with a limited amount of code.

2 Problems

Write three Python programs that:

1. (20 marks) Opens a text file whose path is given on the command line, counts and prints the number of times each word occurs. Words must be converted to lower case only and you must remove all non-alphabetic characters. Hyphenated words should be split into two (or more). Each line of your program's output must have one word and its corresponding frequency in the format "word:frequency". Output must be sorted by descending frequency. Your code must be written in *q1_word_count.py*.
2. (30 marks) Perform the same operation as Q1, but count word pairs. Output is a word-pair followed by the number of times it was found, in the format "word1-word2:frequency". The lines must be sorted by descending frequency. Your code must be written in *q2_pair_count.py*.
3. (50 marks) Write a chat-bot program, *q3_chat_bot.py* that generates responses using a simple data-driven language model. The program must parse word pairs from one or more text file whose paths are given as command line arguments. The user interaction is terminal based (stdin, stdout) and must follow the same format as Q2 and Q3 from Assignment 3. That is Send: and Receive: are written, followed by a

one-line message. We will not use any chat handle here. Each time a user enters a query message (one single line of text ended by new-line), your program must immediately generate a response message where:

- The first word in the response (R1) starts with a capital letter.
- If the user's last word (QN) occurs in the text, R1 must be chosen so that QN-R1 has been seen in the text at least once. If QN was not seen in the text, R1 is allowed to be any word from the text.
- Every word pair in the response (RI-RJ) must have been seen at least once in the text.
- The response can end in three ways. (1) On a stop-pair. These are pairs that ended a sentence in the text at least once. Every time a stop pair is output, the response must end. (2) If a word RI is chosen for which no pair RI-RJ exists, then RI must be the last word in the response. This only happens when RI ends a text and has never been seen otherwise. (3) If twenty words have been correctly generated without the first two conditions being met, then word twenty will end the response.
- The last word in the response must be followed by a period

3 Suggestions and Hints

The following Python functions and libraries are likely to be useful:

- String functions: lower(), upper(), replace(), strip(), split(), join()
- The dictionary type and functions: has$_key(), iteritems(), keys() The sort() function$

- The random library

For sample text files, we suggest the following:

1. I Was A Teenage Hacker from
   `http://www.textfiles.com/history/drake.txt`
2. The Case-Book of Sherlock Holmes from
   `http://www.cim.mcgill.ca/~dudek/holmes.txt`

The holmes.txt file originally comes from `http://www.gutenberg.ca/ebooks/doyleac-casebookofsherlockholmes/doyleac-casebookofsherlockholmes-00-t.txt` but you should not get it there (except in case of emergency) to minimize traffic on that resource.

WARNING: The following statistics have not been confirmed for this year and may have errors due to small differences in the spec from the last time they were used. I will post a version that removes this warning once it's confirmed. The first (I Was A Teenage Hacker) is small and easy to work with, and the second is large and substantive and might make debugging slower (it has about 44,485 word-pairs), so start with the small file first. The "hacker" file, if everything is converted to lower case and most punctuation is stripped (so the word "end." and "end" is regarded as the same) has 866 consecutive words, 433 distinct words, and 793 distinct word-pairs. The fact that the number of word-pairs is a bit smaller than the number of words is because about 75 words pairs like "telephone company" occur twice. For the Holmes file, the most common word pair is "of the" which occurs 447 times while "sherlock holmes" occurs 31 times.

The Case-Book of Sherlock Holmes (1927) by Arthur Conan Doyle, can be downloaded like this:

```
curl http://www.cim.mcgill.ca/~dudek/holmes.txt  > holmes.txt
```

Restrictions You are not expected to know about nor allowed to use the Collections module nor the Counter type since they make the problems too easy if you know them already, and they add too much extra baggage.