LING 165 Lab 3: N-grams

Write a program that generates random sentences using n-grams.

Data

We have a collection of sentences related to corporate strategy. The sentences are from here, a project site for an open-source random text generator. The collection is available as a `txt` file on the gray server:

- `/home/ling165/lab3/bullshit.txt`

The structure of the file is straightforward: one sentence per line.

Task

Write a program that remembers n-grams in the collection and generates sentences by randomly chaining n-grams.

You should allow the user to specify n of n-gram. For example, here's me using my program with $n = 2$ and $n = 3$:

```
[hahnkoo@gray lab3]$ python monkey.py 2
The President of Business Operations Officer organically facilitates interconnected
emotional intelligence diligently engineer our usage-based , siloed mindsets .
[hahnkoo@gray lab3]$ python monkey.py 3
The partners focus on our enterprise-wide sign-off , while the thinkers/planners
significantly adapt an enhanced book value growth .
```

Email me (1) 10 random sentences generated using 2-grams, (2) 10 random sentences generated using 3-grams, and (3) where I can find your code.

More

Assume the following when extracting n-grams from data:

(1) Each sentence is padded with boundary markers: $n - 1$ tokens of `<s>` at the beginning and a single `</s>` at the end.

(2) Each boundary marker is a word.

(3) Any string of characters delimited by white-space is a word. This includes punctuation symbols `;` `,` `.` that appear by themselves.

N-grams form a chain if the *suffix* of an n-gram ($n - 1$ words at the end) matches the *prefix* of the next n-gram ($n - 1$ words at the beginning). For example, the following 3-grams form a chain:

`this is a, is a very, a very nice, very nice example, nice example of`

A chain of n-grams constitutes a sentence if its first n-gram begins with $n - 1$ tokens of `<s>` and its last n-gram ends with `</s>`. For example, the following chain of 3-grams is a sentence:

`<s> <s> this, <s> this is, this is a, is a sentence, a sentence ., sentence . </s>`

The corresponding sentence, with the boundary markers removed, would be:

`this is a sentence .`

To generate a sentence by randomly sampling a chain of n-grams, do the following:

(1) Initialize sentence to an empty string.

(2) Pick an n-gram that begins with $n-1$ tokens of `<s>`.

(3) Append the last word of the n-gram to sentence.

(4) While the last word in sentence is not `</s>`, do the following:

- Pick an n-gram whose prefix matches the suffix of the latest n-gram we picked.
- Append the last word of the n-gram to sentence.

(5) Return sentence.

When you pick n-grams, make sure the choice is random and reflects their frequencies. For example, if `the man` appeared in data twice as often as `the book`, then `the man` should be twice as likely to be chosen as `the book`. One way to do this in Python is to store all tokens of n-grams with common prefix in the same list and then using `random.choice` (see here) to draw from that list.