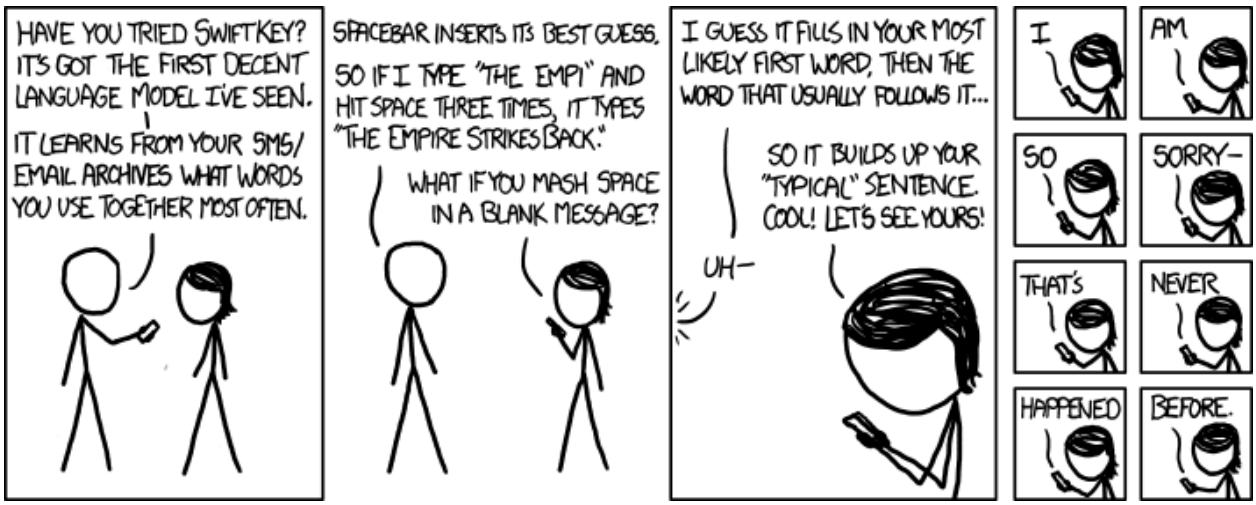
Assignment 1 - Language Modeling and Generation



In this assignment you will implement a Markovian language model and a language generator. The purpose of this assignment is to bring you up to speed with your Pythonian text manipulation skills, simple conditional probabilities and the power of simple language models.

The code you write in this assignment will be used in the next assignment as well.

Submission deadline: 23:59, Nov. 7 (Sunday, before midnight)

Submission Guidelines

1. You should use the course Moodle to submit a single **gz** file containing a single python file. (and only gz!)
2. Your code file should be named **ex1.py**, and must contain all necessary methods/functions that support the specified API.
3. Your code shouldn't print anything to standard output!
4. You should use **python 3.9**.
5. You can import only the following modules: re, sys, random, math, collections
6. You should **document your code** using either Google Style or Pyhton PEP 257. You are encouraged to conform to the style guides in general (e.g. naming conventions, line length etc.).

API

Your code should support the following API.

```
def __init__(self, n=3, chars=False)
def build_model(self, text)
def get_model_dictionary(self)
def generate(self, context=None, n=20)
def evaluate(self,text)
def smooth(self, ngram)
```

Global functions:

```
def normalize_text(text) #not a class method
def who_am_i()  #not a class method
```

Detailed documentation: API
You can add as many class methods as you like.

Notes:

1. The implementation (and the internal data representation) could be much more efficient, however, we wish to keep it simple for the sake of clarity.

An example and a driver

Your submission will not be accepted if your code crashes/throws exception on the following simple driver: ex1_driver.py
An example of expected output (you should have identical output for the lines marked with #*):

```
a cat sat on the mat . a fat cat sat on the mat . a rat sat on the mat .
the rat sat on the cat . a bat spat on the rat that sat on the cat on the
mat .
defaultdict(<class 'int'>, {'a cat sat': 1, 'cat sat on': 2, 'sat on the':
5, 'on the mat': 4, 'the mat .': 4, 'mat . a': 2, '. a fat': 1, 'a fat
cat': 1, 'fat cat sat': 1, '. a rat': 1, 'a rat sat': 1, 'rat sat on': 2,
'mat . the': 1, '. the rat': 1, 'the rat sat': 1, 'on the cat': 2, 'the
cat .': 1, 'cat . a': 1, '. a bat': 1, 'a bat spat': 1, 'bat spat on': 1,
'spat on the': 1, 'on the rat': 1, 'the rat that': 1, 'rat that sat': 1,
'that sat on': 1, 'the cat on': 1, 'cat on the': 1})
a cat sat on the mat . a fat cat sat on the mat . a bat spat on the mat .
a rat sat on the mat . | -10.488
a cat sat on the mat | -4.297
the rat sat on the cat | -4.990
```

Resources

Reading

- Class slides, and/or:
- J&M Chapter 2 (2.2-2.4)
- J&M Chapter 3 (3.1, 3.3.1, 3.4)
- J&M Appendix B

Corpora

Here are some corpora to use. Feel free to add other resources.

- Norvig's big.txt file
- Trump's historical tweets (~14K tweets and retweets by Trump, each tweet in a new line)
- Obama weekly speeches 2009 - 2015
- Complete Works of William Shakespeare in one text file
- Random typing

Think/Experiment:

1. What would you expect from the different resources?
2. Do the different corpora require different preprocessing/normalization?
3. What happens if you train a model on one corpus and test (evaluate) it on a different corpus?

Documentation styles

- Google style guide: http://sphinxcontrib-napoleon.readthedocs.io/en/latest/example_google.html
- Python PEP 257 style guide: https://www.python.org/dev/peps/pep-0257/

Sandbox

We recommend you play with your code a bit - using different corpora to generate the language model, using different n for the n-grams and even generating texts (e.g. Trumpian tweets) using the language models you learnt, and playing with the various parameters that govern the language model and language generation..

Integrity and Cooperation

You should work on your assignments alone and refrain from sharing code snippets. However, you are encouraged to discuss various aspects of the assignment in the dedicated assignment forum and you are welcome to share testers and additional corpora.