HG2051 – Language and the Computer
Computational Linguistics with Python

Michael Wayne Goodman
Postdoctoral Research Fellow, School of Humanities

Week 1: Introduction, Organization, Main Issues

Today's Session

- Personal Introductions
- Administrivia
- Course Overview
  - Why use computers in linguistics
  - What this course is and isn't
- Getting Started
  - Algorithmic thinking
  - Environment Setup
  - Basics of Version Control
  - Running Python
  - Introduction to GitHub
  - Homework 1

Self Introduction

- BS in Computer Science, Minor in Japanese
- MA and PhD in Computational Linguistics
- Dissertation: *Semantic Operations for Transfer-based Machine Translation*
- Interned at the National Institute for Information and Communications Technology (NICT), Japan
- Contracted at Microsoft Research (MSR) on the Machine Translation team
- Research Associate at NTU
- Postdoctoral Research Fellow at NTU

Student Introductions

- Your name
- What do you know about programming?
  - Javascript, C/C++, Excel spreadsheets
  - Algorithms, security concerns, ethical concerns
  - "nothing" is ok!
- What do you want to get out of the course?
  - "nothing" is not ok!

Administrivia

- Schedule: see `https://ntu-hg2051.github.io/`
- Continuous Assessment:
  - Assignments (40%)
  - Group Project (20%)
  - Exam 1 (15%)
  - Exam 2 (15%)
  - Participation (10%)
- Extra Credit: You can get 1∼5% extra credit by getting a patch accepted to an open source project related to the course (e.g., NLTK). Your total grade cannot go over 100%. Contact me if you're interested.

Why use Computers in Linguistics?

- Linguistics without computers is like taking a walk (or a long, hard hike)
  - It can be very pleasant
  - You can see lots of details
  - There is only so much ground you can cover
- Using a software tool is like catching the MRT
  - Very efficient for set routes
  - You have to adapt to it
  - Hard to customize
- Programming is like driving a car
  - It is expensive to start off (you have to learn!)
  - You are free to go where you want to

The Goal of this Course

To learn enough about programming to flexibly analyze data and then do something with it

- Coding is done in Python
- We will learn techniques and some software libraries particular to computational linguistics
- You will be able to write your own programs by the end

HG2051 Prerequisites

- A little linguistic knowledge
  - You know what a word is
  - You know what a part of speech is
  - You know what a parse tree is

  If you don't know these, you will have to do a little background reading

- No computational knowledge
  - You have to be ready to learn
  - If you are a very experienced Python programmer, then you will not learn so much

What HG2051 isn't

- We won't be learning how to build cars
  - this is the prerequisite for further NLP courses
  - ... but we won't be writing taggers and parsers yet
- It is not just an introduction to Python, but rather one motivated by NLP
- It is not very easy, but it is fun

The Three Virtues of a Programmer

- **Laziness**: The quality that makes you go to great effort to reduce overall energy expenditure. It makes you write labor-saving programs that other people will find useful, and document what you wrote so you don't have to answer so many questions about it.
- **Impatience**: The anger you feel when the computer is being lazy. This makes you write programs that don't just react to your needs, but actually anticipate them. Or at least pretend to.
- **Hubris**: The quality that makes you write (and maintain) programs that other people won't want to say bad things about.

Larry Wall, Tom Christiansen, Randal L. Schwartz, and Stephen Potter (1996) *Programming Perl* 2nd Ed, O'Reilly.

Readings

- Readings will come from a variety of freely available sources (no required textbook)
- You must read the material before class
    - I will assume you have done so
    - Programming is not (just) knowledge but a skill; we should spend our class time practicing that skill

Algorithmic Thinking

- Exercise: How to make kaya toast
- Also see:
  `http://www.cookingforengineers.com/`

(short break)