HG 2051: Language and the Computer (Computational Linguistics with Python)

Week 1: Introduction, Organization, Main Issues

Today's lecture/session

- Introductions and preliminaries
- Administrative matters
- Course overview
  - Why computers & linguistics?
  - What this course is (and what it isn't)
- Getting Started
  - Algorithmic thinking
  - Environment Setup
  - Basics of Version Control
  - Running Python
  - Introduction to GitHub
  - Homework 1

Instructor background

Dr. Hiram Ring

I'm a field linguist with interests in:

- language documentation/description (PhD, NTU 2015)
- linguistic typology and language contact
- historical reconstruction
- natural language processing and machine learning (since 2015)
- SEAsian languages, particularly of the Austroasiatic phylum

I also write/record/perform music (www.hiramring.com)
and maintain a website for classification of implicit motives
(www.implicitmotives.com)

Main fieldwork site

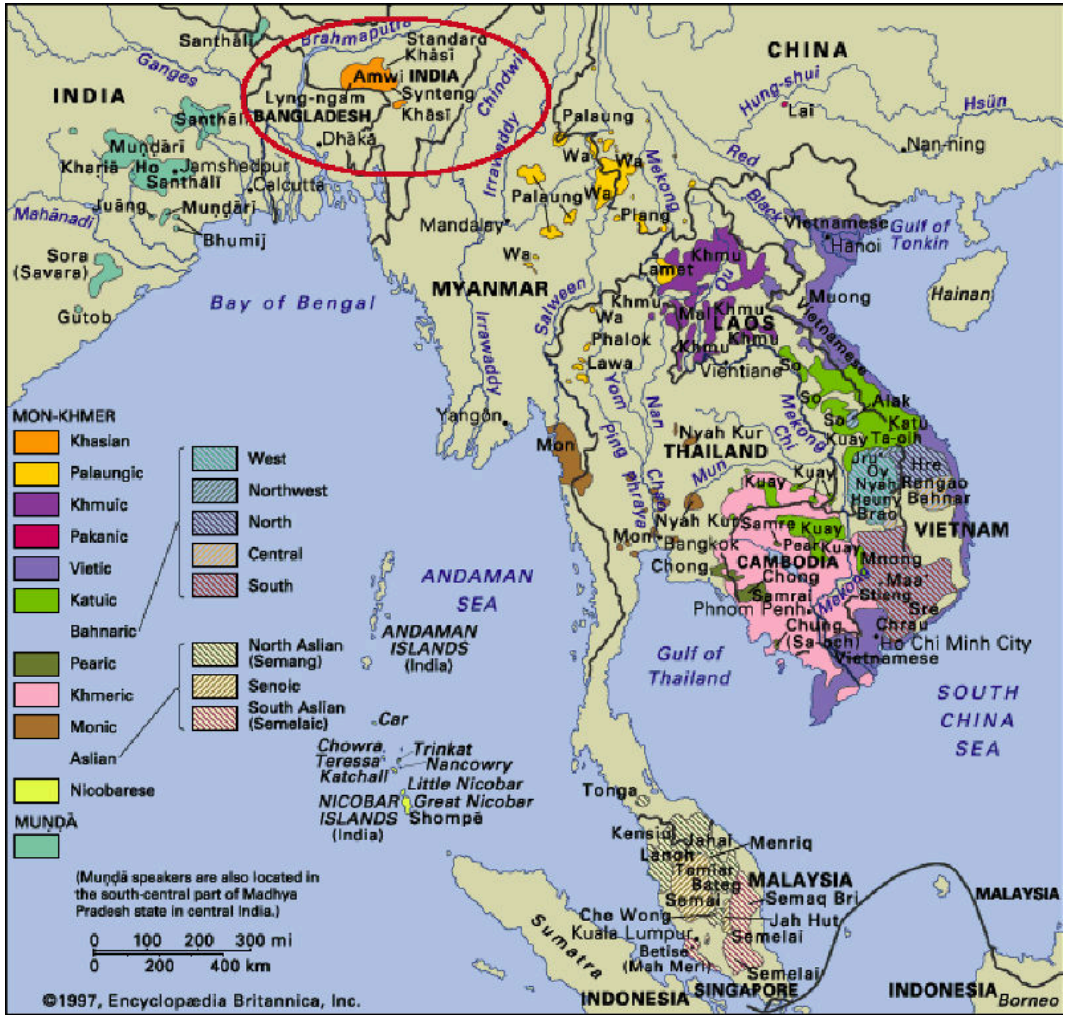
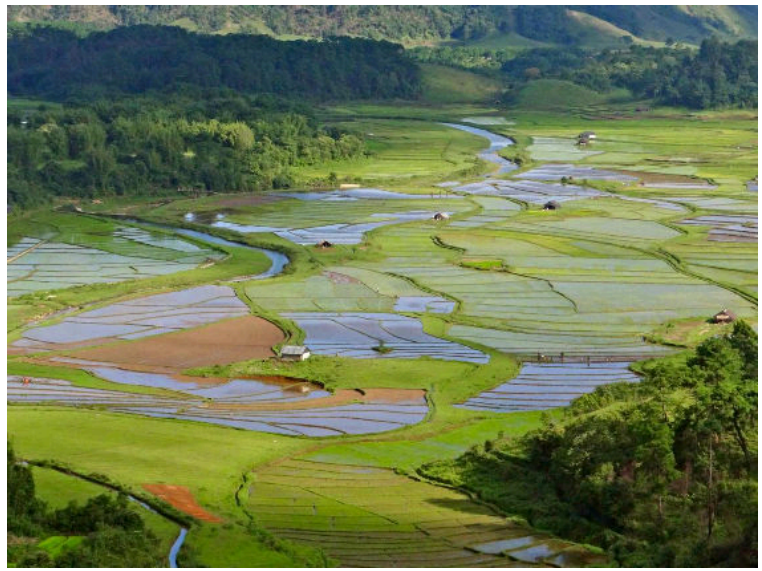


Figure 1: The Austroasiatic languages

Main fieldwork site: Meghalaya, India



Instructor contact

Dr. Hiram Ring
Office: SHHK-03-20
E-mail: hiram.ring@ntu.edu.sg

Consultation hours: email me for an appointment – either in person, or online

Student Introductions

- Name, year, linguistic interest (i.e. field/course of study)
- What background or knowledge do you have concerning programming?
  - Javascript, C/C++, Excel spreadsheets, R, Python etc.
  - Awareness of what programming is used for (algorithms, security, etc..) and related concerns
  - "nothing" is ok
- Why are you taking this course? (What do you hope to gain out of it?)
  - NOT "nothing"

Administrative matters

- Schedule: https://hg2051-ntu.github.io
- Continuous Assessment:
  - Homework (autograded)
  - Project 1 (30%)
  - Project 2 (30%)
  - Quiz 1 (15%)
  - Quiz 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.

Continuous Assessment components

- Homework (autograded):
  - gives you practice writing actual code and submitting it
- Project 1 (CA1: 30%)
  - Write a Python program that uses text corpora to evaluate or produce a new resource, then write a brief 4-page paper describing the process and findings. [TBD]
- Project 2 (CA2: 30%)
  - Work with a group of 3-4 to develop a new resource or improve on an existing resource, writing a program to process the data. Summarize your process and findings in an 8-page cowritten paper. [TBD]

Continuous Assessment components

- Quiz 1 (CA3a: 15%)
  - Midterm quiz (2hrs) involving a programming challenge with the ability to access online resources.
- Quiz 2 (CA3b: 15%)
  - Final quiz (2hrs) involving a programming challenge with the ability to access online resources.
- Participation (CA4: 10%)

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)
- A computer running Windows, Mac OS, or Linux
  - It is possible to learn using school computers, but it will be much harder
- No computational knowledge required
  - 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

Homework

- Homework will comprise various practice problems aimed at familiarizing you with practical programming skills/tools.
- Some general guidelines:
  - Try to type everything on your own (at least at first), don't just copy/paste. This can be time-consuming initially, but the process will help you to remember the coding elements.
  - Try to understand the logical steps/process so you can apply them to different problems in the future.

Algorithmic Thinking

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

(break)

Environment setup, Git, Python, Homework

https://hg2051-ntu.github.io/week1.html