ENGL 516X:
Methods of Formal Linguistic Analysis

Semester: Spring '18

Instructor: Sowmya Vajjala

Iowa State University, USA

9 Jan 2018

Class outline

- Introductions
- Motivation for the course and objectives
- Course logistics
- Course Outline and Syllabus
- Comments from past students

Introductions

About me

1. In ISU as Asst. Professor since January 2016.
2. Education: PhD in Computational Linguistics, 2015.
3. Teaching experience:
   - ALT students@ISU: 516, 520
   - Other Graduate courses@ISU: 515, 590 (Independent study supervision)
   - Undergrad@ISU: 120, 314, 410
   - Topics seminars for computational linguistics students@UTü (2012-13)

About you

1. Name
2. What do you study in ISU?
3. Why do you want to learn computer programming?

About the course

Course Objectives

- Introduce you to the world of computer programming.
- Make you understand why this is an important skill even for applied linguists in today's world
- Make you have a problem solving mind set which will be useful in doing your research too

Expected Learning Outcomes

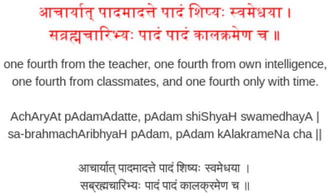
- Given a problem description, think about an approach to solve the problem, design and write computer program(s) to implement the approach
- Understand how to find and fix errors in your programs (debugging)
- Know where to look for to not reinvent the wheel. Read the documentation for existing code and write their own programs extending them.
- Understand how text is represented on computers, and how to work with it using regular expressions
- Write small programs that involve reading, processing and writing content into files on the computer
- Implement simple browser based applications involving collecting and displaying data to the user

What will you not become?

1. Expert programmer
2. Expert Python programmer
3. Mr/Ms Know it all about computers

What will you not become?

1. Expert programmer
2. Expert Python programmer
3. Mr/Ms Know it all about computers



आचार्यात् पादमादत्ते पादं शिष्यः स्वमेधया ।
सब्रह्मचारिभ्यः पादं पादं कालक्रमेण च ॥

one fourth from the teacher, one fourth from own intelligence,
one fourth from classmates, and one fourth only with time.

AchAryAt pAdamAdatte, pAdam shiShyaH swamedhayA |
sa-brahmachAribhyaH pAdam, pAdam kAlakrameNa cha ||

आचार्यात् पादमादत्ते पादं शिष्यः स्वमेधया ।
सब्रह्मचारिभ्यः पादं पादं कालक्रमेण च ॥

Pre-requisities

1. familiarity with using computers
2. interest in learning how to program for computers
3. passion about developing and using information technnology for educational applications
4. background in linguistics is useful but not mandatory.

Motivation

Why should I learn programming?

- Understanding some of the technology: We use a lot of products that involve programming in everyday life. Your computer and mobile phone are the most used, for example.
- You can automate some boring, repetitive tasks using a program. (e.g., everyday, at 5pm, take a backup of folder X and save it on the server).
- You can write your own custom programs to do specific corpus analysis, or identify a specific kind of error in student writing and so on.
- Programming gives you instant gratification :-)

Programming and Applied Linguistics

Some Practical Examples

- develop your own modules in CALL systems.
- write programs that highlight grammar/spelling mistakes in student writings
- create test exercises automatically for the class (e.g., writing code to create fill in the blanks questions)
- Those who want to do user studies: set up our own experiments, to test the phenomenon we want to study (e.g., use of phrasal verbs by English learners of Turkish descent).
- Programs that analyze the student essays and point of lack of coherence or illogical statements (alright, far fetched.. but still).

... and so on.

Course Logistics

Meeting and Location

- meets in Ross 312, on Tuesdays and Thursdays from 12:40-2 pm
- *Office hours:* Tuesdays and thursdays, 2-3 pm (please email beforehand if there are specific issues to discuss)
- course website: on Canvas.

Credits

- Credit Points: 3
  => (3 hours of classroom instruction + 6 hours of additional effort on student's end).
  Note: This course is very different from what you are used to in Applied Linguistics. Be prepared to not "get it right" the first time (No one does!)

Format and Grading

Course Format

- weekly lectures/practical sessions, readings
- participation in discussion forum
- 7 assignments
- final exam - programming project.

Assignments

- 7 Assignments (65%)
- Final project (30%)
- Classroom participation (5%)

Deadlines and timeline of general events are all in the syllabus document. [Take a look]. First 3 assignments are already uploaded. Rest will be uploaded in the coming 2-3 weeks. Final project descriptions also will be up in a few weeks.

Syllabus - Topics

1. introduction to computing and programming.
2. introduction to python and making python work on your computers
3. basic concepts of writing code in python (variables, functions, recursion etc.)
4. Python datastructures: lists, dictionaries, tuples etc
5. strings and regular expressions
6. reading and writing text from files
7. understanding program flow, using other people's code in your code
8. database basics
9. unix fundamentals

Text Book and Other Material

1. Primary textbooks (both are free ebooks)
   1.1 Python for Informatics by Charles Severence
   1.2 How to think like a computer scientist: Learning with Python3
2. Other useful resources:
   - Free online courses in Coursera by Charles Severence, under "Python for Everybody" specialization series
   - Unix for Poets, free ebook by Kenneth Ward Church

... and several other resources online.

Some additional references on Canvas in readings.pdf file.

Relevance of Assignments

- There are two goals for the assignments, and classroom exercises:
    1. Give you more practice with the programming language (learning the correct syntax, using the right vocabulary, teaching you problem solving)
    2. Give you some background in working with text corpora

Relevance of Assignments

- There are two goals for the assignments, and classroom exercises:
  1. Give you more practice with the programming language (learning the correct syntax, using the right vocabulary, teaching you problem solving)
  2. Give you some background in working with text corpora
- Sometimes, the assignments may look very irrelevant to your own background (e.g., if I ask you to reverse a number i.e., 123 should become 321). But, there is a reason behind asking such a question (practice with different functions in python, how to solve a new problem, etc)

Relevance of Assignments

- There are two goals for the assignments, and classroom exercises:
  1. Give you more practice with the programming language (learning the correct syntax, using the right vocabulary, teaching you problem solving)
  2. Give you some background in working with text corpora
- Sometimes, the assignments may look very irrelevant to your own background (e.g., if I ask you to reverse a number i.e., 123 should become 321). But, there is a reason behind asking such a question (practice with different functions in python, how to solve a new problem, etc)
- So, don't think I am stupid or you are stupid. Don't also start imagining you know better than me about how to teach this particular subject. Irrespective of your tons of experience in some other field, we are here in these roles because I have more experience in this topic. So, trust my decisions.
- You can ofcourse give me a wishlist of things you want to see in this class.

Some general rules:

- attendance: not mandatory, but recommended.
- deadline extension: with penalty
- long absence due to illness etc: please inform and follow university procedures.
- cheating and plagiarism: see the course handbook, and university policy against plagiarism.
- classroom behavior: please be punctual and do not do personal work in the class.
- Disability accomodation: Please speak to Disability Resources Office (DRO) to officially request an accomodation.
- reporting grievances: follow standard procedures.

Any questions so far?

A small problem solving question

source: hackerrank.com

Think about this problem described below (work in groups), and try to understand how to solve it:
Davis has staircases in his house and he likes to climb each staircase 1, 2, or 3 steps at a time. Being a very precocious child, he wonders how many ways there are to reach the top of the staircase. Given the number of steps in a stair case, how many different ways can be climb it? Is there a pattern to predict the combinations based on number of steps?
E.g., 1 step - only 1 way. 2 steps, there are two ways (1 step at a time, or 2 steps at once) etc.

Comments from old students

- Kimberley Becker
- Jordan Smith
- Ziwei Zhou

ToDo before next class

- Read Chapter 1 in the textbook (Severence).
- Read the article ”The English Teacher as a programmer” (http://computersandcomposition.candcblog.org/archives/v4/4_3_html/4_3_5_Costanzo.html). Participate in the discussion forum and share your thoughts about it.

Next Class ..

- Discussion about the readings
- A brief history of computing
- How does a computer work?
- What is a program?
- TODO: Go through the Canvas website, check the syllabus document, ask questions if any, on the discussion forum.