Introduction to Programming for Linguists

Adriana Picoral

Spring 2020

Course Description

This course introduces students to programming for dealing with textual data. The course will focus on 1) basic programming skills, including using the command line (e.g., bash, PowerShell) for data management, and 2) Python code writing to manipulate textual data. We also cover concepts related to corpus data, such as nomenclature (e.g., token, type) and issues related to calculating frequencies. Algorithms addressed in this course include tokenization, stemming, lemmatization, and summarization.

Course Learning Goals

At the end of this course, students will be able to:

- use a command line terminal
- run counts on text files using both their command line and Python
- write Python scripts to process text files individually and in batches
- annotate text files for stem, lemma, part of speech, etc.
- produce a summary of frequencies of annotations

Prerequisite Requirements

No prior programming experience is expected or required. Access to a computer with internet access is required for classroom and homework assignments.

Course Readings

Course readings will be mainly assigned from:

- Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Supplementary readings will consist of published articles.

Assignments and Grades

Final grades are based on:

Weekly assignments (50% of total grade)

Every week a small programming task (e.g., count tokens in one text file) will be assigned. Each weekly assignment will give students a chance to practice the skills learned during class time. Study groups to discuss these assignments are encouraged, but programming solutions must be written up individually.

Final project (40% of total grade)

- proposal (5%), code (15%), paper (15%), presentation (5%)

The final project for this course consists of a small corpus analysis. Students are to propose what type of corpus they are to use and what research question they are interest in answering, based on language used in the corpus. This proposal must be approved by the course professor before students start to implement their data analysis.

The paper must have the following sections: introduction, literature review, methods (data description, analysis), results, and discussion. Students deliver a 10-minute presentation on their project during the last week of class.

Attendance and Participation (10% of total grade)

Class attendance is required. Excused absences (professional or personal reasons) do not result in grade points loss. However, students are responsible for materials covered during class and are expected to turn in assignments on time.

Participation is defined as being attentive during class, asking and answering questions posed by peers and professor during class time or on our Slack channel.

Course Schedule

Week 1 - Intro + Command Shell

- syllabus
- code of conduct
- intro to command shell
- navigating files and directories

Week 2 - Command Shell

- navigating files and directories
- counting files
- counting words in files
- checking text file encoding
- organizing data in folder structure

Week 3 - Python

- installing Python
- installing Python packages
- running python from command line
- basic structure of a Python script
- running a Python script

Week 4 - Python + If blocks

- argument parsing (argparse package)
- if conditionals
- writing error messages

Week 5 - Python + For loops

- argument parsing (argparse package)
- reading a text file
- for loops

Week 6 - Tokenization

- reading text files in Python
- splitting by space
- dealing with sentence tokenization
- counting tokens and types

Week 7 - Python + Functions

- writing a function in Python
- reading text files in Python
- calling token counting function in Python

Week 8 - Tokenization + Frequency

- frequency of tokens and types
- calculating normalized frequencies
- Zipf's law

Week 9 - Tokenization + Frequency

- calculating measures of association strength
- intro to stopwords
- removing stopwords from counts

Week 10 - ngrams

- intro to ngrams
- extracting ngrams from text files
- calculating frequencies of ngrams
- calculating measures of association strength of ngrams

Week 11 - Stemming

- intro to stemming
- Porter Stemmer
- Snowball Stemmer
- Lancaster Stemmer

Week 12 - Stemming + File annotation

- stemming text files
- writing out text files in Python
- annotating text files
- different annotation formats
  - underscore and angled brackets
  - tab and space separated
  - CoNLL format
- creating folders in Python

Week 13 - Regular Expressions

- intro to regex
- identifying patterns in text files using regex
- debugging regular expressions

Week 14 - NLTK

- installing NLTK
- tokenization in Python with NLTK
- lemmatization in Python with NLTK
- part of speech annotation with NLTK

Week 15 - Summarization

- sentence representation
  - bag of words
  - one-hot encoding
  - TF-IDF
- scoring sentences based on frequency counts
- Luhn algorithm

Week 16 - Final Project Presentations