Ling 441
Computational Linguistics
Fall 2015

Vitals

Lectures:	Tue/Thu, 4–5:30, 473 Lorch Hall
Instructor:	Steven Abney (`abney@umich.edu`), 412 Lorch Hall
Course materials:	on CTools
Office hours:	Tue/Thu, 10:00–11:00, or by appointment

My office hours are times specifically set aside for meetings with students. But any time my door is open and I'm not in a meeting, feel free to stop in.

Description

The goal of computational linguistics is the construction of explicit models of language processing, generally implemented in software. We can distinguish at least four varieties of computational linguistics, differing in the larger motivations for studying language processing:

- **Natural language processing** is the branch of AI that concerns the language portion of an artificially intelligent agent. This is in many ways the ultimate goal of the field.
- **Computational psycholinguistics** asks which of the possible models of language-using agents gives the best account of psycholinguistic data.
- **Human language technology** focusses on self-contained technologies, the most important of which are machine translation, information extraction, and spoken language dialogue systems.
- **Digital linguistics** employs language processing technology in service of language documentation and linguistic research.

This class will cover foundational material that applies in all of these areas. The first part of the course focuses on getting useful information out of text corpora, culminating in language learning from texts. The second part of the course focuses on parsing and semantic interpretation, and on writing "computational" grammars to drive parsers and semantic interpreters.

Prerequisites

One cannot get a real idea of what computational linguistics is about without writing programs. We will use the Natural Language Toolkit (NLTK), which is a library module for the programming language Python.

Previous familiarity with Python is not necessary. We will learn it as we go along. (Even though it is a means to an end in this course, the ability to use Python is a valuable skill in itself.)

It *will* be assumed that students have done at least a little programming before. We will cover everything from scratch, but we will go quickly through basic notions like variables, if-then statements, loops, and simple function definitions. It is assumed that students have used a plain-text editor like `vim` or `emacs` or `pico`.

That being said, the degree of computational expertise assumed is minimal. This course is explicitly designed for students in linguistics, cognitive science, language departments, etc. Students with advanced computational skills may wish to consider taking EECS 595, Introduction to Natural Language Processing.

On the linguistics side, there are no specific prerequisites. It is recommended that linguistics students take Ling 315 (Syntax) and Ling 316 (Semantics) first. Some basic grammatical knowledge will be assumed: familiarity with the basic parts of speech, the ability to identify subject and predicate, prepositional phrases, and the like.

Text

The textbook is Bird, Klein, and Loper, *Natural Language Processing with Python,* first edition (O'Reilly, 2009, ISBN 978-0-596-51649-9). The book is available online; there is a link on the CTools site.

Handouts and assignments will be posted to the CTools site.

Requirements

The grade will be based on regular homeworks. There will probably be some quizzes and/or tests as well.

All materials will be posted on CTools, and homeworks will be submitted on CTools. Homeworks will generally be due at the beginning of the next class meeting. However, there is a 24-hour grace period to turn in the homework without penalty. If you have trouble with a homework, you are encouraged to ask questions in class, then complete it and turn it in within 24 hours.

Schedule

The plan is to work through a large part of the book. The following is a tentative schedule. See the CTools site for the current schedule as the term proceeds.

1. Intro

Python crash course
2. Python expressions
3. Functions, conditionals, iteration

Corpora
4. Wordlists and frequencies
5. Corpora and lexica
6. Collocations
7. Language models

Lowlevel processing
8. Files, URLs, Unicode
9. Regular expressions
10. Part-of-speech tagging

Information extraction
11. Text classification
12. Chunking and named-entity recognition

Parsing
13. Syntax
14. Parsing
15. Grammars
16. Trees
17. How parsers work

Feature grammars
18. Feature grammars
19. Grammar writing
20. Movement

Interpretation
21. Predicate calculus
22. Semantic interpretation
23. Quantifiers
24. Reasoning