Computational Linguistics I

LING 4424 / CS 4744 / COGST 4240
Spring 2025
TuTh 11:40AM-12:55PM
Hollister B14

Mats Rooth
Morrill 203A (Enter through the Linguistics main office.)
Office hour: Weds 4-5 (I will stay later if people are waiting. Check for announcements about rescheduling.)

Fengyue Lisa Zhao
Morrill B11 (located in the basement, enter through the phonetics lab.)
Office hour: Thursday 9:30–10:30 or by appointment

This course introduces computational methods and models in varied subfields of linguistics, including syntax, semantics, phonology, and phonetics. It covers formalisms and computational models that are relevant to theoretical linguistics, and nuts-and-bolts computational methods. The emphasis is on symbolic models. Neural models are covered in Natural Language Processing (CS 4740/LING 4474/COGST 4740) and Computational Linguistics II (LING 4434/CS 4745).

The class includes a project component that is developed over the semester. A project report including text, code, and experimental results is submitted after the end of classes on the date designated by the registrar.

Prerequisites

Elementary Python (CS 1133 suffices) and
LING 1101 (Introduction to Linguistics) or CS 2800 (Mathematical Foundations of Computing) or PHIL 2310 (Introduction to Deductive Logic). CS majors should have completed CS 2800, whatever course number for this class that they register for.

Topics

1. Tree syntax, Context free grammar
2. Tabular Parsing
3. Feature constraint grammar
4. Logical semantics and truth checking
5. Finite state transducers
6. Generative phonology
7. Non-CF syntax, Minimalist grammar, Multiple context free grammar
8. HMM speech recognition

We will work with topics 1-7 computationally using Python toolkits. As much as possible, the NLTK toolkit is used. Depending on available time, topic 8 will be covered only theoretically, or with a toolkit. Possible additional topic:

9. OT phonology

Project

The project is an end-to-end system which maps a given audio utterance of a sentence, and a given string, to a truth value. For instance, given an audio utterance of the sentence 'no vowel is capitalized and adjacent to a vowel', and the string 'OEUE', the system should return False. Some data from 2023 is at https://github.com/MatsRooth/stringtruth. There are deliverables for the project throughout the semester.

While you will submit an individual project report, data and some methodology are shared. While everybody does the same basic project, there is ample scope for doing something original and creative.

Requirements

Six problem sets
Midterm prelim (covering 1-4 and possibly 5) - In class on **Thursday, March 20**.
Second prelim (covering 5-8) - In class on **Thursday, April 24**.
Project
Participation in class and on forum

Class attendance is obligatory. You may miss two classes without penalty, after that the penalty is 2 points per missed class. Email the instructor and the TA if you plan to miss a class, or otherwise miss one. Sign in at the start of class.

Please submit problem sets on time. There is a penalty of 5 points per day for problem sets submitted late.

Letter grades are assigned on a curve, with the distribution typical in Linguistics classes. Scores on problem sets with unusually low means are scaled up linearly before scores are distributed.

Computational environment

In Computational Linguistics and NLP, Python is the language of choice for research and development, communicating ideas, and exchanging functionality. It is a basic teaching language at Cornell. You should have familiarity with elementary Python coming into the class.

Many lectures and problem sets use Jupyter notebooks. It will be necessary to install various packages, some requiring different versions of Python. Virtual environments (virtualenv) are a good way of dealing with this. Installation information will be distributed as we go. While not everything has been tried out in advance, we believe things will work on your laptops using the OSX and Linux operating systems, and under the Windows operating system either natively or with Windows Subsystem for Linux (WSL). Apparently the NLTK parts work in Google Colab.

These are some of the toolkits we will use.

NLTK (Natural Language Toolkit) --- Python platform for working with natural language models and data. It includes nice graphics, e.g. for drawing trees, and works in Jupyter notebooks.

HFST (Helsinki Finite State Transducer Technology) --- Toolkit for the Finite State Calculus, which is a language of extended regular expressions including operations of intersection and complement (on top of the usual operations for regular expressions), and operations on relations in addition to sets. It works in notebooks. We use it for Phonology.

Parsers in Python for minimalist grammar and multiple context free grammar from Edward Stabler and Peter Ljunglöf.

Kaldi and PyKaldi --- Toolkit for Hidden Markov modeling of speech signals.

Readings

Natural Language Processing with Python, by Steven Bird, Ewan Klein, and Edward Loper. Online.
Online documentation for NLTK.
Finite State Morphology, by Kenneth R. Beesley and Lauri Karttunen ($40 or less at Amazon or Abebooks). Used for computational phonology.
Lecture notes from Edward Stabler, Computational Linguistics (2013 version).
Lecture notes prepared for the class.

Course mechanics and interaction

Basic mechanics such as announcements and homework submission are via Canvas. We will use Ed to answer questions about the lectures, homework, and the project and other discussion. It is linked through Canvas. Please post your questions there, and answer questions when you can.

Academic Integrity

Your conduct in the course is governed by Cornell policy on academic integrity. A key idea is to attribute any sources or assistance you use. Assignments will state whether group work is permissible. When group work is not permitted, you may not look at solution code or solution text written by another student. It's permissible to discuss matters of interpretation or general strategy. This should preferably be done on the forum. It's permissible to solve a problem by finding the solution with internet searches, in a textbook, or in a technical article. If you do that, cite the source in your submission and say how you used it, but don't post the reference on the forum. In coding problems, it is permissible to use an AI coding system or AI-assisted IDE. If an AI made a substantial contribution to your solution, cite it in your submission and say how you used it. In parts calling for you to write sentences and paragraphs in English, it is not permissible to use AI text generators.

Lectures and course materials are copyrighted, you may not record them or convey them to note-taking services or the like.

Special Accommodations

Please give the instructor any Student Disability Services (SDS) accommodation letter as early as possible so that needed academic accommodations can be arranged. If you need an immediate accommodation, please speak with the instructor after class or email mr249@cornell.edu and/or SDS at sds_cu@cornell.edu. SDS is located on level 5 of Cornell Health, 110 Ho Plaza, 607-254-4545, https://sds.cornell.edu/.

Sketch of problem set topics

PS1 Grammar and tree representations in NLTK, NLTK parsing

PS2 Feature grammars, logical semantics

PS3 Truth checking, preliminary grammars for project, more parsing

PS4 Finite state transducers, finite state generative phonology

PS5 MCFG and MG grammars and parsing (tentatively will be due around April 14)

PS6 HMM, project dry run, possibly OT phonology (tentatively will be due around April 28)