Syllabus

1 Overview

Prerequisites: none.

This course is a practical tour of computational methods that will be useful for practicing linguists. The basics of how text (and possibly audio files) are encoded on computational devices will be discussed, as well as common methods for manipulating these files, extracting their contents, and converting between formats. In addition, modern tools and algorithms for indexation and search, including regular expression grammars and databases, will be covered. Students will learn how to train common tools (e.g., concordances, indices, part of speech taggers, chunkers, and parsers) in a variety of languages, and will learn the basics of statistical inference. The tools, methods, and metrics of modern annotation in computational linguistics will be discussed in considerable depth, including the mechanisms for crowdsourced annotation. After completing this course, students will be able to construct simple programs in Python and Javascript. No background in programming or UNIX will be assumed.

1.1 Course Schedule

The course will be divided into 7 subject areas, with the following order of presentation.

1. Introductions (2 week): Introductions to programming, basic concepts and data structures. Overview of file types and encoding; conversion systems. Basic web scraping.
2. Regular Expressions and Chunking (1 week): Introduction to Regular Expressions; Building chunkers and extractors.
3. Sorting and Searching (2 weeks): Tools for gathering data; methods for compiling data and doing simple queries; concordance building; indexing a corpus
4. Linguistic Annotation Guidelines (2 weeks): Best practices in annotation; metrics for interannotator agreement; using crowdsourcing to gather annotations
5. Advanced Querying (1 week): Introduction to structured query languages, including SQL and tgrep.
6. Interacting with the Web (1 week): CGI scripting; javascript visualization
7. Part of Speech Tagging and Parsing (1 week): Introduction to HMMs; training POS taggers. Introduction to chart parsing; training parsers.

2 Requirements

The course has 4 formal requirements: lecture, section, weekly problem sets, and a final project.

2.1 Lecture

It is imperative that you attend class, especially given that there is **no textbook** for the class. This class will also require diligent and accurate note-taking – please do not rely on class slides/handouts to summarize information for you. Speaking up in class will affect your participation grade.

2.2 Section

Section attendance is **mandatory**, and will affect your participation grade. In section we will discuss the issues raised in class, go over technical details that passed by too quickly, and help with the homework problems. If you cannot attend one of the two sections, you may arrange to visit a TA's office hours. If you cannot do either, you should not take this class.

2.3 Weekly Problem Sets

There will be **7 homework assignments** in this course. While many of the problems will be routine applications of what was discussed in class, several will require imagination, cleverness, and lots of thinking-time.

Homework will be given out on Wednesdays and be due **at the start of class** on the following Wednesday, unless indicated otherwise. **You fail the course if you miss more than 2 assignments.** You should work on homeworks in pairs and submit one joint assignment. Please indicate on your submission who you collaborated with. If you collaborated with no one, please mark it.

We will deduct points for the following:

- not listing your collaborators **I will deduct 30% for this!**

LATE HOMEWORK POLICY: Each student is allowed two late assignments, which must be turned in within one week of the original due date. Late submissions will be given half credit.

2.4 Final Project

There will be a final project due in class on the last day of class. The final project for this course is worth 51% of your grade. The nature of the final project will be assigned by the instructor. Students will work in teams of 2 or 3 to implement a project from a list I will curate. Some examples:

1. A PARSED CORPUS OF IRISH: Students will take Jim McCloskey's existing corpus of Irish examples and produce an indexed parsed corpus as a final project. The final project will include visualization layers to assist in the annotation and examination of the results.
2. IMPLICIT ARGUMENTS IN ENGLISH: Building on the lexicon in NomBank and VerbNet, students will construct a web corpus of English containing annotations for implicit arguments. The final project will include a paper summarizing the trends for which kinds of arguments go implicit and the predicates that permit them.
3. A LEXICAL ONTOLOGY: Students will construct a database-driven system for storing lexical items, arbitrary lists of parameters, and links to corpus examples using a graph database.

2.5 Grade Computation

We will use the following system to compute your grades:

- ▷ Problem sets: 49% (each problem set is worth 7% of your grade)
- ▷ Final Project: 51%