Term Information

Effective Term	Summer 2014

General Information

Course Bulletin Listing/Subject Area	Linguistics
Fiscal Unit/Academic Org	Linguistics - D0566
College/Academic Group	Arts and Sciences
Level/Career	Graduate, Undergraduate
Course Number/Catalog	5050
Course Title	Technical Tools for Linguists
Transcript Abbreviation	Technical Tools
Course Description	Practical training in standard computational tools for tackling different kinds of linguistic research. Students will learn computational techniques to access, search and format linguistic datasets, including text corpora, speech and audio, structured representations such as parse trees, and experimental measurements. The course will also cover data exploration and basic modeling.
Semester Credit Hours/Units	Variable: Min 1 Max 3

Offering Information

Length Of Course	14 Week, 7 Week, 4 Week (May Session), 12 Week (May + Summer)
Flexibly Scheduled Course	Never
Does any section of this course have a distance education component?	No
Grading Basis	Letter Grade
Repeatable	No
Course Components	Laboratory, Lecture
Grade Roster Component	Lecture
Credit Available by Exam	No
Admission Condition Course	No
Off Campus	Never
Campus of Offering	Columbus

Prerequisites and Exclusions

Prerequisites/Corequisites

Exclusions

Cross-Listings

Cross-Listings

Subject/CIP Code

Subject/CIP Code	16.0102
Subsidy Level	Doctoral Course
Intended Rank	Junior, Senior, Masters, Doctoral

Requirement/Elective Designation

The course is an elective (for this or other units) or is a service course for other units

Course Details

Course goals or learning objectives/outcomes

- Students will gain hands-on experience gathering, formatting, and manipulating data.
- Students will learn to use corpus, field, and experimental data, as well as to combine data from multiple sources.
- Students will learn to work with existing computational tools.
- At the end of the course, students will be able to process massive amounts of linguistic data.

Content Topic List

- Accessing and navigating corpora
- Linguistic data manipulation and visualization
- Automatic processing of structured linguistic representations
- R scripting
- Praat scripting

Attachments

- 5050syllabusFinal.pdf: pdf

  (Syllabus. Owner: McGory,Julia Tevis)
- 5050syllabusFinal2.pdf: updated syllabus

  (Syllabus. Owner: McGory,Julia Tevis)

Comments

- New syllabus has been uploaded that reflects suggested changes. Links to texts have been removed in syllabus, and students are directed to OSU bookstores to purchase texts. *(by McGory,Julia Tevis on 09/25/2013 04:17 PM)*
- 7. A list of required texts and other course materials, and information on where they are available (the links didn't work and the third textbook doesn't say where it can be purchased. *(by Heysel,Garett Robert on 09/23/2013 01:58 PM)*

Workflow Information

Status	User(s)	Date/Time	Step
Submitted	McGory,Julia Tevis	09/20/2013 03:50 PM	Submitted for Approval
Approved	McGory,Julia Tevis	09/20/2013 03:51 PM	Unit Approval
Revision Requested	Heysel,Garett Robert	09/23/2013 01:58 PM	College Approval
Submitted	McGory,Julia Tevis	09/25/2013 04:18 PM	Submitted for Approval
Approved	McGory,Julia Tevis	09/25/2013 04:18 PM	Unit Approval
Approved	Heysel,Garett Robert	09/28/2013 06:47 PM	College Approval
Pending Approval	Vankeerbergen,Bernadette Chantal Nolen,Dawn Jenkins,Mary Ellen Bigler Hogle,Danielle Nicole Hanlin,Deborah Kay	09/28/2013 06:47 PM	ASCCAO Approval

Proposed new course: Technical Tools for Linguists
Course Number: Linguistics 5050

Instructor Name: _____
Office: _____
Phone: _____

Meeting Date/Time:
Classroom Location:

Course description:
This course offers practical training in standard computational tools for tackling different kinds of linguistic research. Students will learn computational techniques to access, search and format linguistic datasets, including text corpora, speech and audio, structured representations including parse trees, and experimental measurements. The course will also cover data exploration and basic modeling.

Course goals, learning objectives/outcomes:

1. Students will gain hands-on experience gathering, formatting, and manipulating data.
2. Students will learn to use corpus, field, and experimental data, as well as to combine data from multiple sources.
3. Students will learn to work with existing computational tools.
4. At the end of the course, students will be able to process massive amounts of linguistic data.

The course is designed to stand alone, but also to provide an introduction to the graduate Computational Linguistics sequence. It is not a prerequisite for the Computational Linguistics courses, but is helpful for students who lack any prior experience with computational tools.

Content topic list

- Accessing and navigating corpora
- Linguistic data manipulation and visualization
- Automatic processing of structured linguistic representations
- R scripting
- Praat scripting

Required texts and Course materials Books (*All can be purchased online at Amazon.com or at the OSU bookstores*.)

1. R. H. Baayen. 2008. *Analyzing Linguistic Data: A Practical Introduction to Statistics using R*. Cambridge University Press.

2. Peter Dalgaard. 2008. *Introductory Statistics with R, 2nd edition*. Springer.

3. Keith Johnson. 2008. *Quantitative Methods in Linguistics*. Blackwell.

Tutorials:

- R tutorial [http://www.cyclismo.org/tutorial/R/]
- Python tutorial [http://docs.python.org/3.0/tutorial/]
- Unix for poets [http://ufal.mff.cuni.cz/~hladka/tutorial/UnixforPoets.pdf]

Assignments

The assignments are designed to be relevant to linguistic research, and will be connected to the work and interest of the students. There will be 5 assignments, one for each unit in the class. The assignments are described in detail above; each one will require students to write a short program to perform some analysis of a dataset (for instance, assignment 1 is to write a Python program measuring utterance length by men and women in a section of the Fisher corpus). Students will work on the assignments both in class and at home, and will be encouraged to work collaboratively in small groups, but everyone must turn in his/her own separately written assignment.

Syllabus

[weeks 1-3] Unit 1: Basic data manipulations

- Introduction and motivation
  - Case study: **Do women talk more than men?**
  - How to use dialogue corpora to test a hypothesis
- Basic Unix environment
  - How to access and navigate our corpora directories
- A computer language to deal with human language: Introductory Python
  - Basic file IO
  - Decision-making: logic, comparatives, conditionals

Week 1 will discuss the basic goals and assumptions of data-driven corpus linguistics, giving a broad overview of what language corpora are available, how they are created, where to find them and what problems they might be helpful in solving. We will introduce the Fisher dialogue corpus and instruct students in the tools necessary to find, view and search files from Fisher by hand using the standard Unix environment. Students will do several in-class exercises designed to build competence with the Unix command line tool set, and will conduct a small corpus investigation of whether women or men talk more by hand-analyzing Fisher data.

Weeks 2-3 will introduce Python as a tool for programmatically reading and searching large amounts of data. We will explain Python's basic representations of numbers and strings. Students will gain experience using Python as a calculator and searching for words or phrases from the interactive Python shell. Students will learn the basic constructs of imperative programming (loops and conditionals).

Assignment 1 Students will have to write a Python program to decide the women-vs-men problem by counting utterance length in a gender-annotated dialogue corpus. Students

will provide a set of statistics, coming out of their Python program, to answer the question of whether women talk more than men.

[weeks 4-6] Unit 2: Reading text and counting words
- Case study: **Investigating Zipf's law**
- Counting instances of a word in a file
- Dictionaries
- Counting all words/bigrams

Weeks 4-6 will focus on structured data types (lists and dictionaries). Students will learn to store and manage data, to develop appropriate data models for simple structured problems, and to write simple procedures for processing stored data.

Assignment 2 Students will write Python programs for counting unigrams and bigrams that can verify the statistical pattern of word frequencies known as Zipf's Law.

[weeks 7-10] Unit 3: Dealing with linguistic structured representations
- Case study: **Which verbs appear most often in the passive construction?**
- Field-structured: CSV, space-delimited
- Tree-structured: Penn TreeBank parses
- Structures with internal references: CoNLL parses
- XML parsing
- Internationalization, non-standard character sets:
    How to deal with Arabic, Chinese, Hindi or Cyrillic?

Weeks 7-10 will extend the ability to use data structures to familiarize students with the use of external libraries for reading and manipulating common data formats. We will introduce the NLTK libraries for reading parse trees. Students will gain expertise at searching for and understanding documentation.

Assignment 3 Students will build Python word counting programs for several formatted datasets, using pre-developed libraries to read the data into appropriate representations and computing lists of verbs which tend to appear in passive constructions. If time allows, students will compare the suitableness of constituency and dependency trees as representations for such a study.

[weeks 11-12] Unit 4: R scripting
- Case study: **Dative alternation: how do children differ from adults?**
- The R language:
    - Variables, control statements and data structures in R
    - Data exploration and visualization

Students will learn the basics of data storage and manipulation in R, applying their understanding of Python programming to "translate" basic concepts into a new language.

Assignment 4 Students will be to write a statistical analysis of the provided dataset using

the language and create a visualization of the results.

[weeks 13-14] Unit 5: Praat scripting
- Case study: **Automatically extracting measurements to make vowel space plots**
- The Praat language
    - Variables, control statements and data structures in Praat
    - Manipulation of audio data

During the last two weeks, students will learn the basics of Praat scripting, applying their understanding of Python programming to "translate" basic concepts into a new language.

Assignment 5 Students will write a program using Praat's built-in routines to analyze an audio dataset and create a vowel plot, as commonly done in phonetic/sociophonetic analyses.

Class meetings
These will be hands-on, problem-oriented sessions. Students are strongly encouraged to bring a laptop, so that they can investigate and solve problems during in-class activities.

Grade components
- Assignments (90%) will be assigned at the end of each unit (every 2 or 3 weeks) and students will have one week to complete them. There will be five assignments in total. Each is worth 18% of the final grade.

- Attendance (10%) is included as part of the final grade. "Attending" means coming to class, paying attention, verbally participating in class discussions and completing in-class exercises.

Grading Scale: Standard OSU grading scheme

A	93–100	A-	90–92		
B+	87–89	B	83–86	B-	80–82
C+	77–79	C	73–76	C-	70–72
D+	67–69	D	60–66		
E	0–59				

Requirements
The class can be taken for 1 or 3 credits:

- If students are taking the course for 1 credit, they are expected to attend all classes, participate in all class activities, complete all assigned readings and show evidence of having completed the readings during class discussion.

- If students are taking the course for 3 credits, they are expected to attend all classes, participate in all class activities, complete all assigned readings, show evidence of having completed the readings during class discussion, and complete all assignments.

Academic Misconduct
It is the responsibility of the Committee on Academic Misconduct to investigate or establish procedures for the investigation of all reported cases of student academic misconduct. The term "academic misconduct" includes all forms of student academic misconduct wherever committed; illustrated by, but not limited to, cases of plagiarism and dishonest practices in connection with examinations. Instructors shall report all instances of alleged academic misconduct to the committee (Faculty Rule 3335-5-487). For additional information, see the Code of Student Conduct.

Students with disabilities that have been certified by the Office for Disability Services will be appropriately accommodated and should inform the instructor as soon as possible of their needs. The Office for Disability Services is located in 150 Pomerene Hall, 1760 Neil Avenue; telephone 292-3307, TDD 292-0901.