Syllabus for LIN 637

Computational Linguistics 2 – Spring 2019

N250 SBS, TTh 8:30-9:50

January 28, 2019

(page 1 of 4)

Instructor	**Office Locations**
Jeffrey Heinz `jeffrey.heinz@stonybrook.edu`	N237 SBS L160 IACS
Office Hours MW 2:30-4:00pm and by appointment	

`https://github.com/Compling2-Spring2019-SBU/main`

1 Course Outline

1.1 Bulletin Description

An introduction to the theoretical foundation of computational linguistics. The course emphasizes the importance of algorithms, algebra, logic, and formal language theory in the development of new tools and software applications. Empirical phenomena in phonology and syntax are sampled from a variety of languages to motivate and illustrate the use of concepts such as strictly local string languages, tree transducers, and semirings. Students will develop familiarity with the literature and tools of the field.

1.2 Full Description

This course acts as the bridge from introductory courses in linguistics (Syntax 1, Phonology 1, Phonetics) and computational methods (Statistics, Mathematical Methods in Linguistics, Computational Linguistics 1) to advanced courses and seminars in computational/ mathematical linguistics. In contrast to the NLP courses offered by the department of computer science, this course focuses on studying the properties of natural language from a computationally informed perspective. The question is not how computers can solve language-related tasks, but how language can be conceptualized as a computational problem. This emphasis is also reflected in the selection of topics for this course.

1.3 List of topics

- *Phonology and Morphology*

- The role of formalization
- String languages
- Regular languages
- String transductions
- Subregular classes
- Computational perspectives on phonological theories like SPE and OT

- *Syntax*
  - Tree languages
  - Syntax is more complex than phonology
  - Mildly context-sensitive formalisms (TAG, MGs)
  - Tree transductions
  - Regular representations of MCS formalisms

1.4 Prerequisites

The only official prerequisite is Computational Linguistics 1 (Lin 537) or comparable programming skills in Python. Python will be used to illustrate formal concepts, and some of the homeworks will require you to implement an algorithm or procedure in Python. Prior experience with git and markdown is useful for the homeworks but not required.

It is also helpful to have some basic familiarity with linguistics (phonemes, phrase structure rules, syntactic trees) and mathematics (sets, functions, relations, and first-order logic as covered in Semantics 1, for instance).

2 Grading

- **Homework**
  - Exercises, programming assignments, or critical evaluations of assigned readings
  - Homework submission and grading is done via github.
  - No late hand-ins!
  - Collaboration on homework problems is encouraged as long as you write up the solutions by yourself, using your own words, examples, notation, and code.
  - 2/3 of final grade

- **Project**
  - You will develop a final project in this class.
  - Projects include theoretical research, program/software/tool development, or a software-aided study of language data.

- A one page project proposal is due Thursday March 14.
- The project itself is due on Thursday May 16.
- 1/3 of final grade

- **Readings**
  - At most two readings per week
  - It is presupposed in the lectures that you have done the required readings.
- **Feedback on Lecture Notes**
  - Thomas plans to publish these lecture notes as an open-access textbook with *Language Science Press*.
  - Every week, you should upload to Github feedback for the relevant units, where you spot typos, suggest exercises, pictures, examples, and so on.
  - Feel free to comment on other student's suggestions if you (dis)approve, expand their ideas, and so on.
- **Workload per Credits**
  - *0 credits*: none, but I highly recommend that you at least read the assigned papers as they will be important for following the lectures
  - *1 credit*: attend
  - *2 credits*: attend, readings
  - *3 credits*: attend, readings, homework & project

3 Online Component

This class uses some online tools to facilitate homework collaboration and submission, student discussions, and dynamic lecture evaluation.

- **Homework submission**
  How it works: Homeworks will distributed via a github repository. You can fork this repo and upload your own code, or checkout other students' forks to see how they dealt with the problem. In order to submit a homework you upload your solution to your fork and issue a pull request. After the due date, I'll upload my solution to the repository.

  Why we do it: This setup mimics the modern workflow in collaborative development projects. Git is one of the best-known version control systems, and github is the biggest online service for hosting git repositories. Familiarity with version control systems is an essential job requirement for computational linguists, and it is also very helpful for academic work. See this discussion on Stackflow for some ideas how git can be used in conjunction with Latex: http://stackoverflow.com/questions/6188780/git-latex-workflow

What you'll need: A github account (the free tier is enough) and a way of uploading your code to a github repository. Linux users will install git via the command line, whereas Windows and Mac users should download and insatall the github app, which comes with a nice GUI.

4 University Policies and Services

Student Accessibility Support Center (SASC) Statement: If you have a physical, psychological, medical or learning disability that may impact your course work, please contact the Student Accessibility Support Center (SASC), ECC (Educational Communications Center) Building, room 128, (631) 632-6748. They will determine with you what accommodations, if any, are necessary and appropriate. All information and documentation is confidential.

Students who require assistance during emergency evacuation are encouraged to discuss their needs with their professors and the staff at the Student Accessibility Support Center (SASC). For procedures and information go to the following website: `http://www.stonybrook.edu/ehs/fire/disabilities`.

Academic Integrity Statement: Each student must pursue his or her academic goals honestly and be personally accountable for all submitted work. Representing another person's work as your own is always wrong. Faculty are required to report any suspected instances of academic dishonesty to the Academic Judiciary. Faculty in the Health Sciences Center (School of Health Technology & Management, Nursing, Social Welfare, Dental Medicine) and School of Medicine are required to follow their school-specific procedures. For more comprehensive information on academic integrity, including categories of academic dishonesty, please refer to the academic judiciary website at `http://www.stonybrook.edu/commcms/academic_integrity/index.html`

Critical Incident Management Statement: Stony Brook University expects students to respect the rights, privileges, and property of other people. Faculty are required to report to the Office of Judicial Affairs any disruptive behavior that interrupts their ability to teach, compromises the safety of the learning environment, or inhibits students' ability to learn. Faculty in the HSC Schools and the School of Medicine are required to follow their school-specific procedures.