RUTGERS
School of Arts and Sciences

Computational Linguistics (01:615:455) **Fall 2021**

Lecture	Tues and Fri, 11:00am–12:20pm (on Zoom)
Course website	`canvas.rutgers.edu` (notifications will be sent to your **Rutgers email**)
Instructor	Simon Charlow
Email	`simon.charlow@rutgers.edu`
Student support hours	Weds and Fri, 1–2pm (**I'm also available by appointment**)

Lectures and student support hours will be conducted using Zoom. Lectures will be recorded and posted to Canvas. Recording by anybody other than me is prohibited, and recordings may not be distributed. You should use your real name as your Zoom name, and you're encouraged but not required to have your camera on.

About this course

Computational linguistics is a large and vital field comprised of people interested in language from a variety of perspectives, together with technologies they've developed for working with language. Computational approaches to language power industrial applications, as well as research into the human faculty for language.

This course will introduce you to some of the fundamental concepts and techniques of computational linguistics. Computation is like language in some ways: it's practically bounded in space, time, and memory, yet nonetheless capable of doing an infinite variety of things, with this behavior specified, constrained, and ultimately enabled by finite, rule-governed systems. Learning about these connections has practical and scientific benefits. On one hand, it helps us gain a deeper understanding of the computational nature of language, and of linguistic creativity/generativity. On the other hand, once a task is specified coherently and in sufficient detail, computers can perform boring, rote, and big tasks extremely quickly, enabling us to test theories quickly at scale, and helping us to clarify and sharpen our thinking.

Course learning goals

At the completion of this course, students will be able to:

- Understand and design computational tools for generating and manipulating natural language
- Understand and apply computational techniques to the analysis of empirical phenomena in different areas of linguistics (morphology, phonology, semantics, syntax)
- Characterize the computational properties and power of different aspects of our linguistic competence

Department learning goals met by this course:

- Students will reason about language; identify how incorrect or irrational assumptions and prejudices distort understanding of language; demonstrate knowledge about language in the world including a sophisticated understanding of linguistic and cultural variation, and evaluate popular views on the nature of human languages and their speakers.
- Majors and Minors will also demonstrate technical mastery over the tools of linguistic analysis in syntax, phonology and semantics and apply linguistic theory in these areas. They will investigate linguistic data and analyze it; demonstrate strong problem-solving skills; extend their understanding of theoretical linguistics into other domains of linguistic research; apply the techniques of linguistics that they have learned in the core courses to new topics; and access current research in the field. Some students will investigate language in a broader context, where it can be systematically and rationally explored using their sophisticated understanding how language works.

Week	Dates	Topic	Notes
1	09/03	Introduction: installing the Haskell platform	
2	09/07, 09/10	Programming: types, variables, substitution, lists, recursion	
3	09/14, 09/17	Basic corpus methods and text processing	
4	09/21, 09/24	*n*-gram models	
5	09/28, 10/01	Regular expressions (regexps)	
6	10/05, 10/08	Regexps (cont.)	10/05 is async
7	10/12, 10/15	Intro. to finite-state automata (FSAs)	
8	10/19, 10/22	FSAs ↔ regexps	
9	10/26, 10/29	Generalizing: FS transducers, weighted FSAs, semantic automata	
10	11/02, 11/05	Intro. to context-free grammars (CFGs): motivation, generation	
11	11/09, 11/12	Parsing with CFGs	Proposals due 11/12
12	11/16, 11/19	Parsing with CFGs (cont.)	
13	11/23	Additional topics: Learning? Context-sensitivity?	
14	11/30, 12/03	Additional topics (cont.), begin mini-project presentations	
15	12/07, 12/10	Mini-project presentations (cont.)	Projects due 12/17

Table 1: A tentative schedule. Preliminary and subject to change, depending on our speed, and your interests.

Programming

This course's written work involves writing and modifying computer programs. I don't presuppose that you have any prior experience with programming, or any specific mathematical background going beyond what you've learned in the prerequisite 615:201, 'Introduction to Linguistic Theory'.

The programming language we'll use is Haskell. **As soon as possible, download and install the Haskell platform: `https://haskell.org/platform`**.[1]

Programming is a tool that helps (or, depending on your perspective, forces) us to clarify and sharpen our understanding of the key concepts and techniques of the course. (It also helps us automate stuff that we'd rather not do, or that would take us too long to practically do.) Our focus in this course is on these concepts and techniques, and not on the finer points of our particular programming language. In other words, this course isn't going to turn you into an industrial-strength Haskell hacker (though it will give you a solid start, if that's something you're interested in). Similarly, you won't need to consult outside resources on Haskell (beyond the assigned reading), and probably shouldn't plan to: my goal is to give you everything you need, stripping out all the 'extra' stuff which would make your task more arduous and daunting than it needs to be.

If you eventually want to dive deeper (and I hope you do), Graham Hutton's *Programming in Haskell* (Cambridge University Press, 2nd edition, 2016) is a well written introductory text, and a great resource/reference.

Readings

Readings will be posted to Canvas as pdfs. You do not need to buy a textbook for this course. You do need to do the reading, in advance of the class for which it's assigned. It will be extremely easy to fall behind in the course if you do not stay current with the reading. Fortunately, keeping up with the reading will be very manageable.

Evaluation and required work

Grades are based on regular (≈weekly) coding assignments, a final mini-project, and class involvement. There is no final exam. The percentage-wise breakdown is on the left, and the corresponding letter grades are on the right:

[1] Note that this involves some use of the command line. If this is totally unfamiliar to you, don't worry — we'll go over it. In the meantime, you might find it useful to track down and review a tutorial on your operating system's command prompt.

Homework	**75%**
Final mini-project	**20%**
▷ Proposal	5%
▷ Presentation	5%
▷ Project	10%
Class involvement	**5%**

A	⩾90%
B+	85–89.9%
B	80–84.9%
C+	75–79.9%
C	70–74.9%
D	60–69.9%
F	⩽59.9%

The mini-project is not supposed to be huge: roughly the ‘size’ of two homework assignments is a reasonable way to think about what I have in mind. There are two options for the final mini-project:

- A program written in Haskell that builds on one of the techniques that we learned in the class.
- A research paper of ⩾5 pages on some issue in computational linguistics, either theoretical or applied.

The mini-project has three parts: a proposal of 250–500 words (Week 11), a short presentation of 5–10 minutes (Weeks 14 & 15), and the project itself (end of the course). See the Table 1 schedule for precise dates. We’ll discuss the mini-project more around the middle of the term, and I’ll offer some suggestions for possible topics.

Please note the following policies on late work:

- Late assignments receive an automatic 15% reduction.
- Assignments more than 24 hours late will not be accepted for credit.

Class involvement includes attendance, participation in discussions (in class and/or on Canvas), and evidence of reading. You’ll earn all of these points if I can see that you’re engaged with the course.

Getting help

There will be a Canvas forum for discussions and questions about the content of lectures, readings, and the homework. You are especially encouraged to post questions there when the answer might benefit other participants in the class (it goes without saying, but you are also welcome to email me, and to drop into student support hours, no appointment needed). Of course, you must not reveal the answers to any homework questions on the forum (see also the ‘Academic integrity’ section below for information on how to work with your classmates on assignments).

The material in this course, and the skills you’ll develop, are likely to seem challenging at times, at least initially. But I’m very committed to helping you succeed. If you keep up with the reading, come to class and participate, give yourself sufficient time with the assignments, and avail yourself of resources to help you get unstuck if/when you do get stuck, you’ll be in great shape.

Academic integrity

Please review the university’s academic integrity policy: `academicintegrity.rutgers.edu`. It’s unethical and unacceptable to pass off anyone else’s work as your own. All suspected violations will be reported to the Office of Student Conduct.

Discussion of homework with your classmates may be helpful, and is encouraged, but the work that you hand in needs to be a product of, and an reflection of, your individual understanding. When you discuss work with others in the course, you may do so at a whiteboard (or something equivalent). Once the discussion’s done, erase the board; don’t take any notes with you; and write up your submission on your own. Your ability to do so demonstrates that you understand what you hand in, and can explicate it if need be.

All course materials are copyrighted by me, or by their respective authors. It is unacceptable to share or post any course materials in any venue without explicit permission from me.

Attendance

Attendance is required and is recorded automatically by Zoom. Frequent unexcused absences will make it difficult for you to succeed in the course, and will cause you to lose Class involvement points. If you expect to miss one or two classes, **please use the University absence reporting website** (`sims.rutgers.edu/ssra`) to indicate the date and reason for your absence. An email will automatically be sent to me.

Counseling, ADAP & Psychiatric Services (CAPS)

CAPS is a University mental health support service that includes counseling, alcohol and other drug assistance, and psychiatric services staffed by a team of professionals within Rutgers Health services to support students' efforts to succeed at Rutgers University. CAPS offers a variety of services that include: individual therapy, group therapy and workshops, crisis intervention, referral to specialists in the community, and consultation and collaboration with campus partners. Visit `http://health.rutgers.edu/medical-counseling-services/counseling/`.

Disability services

Rutgers University welcomes students with disabilities into all of the University's educational programs. In order to receive consideration for reasonable accommodations, a student with a disability must contact the appropriate disability services office at the campus where you are officially enrolled, participate in an intake interview, and provide documentation: `https://ods.rutgers.edu/students/documentation-guidelines`. If the documentation supports your request for reasonable accommodations, your campus's disability services office will provide you with a Letter of Accommodations. Please share this letter with your instructors and discuss the accommodations with them as early in your courses as possible. To begin this process, please complete the registration form: `https://webapps.rutgers.edu/student-ods/forms/registration`.

General policies

The following policies are provided at `ling.rutgers.edu` > Undergraduate > Department Learning Goals and Policies (a direct link is here):

- Rutgers Code of Student Conduct
- Policy on Religiously Observant Students
- Students with Disabilities
- Attendance Policy
- Academic Integrity
- Complaints
- Canvas Policy
- Email and Contact Information Policy

It is the responsibility of all students to read and abide by the policies above.