Language and Computation

(Preliminary) syllabus for LING 227.01 / PSYC327.01 / LING627.01, Yale University, spring 2014

Course and contact information

Meeting:	Tuesday and Thursday 4.00-5.15, in *William L. Harkness Hall* (WLH) 113.
Instructor:	Tamás Biró
Email:	tamas.biro@yale.edu (backup: birot@birot.hu)
Office:	370 Temple Street, room 302
Office hours:	Thursdays, 5.30–6.00, or by appointment
Websites:	Classes*v2, and backup: http://www.birot.hu/courses/2014-LC/
Teaching Assistant:	TBA

Description

Computational linguistics is the study of natural language from a computational perspective. It encompasses both applied (engineering) and theoretical (cognitive) issues, ranging from speech and language technology to formal aspects of theoretical linguistic models. Have you ever wondered how your navigation system can pronounce street names? Have you been surprised by the capabilities of and mistakes made by translation systems? Interested by the "language software" in your brain? Ever wished to automatically process large corpora, and discover linguistic structures therein? Do you want to move beyond pen-and-paper speculations about the learnability of your favorite theoretical linguistic framework? Then *Language and Computation* is for you!

This course provides a taste of the various fields in this extremely broad and bourgeoning discipline, without aspiring to being all-embracing. By the end of the course, you will have

1. a *better understanding* of the role of computing in language, and of language in computing;
2. a *novel perspective* on language – which is close, but not identical to the perspective of the theoretical linguist – and on the complexities of processing it by the mind or by a computer;
3. an *understanding* of the computational problems arising at various levels of language: phonetics and phonology, morphology, syntax, semantics and discourse, as well as
4. in various applications, such as speech synthesis and speech recognition, part-of-speech tagging, information extraction, question answering, dialogue systems and machine translation;
5. *familiarity* with basic concepts in natural language processing (formal language theory, probabilistic models, parsing, machine learning, precision and recall…), together with
6. some *acquaintance* with the most important approaches and algorithms, as well as
7. the *skills* to learn in the future about more on advanced and contemporary topics; and finally,
8. the basic *skills* needed to implement these algorithms yourself on a computer.

Form

The course meets twice a week for an (informal) **lecture**. The goal of the lectures is twofold: to ease the understanding of the **reading**, and to complement it. Readings are assigned in two forms: **pre-readings** must be completed before lectures, and each lecture will presuppose familiarity with the pre-reading assigned, while **post-reading** will include material either overlapping with the lecture or complementing it. Studying the post-reading is recommended immediately after class, but must be done by the exam.

Optional sections for consultation are offered by the TA on Mondays, between 1:30-2:20 and 2:30-3:20.

Python programming: Programming is not the focus of this course, but knowing how to program is an essential skill needed to do computational linguistics. In fact, knowing how to program is necessary in order to understand the computational linguist's perspective, research questions and answers.

Following most of the course only requires comprehending theoretical concepts, mathematical formulae and pseudo-code, which will be facilitated by the lectures. Yet, in order to improve the efficacy of the learning process, and to acquire practical skills, some of the homework assignments will involve programming. Using the *Python* programming language will be highly recommended for practical reasons. Moreover, the course will also include a survey of *Python Natural Language Toolkit* (NLTK).

Those unfamiliar with *Python* are offered two solutions:

1. If you already have some prior programming experience (for instance in C, Perl or Java), then you are encouraged to learn the basics of Python by yourself during the first third of the semester. No advanced topics in Python are required. Students are encouraged to use *Magnus Lie Hetland*'s book (see below), and the exact chapters to be worked through will be provided each week.
2. **Optional programming sessions** may be offered, if necessary, in the first third of the semester by the instructor to those lacking any programming experience. Priority will be given to graduate and advanced undergrad students in linguistics. The suggested meeting time for these sessions is *Wednesday, 4:30-6:00*, and we shall work through a significant part of *Magnus Lie Hetland*'s book.

Requirements

Class participation (10%): Students are expected to actively participate in the discussions in class.

Homework assignments (40%): There will be approximately seven assignments, on a weekly-biweekly basis, in order to deepen your understanding of the material, acquire the necessary skills, and providing you with hands-on experience using computational techniques.

Midterm exam (20%): The take-home exam assigned early March will focus more on concepts and theory, but also to some degree on practical skills. Please note that the exams are not only meant to measure your knowledge acquired so far, but are also part and parcel of the learning process.

Final exam (30%): An in-class, on-paper, open-book exam – focusing more on concepts and theory, but partially also on the practical skills acquired during the semester – is scheduled on **Fr May 2 at 9.00**.

Term paper (grad students only): Graduate students will additionally work on a project in some area related to computational linguistics, with a short term paper (10-15 pages) describing the project and due **April 30**. The topic should include some programming, but is otherwise fairly flexible, and it is ideally related to other interests of yours. I encourage you to talk to me about your project ideas as early as possible, even if it is very vague. A 1-2 page prospectus will be due **March 10**.

Policies

Late assignments will only be accepted in exceptional cases. Collaboration is not permitted in the case of the exams. In the case of the homework, you must program alone (otherwise you will not learn how to program), and write up your solutions alone. It is permitted (1) to discuss the questions with each other, (2) to ask each other for help if you get stuck in programming (especially in debugging), and (3) to compare the outputs of the programs to make sure you have no bugs. In such cases, however, you always must list the name of your fellow student(s) helping you on your assignment.

Academic honesty

Yale does not tolerate plagiarism, and Yale policy will be fully enforced. For more information, refer to *http://yalecollege.yale.edu/content/cheating-plagiarism-and-documentation*. Useful resources on citing include *http://writing.yalecollege.yale.edu/using-sources* and *http://writing.yalecollege.yale.edu/advice-students*. Please feel free to consult the lecturer in case of doubts.

Textbook(s)

The main textbook should be available in the bookstore, but you may also find it (cheaper?) online:

> Jurafsky, Daniel, and James H. Martin (2008). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*. 2nd edition. Prentice Hall. ISBN: 0131873210.

Make sure you buy the second edition. Within the second edition, the hardcover "US edition" is preferred, but you can also buy the "international edition" (which might have a couple fewer typos). The two differ in page numbers (but not in section numbers), and also have about half different exercises. Although we will not cover the entire text, you will find it a valuable reference book even on the long term.

As for Python, I recommend the following book for self-study, which we shall also use during the optional programming sessions:

> Hetland, Magnus Lie (2008). *Beginning Python: From Novice toPprofessional.* 2nd edition. Dreamtech Press. ISBN: 1590599829.

Although this book is available online to the Yale community through the Yale Library, it is probably a good investment if you have no other Python book. Further chapters and articles to be read will be made available on the website of the course (http://www.birot.hu/courses/2014-LC/) and on Classes v2.

Tentative Schedule

Week 1:	Tu, 01/14.:	Introduction	Th, 01/16.:	Language as computation
Week 2:	Tu, 01/21.:	Regular expressions for NLP	Th, 01/23.:	Words, transducers, edit distance
Week 3:	Tu, 01/28.:	N-grams, bags of words	Th, 01/30.:	Part-of-speech tagging (1)
Week 4:	Tu, 02/04.:	Automata	Th, 02/06.:	Probability, Markov models, POS (2)
Week 5:	Tu, 02/11.:	Speech synthesis	Th, 02/13.:	Speech recognition
Week 6:	Tu, 02/18.:	Machine learning (intro)	Th, 02/20.:	Machine learning (examples)
Week 7:	Tu, 02/25.:	Computational syntax, CFG	Th, 02/27.:	Parsing (non-probabilistic)
Week 8:	Tu, 03/04.:	Parsing (probabilistic)	Th, 03/06.:	Python NLTK toolkit
Prospectus due: **March 10 (graduate students).**			***Midterm take-home*****, due: March 25.**	
Week 9:	Tu, 03/25.:	Computational phonology intro	Th, 03/27.:	Finite-state phonology
Week 10:	Tu, 04/01.:	implementing Optimality Theory	Th, 04/03.:	Learning Optimality Theory
Week 11:	Tu, 04/08.:	Computational semantics	Th, 04/10.:	Computational discourse
Week 12:	Tu, 04/15.:	Applications (IE, QA)	Th, 04/17.:	Applications (dialogue systems)
Week 13:	Tu, 04/22.:	Machine Translation	Th, 04/24.:	Summary
Term paper due: **April 30 (graduate students).**			**Final exam: Friday, May 2, 9.00 am.**	