

[Home](#) | [Course Info](#) | [Schedule](#) | [References](#) | [FAQ](#) | [Misc.](#) | [HuskyCT](#)

Ling 3798: Introduction to Computational Linguistics

Fall, 2013

Oak Hall 308

MWF 9:05-9:55 AM

Instructor: Stefan Kaufmann

Email: stefan.kaufmann@uconn.edu

Office: Oak Hall, Rm. 364

Phone: 860-486-8123

Office Hours: Wed 10:30-12:00
and by appointment

Last updated: October 07, 2013

[Home](#) | [Course Info](#) | [Schedule](#) | [References](#) | [FAQ](#) | [Misc.](#) | [HuskyCT](#)

Ling 3798: Introduction to Computational Linguistics

General Information

Course Description: This course is an introduction to computational methods in empirical linguistic analysis and natural language processing. Topics include the use of text corpora and other sources of linguistic data; morphological analysis, parsing and language modeling; applications in areas such as information retrieval and machine translation. The main objective is to familiarize students with core questions and approaches in the field. Theoretical material on such topics as formal languages, automata and complexity, finite-state and context-free methods, n -grams etc. will be supplemented with practical exercises and mini-projects to give students some hands-on experience in the use of corpora and the implementation of algorithms.

Programming skills are not required, but are a plus. Students' projects may be scaled to their level of expertise. Some background in linguistics is definitely an advantage.

Registration: Students should register for LING 3798, listed with the title "Variable Topics." (The course is a new addition to the Linguistics offerings and does not yet have its own course number.)

Prerequisites: At least one course in Linguistics, or permission of the instructor.

Teaching method: Lectures, discussions, in-class exercises, homeworks, programming projects.

Evaluation: Six take-home assignments (60%); final exam (30%); participation (10%). Substitution of individual programming project(s) for some of the homeworks can be negotiated. The format of the final exam will be announced later in the semester.

Readings: Jurafsky, D. and J.H. Martin. 2009. *Speech and Language Processing*. 2nd edition. Prentice Hall.

This is a good book to own, but students are not required to buy it, since we will only use excerpts from it and supplement them with other readings where appropriate.

Notice to students with disabilities: In compliance with Section 504 of the 1973 Rehabilitation Act and the Americans with Disabilities Act, UConn is committed to providing equal access to all programming. Students with disabilities seeking accommodations are encouraged to contact the [Center for Students with Disabilities \(CSD\)](#). CSD is located in Wilbur Cross Building, Room 224. Additionally, I am available to discuss disability-related needs during my office hours or by appointment.

Last updated: August 30, 2013

[Home](#) | [Course Info](#) | [Schedule](#) | [References](#) | [FAQ](#) | [Misc.](#) | [HuskyCT](#)

Ling 3798: Introduction to Computational Linguistics

Schedule

Note: This schedule is subject to change, but not without notice. Any changes will be announced in class and reflected on these pages. Be sure to check regularly.

Dates	Topics	Methods	Readings
Aug 26 [Mo]	<i>Introduction</i> Computers in linguistics and Natural Language Processing The nature and use of text corpora	Some programming basics	Jurafsky & Martin Ch. 1
Aug 28, 30 [We, Fr]	<i>Regular expressions</i> Pattern matching Corpus search and counting	File handling, control structures	J&M Ch. 2.1
Sep 2 [Mo]	<i>No class (Labor Day)</i>		
Sep 4,6 [We,Fr]	<i>Regular languages</i> Finite-state automata Operations and closure properties Pumping Lemma	Set theory	J&M Ch. 2.2
Sep 9-13 [Mo-Fr]	<i>Finite-state linguistics</i> Transducers Morphological analysis	NLTK	J&M Ch. 3
Sep 16-20 [Mo-Fr]	<i>N-grams</i> Language modeling Smoothing Evaluation	Probability theory NLTK	J&M Ch. 4
Sep 23-27	<i>Part-of-Speech Tagging</i>	NLTK	J&M Ch. 5

[Mo-Fr]	Word classes and tagsets Rule-based and stochastic POS tagging Hidden Markov Models Evaluation		
Sep 30-Oct 4 [Mo-Fr]	<i>Further topics in sequence linguistics</i> Maximum entropy models	Information theory Programming exercises	J&M Ch. 6
Oct 7-11 [Mo-Fr]	<i>Context-free languages</i> Syntactic ambiguity Context-free grammars Push-down automata Pumping lemmas Chomsky Hierarchy		J&M Ch. 12, 16
Oct 14-21 [Mo-Mo]	<i>Parsing</i> Search and dynamic programming Chart parsing (CKY, Earley) Chunking Evaluation	NLTK	J&M Ch. 13
Oct 23, 25 [We, Fr]	<i>No class</i>		
Oct 28-Nov 1 [Mo-Fr]	<i>Further topics in tree linguistics</i> Probabilistic context-free grammars Statistical parsing	Treebanks; <code>tgrep</code>	J&M Ch. 14 Tgrep User Manual
Nov 4, 6 [Mo, We]	<i>Meaning</i> Some approaches and desiderata Syntax-semantics interface Compositionality	Logic	J&M Ch. 17
Nov 8 [Fr]	<i>No class</i>		
Nov 11-15 [Mo-Fr]	<i>Word meaning</i> Semantic ambiguity Semantic relations Semantic roles	WordNet, FrameNet NLTK	J&M Ch. 19
Nov 18-22	<i>Computational lexical semantics</i>	NLTK, <code>infomap</code>	J&M Ch. 20

[Mo-Fr]	(Un)supervised word sense disambiguation Classifiers Vector-space semantics
Nov 25-29 [Mo-Fr]	<i>Thanksgiving break</i>
Dec 2-6 [Mo-Fr]	<i>TBA</i>
Dec 9 - 13 [Mo-Fr]	<i>Finals week</i> Our slot: Wed, Dec 11, 8-10am

[Home](#) | [Course Info](#) | [Schedule](#) | [References](#) | [FAQ](#) | [Misc.](#) | [HuskyCT](#)

Ling 3798: Introduction to Computational Linguistics

Readings

- Bird, S., E. Klein, and E. Loper. 2009. *Natural Language Processing with Python*. O'Reilly.
- Dickinson, M., C. Brew, and D. Meurers. 2012. *Language and Computers*. Wiley-Blackwell.
- Johnson, K. 2008. *Quantitative Methods in Linguistics*. Blackwell Publishing.
- Jurafsky, D. and J.H. Martin. 2009. *Speech and Language Processing*. 2nd edition. Prentice Hall.
- Lutz, M. and D. Ascher. 2004. *Learning Python*. 2nd edition. O'Reilly.

Links

Python

- [Python Standard Library](#)
- [Try Python Simulator](#)
- [Visual Python Tutor](#)

NLTK

- [NLTK Home](#)
- [NLTK Book](#)

Regular expressions

- Grant Skinner's [RegExr](#) testing tool
- Steven Livithan's [RegexPal](#) testing tool
- [Python Regular Expressio HOWTO](#)
- [Python Regular Expression Syntax](#)
- [The Odds & Ends of Regular Expressions](#) (Python Central)

Finite-state machines

- [OpenFst Library](#)
- [Stuttgart Finite State Transducer](#)
- [J. Novak's slides on operations on weighted FSTs](#)

POS Taggers

- [Xerox FST POS tagger online demo](#)

Last updated: October 07, 2013

- [How do I tell IDLE where to find my files?](#)
 - [How do I scroll through the command history in IDLE?](#)
-

How do I make IDLE find my files?

You need to keep your files (e.g., data and Python scripts) in directory in which you can permanently store things. The P: drive is a good option (it's automatically mounted when you log in with your NetID); alternatively, you could use a USB stick, or a cloud-based system like Dropbox or Google Drive or the like. For instance, if I store the file `moby.txt` in my Dropbox directory and start IDLE from the Windows Startup menu, I get an error message:

```
>>> myfile = open('moby.txt', 'r')

Traceback (most recent call last):
File "", line 1, in
myfile = open('moby.txt', 'r')
IOError: [Errno 2] No such file or directory: 'moby.txt'
>>>
```

Basically there are two options:

1. Change the current working directory, so Python will keep looking for (and writing to) files in this directory for the remainder of this session. This requires that you import the module `os` (short for "operating system"):

```
>>> import os
>>> os.getcwd()
'C:\Python27'
>>> os.chdir(r'C:\Users\stk12004\Dropbox\Classes\UConn\3798\Materials')
>>> myfile = open('moby.txt', 'r')
>>>
```

2. OR you can refer to the file with its full path. In this case, the current working directory remains as before, so you'll have to give full paths throughout the session:

```
>>> myfile = open(r'C:\Users\stk12004\Dropbox\Classes\UConn\3798\Materials\moby.txt', 'r')
>>>
```

[↑ top of this page](#)

How do I scroll through the command history in IDLE?

One convenient feature of the IDLE environment is that it lets you scroll through your command history. By default, you do that with `Alt-p` and `Alt-n` for the next and previous command, respectively. Some people prefer to use the Up- and Down- arrows instead. To set this up, do the following in the IDLE window:

```
Options -> Configure IDLE -> Keys
select "history-next" in the "Custom Key Bindings" menu
click "Get New Keys for Selection"
select "Down Arrow"
click "OK"
```

You will be prompted (just the first time to give a name to the key setting. Give it one. Then do the analogous thing for "history-previous" and the up-arrow. Adjust any other settings to your liking.

[↑ top of this page](#)