
Course Plan: Computational Tools for Psycholinguistics

Total number of lectures: 14 (1st Nov 2011 - 11 Feb 2012)

Total labs/lectures: 14

Assignments: 7

Mini project: 1

Course Description:

The course has the following goals:

- It will introduce basic shell/Linux commands
- Give an introduction to the basic syntax of Python
- Use Natural language toolkit (NLTK) to process text.

All this will be looked at from a perspective of handling natural language text.

Prerequisites:

Some basic knowledge of Linguistics will be assumed during the lectures. Knowledge of some prior scripting language will be very handy (but is not compulsory).

Evaluation:

Assignments: 30%

Mini-project: 30%

Final exam: 40%

Lecture no.	Topic
1	<i>Linux/Shell: Basics, some handy commands</i> <ul style="list-style-type: none">● Bash profile, Environment variables● File organization: cp, mv, rm, mkdir, cd, rmdir, pwd.● Listing files and directories: ls, less, more, cat, head, tail, vim editor● File permission● man, info● Command line options, pipes, wild cards● Advanced: cut, paste, diff, tr, uniq, sort, grep, wc● tar, gzip
2	<i>Linux/Shell (contd)</i> [Assignment 1]
3	<i>Python I: Basics</i> <ul style="list-style-type: none">● Numbers, strings, Unicode strings, lists● Basic coding

4	<p><i>Python II: Control flow</i></p> <ul style="list-style-type: none"> ● if, for, range ● Functions <p>[Assignment 2]</p>
5	<p><i>Python III : Data structure</i></p> <ul style="list-style-type: none"> ● List ● Set ● Dictionary
6	<p><i>Python IV: Modules, I/O</i></p>
7	<p>[Test]</p>
8	<p><i>NLTK I : Language Processing and Python</i></p> <ul style="list-style-type: none"> ● How to install NLTK? ● Loading text, Concordance, Dispersion plot ● Vocabulary: token, word type, Average count, lexical diversity <p><i>Python revision</i></p> <ul style="list-style-type: none"> ● Lists: concatenation, append, index, slicing ● Variable: assignment, reserved words, naming rules ● Strings
9	<p><i>NLTK I contd</i></p> <ul style="list-style-type: none"> ● Frequency distribution ● Fine-grained selection of words ● Collocations, Bigrams <p><i>Python revision</i></p> <ul style="list-style-type: none"> ● Conditionals: relational operators, comparison operators ● Iterations on every element ● Nesting blocks ● Looping with conditions <p>[Assignment 3]</p>

NLTK I contd

Automatic Natural language understanding

- Word sense disambiguation
- Pronoun resolution
- Generating language output
- Machine Translation
- Spoken language systems
- Textual entailment

NLTK II : Accessing text corpora and lexical resources

Gutenberg Corpus

- Average word length, avg. sentence length, lexical diversity score

Web and Chat text

Brown corpus

- Stylistics

Reuters corpus

Inaugural address corpus

Basic NLTK corpus functionality

Loading your own corpus

Conditional frequency distribution

- Plotting and Tabulating Distributions
- Generating random text with bigrams

11	<p><i>NLTK II contd</i></p> <p><i>Python revision</i></p> <ul style="list-style-type: none">● Functions: Local variables● Methods <p><i>Lexical resources</i></p> <ul style="list-style-type: none">● lexical entry/lemma, homonyms● Wordlist corpora: stopwords● Pronouncing dictionary● Comparative wordlist● WordNet: Sense, Synonyms, Hierarchy, Hyponyms, etc.● Semantics similarity <p>[Assignment 4]</p>
12	<p><i>NLTK III : Processing raw text</i></p> <ul style="list-style-type: none">● Tokenization● Processing Feeds● Reading local files● Reading other formats and user input● Text processing with Unicode <p><i>Python revision</i></p> <ul style="list-style-type: none">● Strings: Basics, concatenation, Printing● Accessing characters and substrings

13	<p><i>NLTK III contd.</i></p> <ul style="list-style-type: none">• Regular expressions: Metacharacters, ranges and closures• Useful applications using Regular expressions• Normalizing text: Stemmer, Lemmatization• Segmentation <p>[Assignment 5]</p>
14	Final Exam

Primary references:

<http://docs.python.org/tutorial/index.html>
[Natural language processing with Python](#)
<http://www.nltk.org/book>

Additional references:

[Think Python](#)
[Dive into Python](#)
[A byte of python](#)