Computational Methods in Linguistics

Bender and Wassink 2012
University of Washington
Week 3: 4/9-4/11

Overview

- Version control: svn tutorial
- Python basics
- Python example: removing mark up
- Debugging basics
- Reading questions
- Discussion of research questions

Version control

- Maintain snapshots of work in progress
- Collaborate with others on the same documents
- Maintain log of what changed and why between snapshots
- With plain text files: smart merge
- How do you do these things now?

Version Control with svn on lemur

- "Free" back up on a separate machine
- Access your files from anywhere
- Never lose track of what's the most current version
- Work fearlessly: easy to back out errors
- Clean record keeping (who changed what and why, via log messages)

Work flow with version control

- (One time only): Import directory into svn repository
- (One time only): Check out svn repository
- svn update
- do some work
- svn update
  - resolve any conflicts
- svn commit

svn conceptual model

- Repository (stored on lemur) stores current version plus revision history (plus log files)
- Working copy (local machine) reflects repository, stores diffs (local changes)

Typical commands

- svn import dir svn://lemur.ling.washington.edu/shared/foo
  - Create repository out of directory “dir”
- svn checkout svn://lemur.ling.washington.edu/shared/foo
  - Make local copy of repository “foo”
- svn update #invoked in foo
  - Merge any changes in repository to local copy
- svn commit -m “log message” #invoked in foo
  - Commit local changes to repository

There's more (but you don't need to worry about it)

- Branches
- Merge
- Rolling back to older versions
  - ... cross these bridges when the need arises

Kinds of software

- Quick throw-away script to test an idea
- Something that you'll use as part of a research project (cf quality control)
- Something that you'd share with other people who need to do the same thing
- Something you'd provide as an off-the-shelf solution to other people

Python basics

- An *interpreted* language
  - Not quite as fast as e.g., C++
  - Can play with handy interpreter prompt
  - The language defines a syntax and semantics and provides a set of built-in functions
  - Packages (including NLTK) define many many other useful functions
  - NLTK (Bird et al) is a set of python implementations of lots of useful NLP stuff

How do I get it?

- On patas, at the prompt type "python" or "python2.6" (python2.6 is required for NLTK)
- To install on your own machine: http://python.org
- To install NLTK locally, follow directions in NLTK book exerpt

Prompt v. scripts

- If you don't give python an argument, it'll open up the python interpreter in interactive mode (demo)
- You can also give an argument which is a file containing python commands
    - (Any further arguments would then be accessible to that python script.)
- Demo python prompt

Some fundamentals

- Variables: store values, allow for abstraction
  - Variable names can be any combination of numbers, letters, and _
- Functions: Take zero or more arguments, do some computation, and return a value and/or have other side-effects
  - Examples: print(), len(), int(), raw_input()

Data types

- integers: 7, 2, -5
- strings: “hello”, “bom dia”, “7”
- lists: [-7, 2, 5], [“hello”, “aloha”]

Control structures

- if [test] else
- while [test]
- for [variable in range]

Example

- Removing mark up
- Input: shortscrappletranscript.txt, with mark up
- Output: Just the text, ma'am
- Look at input: What are we trying to get rid of?

Debugging basics

- Test frequently --- each change, if possible
- Keep track of where you've been (don't go around in circles)
- Maintain snapshots of working versions (through svn) and compare (svn diff) to see where the changes are

Debugging basics

- Make use of error messages, including pointers to line numbers
- ... but keep in mind the actual problem might be elsewhere (i.e., earlier)
- Demo debugging by breaking remove-markup.py

Reading Questions

- What is OOP?
- Why do count() and concordance() return different numbers?
- What else besides Python is used in compling?
- How important are those version issues?

Reading questions

- How can we find different uses/senses of a word?
- How does the text's concept of lexical diversity relate to other uses of that term?

Overview

- Version control: svn tutorial
- Python basics
- Python example: removing mark up
- Debugging basics
- Reading questions
- Discussion of research questions