NLTK:
Introduction to Natural Language Processing

Steven Bird   Edward Loper   Ewan Klein

University of Melbourne, AUSTRALIA

University of Pennsylvania, USA

University of Edinburgh, UK

Python: Key Features

- simple yet powerful, shallow learning curve
- object-oriented: encapsulation, re-use
- scripting language, facilitates interactive exploration
- excellent functionality for processing linguistic data
- extensive standard library, incl graphics, web, numerical processing
- downloaded for free from `http://www.python.org/`

Python Example

```
import sys
for line in sys.stdin.readlines():
    for word in line.split():
        if word.endswith(’ing’):
            print word
```

1. whitespace: nesting lines of code; scope
2. object-oriented: attributes, methods (e.g. `line`)
3. readable

Comparison with Perl

```
while (<>) {
    foreach my $word (split) {
        if ($word =~ /ing$/) {
            print "$word\n";
        }
    }
}
```

1. syntax is obscure: *what are:* `<>` `$` `my` `split` ?
2. “it is quite easy in Perl to write programs that simply look like raving gibberish, even to experienced Perl programmers” (Hammond *Perl Programming for Linguists* 2003:47)
3. large programs difficult to maintain, reuse

What NLTK adds to Python

NLTK defines a basic infrastructure that can be used to build NLP programs in Python. It provides:

- Basic classes for representing data relevant to natural language processing
- Standard interfaces for performing tasks, such as tokenization, tagging, and parsing
- Standard implementations for each task, which can be combined to solve complex problems
- Extensive documentation, including tutorials and reference documentation

Installing Python and NLTK

1. Install Python, Numeric
2. Install NLTK-Lite, NLTK-Lite-Corpora
3. Set environment variable `NLTK_LITE_CORPORA`

For detailed instructions, see:

- http://nltk.sourceforge.net/install.html
- CDROM: /webpage/install.html