Ling 555 — Programming for Linguists

Python — Linguistic Examples and Functions (part I)

Robert Albert Felty

Speech Research Laboratory
Indiana University

Oct. 15, 2008

Prereqs

While we are waiting

Please download the courseBackground.txt file from:
http://robfelty.com/teaching/L555Fall2008/
resources/sapir.txt
`curl -o courseBackground.txt`
`http://robfelty.com/teaching/L555Fall2008/`
`resources/sapir.txt`

For Wednesday:

Read Chapter 6 on Abstraction (Functions)

Outline

1. Homework questions

2. Linguistic Examples
   - frequency
   - co-occurrence
   - Types and tokens

3. Resources

4. Functions
   - parameters
   - Return values

Homework questions?

frequency

What to do with a dict?

You could construct a dictionary of a corpus's word frequencies. Each word would be a key, and the corresponding value would be its frequency.

co-occurrence

Definition

One way to measure semantic relatedness is by co-occurrence frequency. If two words frequently occur close to each other in text, they probably share a semantic relationship.

Different kinds of frequency

- Some words might be very frequent, and relatively context independent. That is, they occur with many other words
- Other words might be frequent, but only occur with a few other words
- These types of words are probably stored and accessed differently in the brain

Type Token Ratio

Definition

The type-token ratio is defined as the ratio between the number of types (unique words) and the number of tokens (total number of words) in a text.

- Type-token ratio is one way to measure linguistic density (sometimes referred to as diversity or complexity).
- While English is sometimes to reported to have as many as 500,000 words, most people only know somewhere in the 10,000-50,000 range.
- One can expect that people who know lots of words will end up repeating words less. This can be measured by the type-token ratio.

Resources for you:

Some of the examples we have covered in class today can be found on the website at:
http://robfelty.com/teaching/L555Fall2008/resources/py
The files from today are:

1. co-occurrence.py

What is a function?

Definition

A function is something you can call (possibly with some parameters, the things you put in the parentheses), which performs an action and returns a value.

Example

```python
def hello(name, greeting):
    return greeting + ", " + name"
print hello('rob', 'hallo')
```

Define first, then call!

In python, a function must be defined before you can call it. That is, define it on line 10, call it on line 15.

Parameters (Arguments)

Definition
Parameters (also known as arguments) are inputs to functions.

Example
When you use the `min()` function, you pass the function a list as a parameter

3 types of Parameters

positional Positional parameters must be entered in the correct order
```
hello(name, greeting)
```

keyword Keyword parameters can be entered in any order
```
hello(greeting='Servus',
name='rob')
```

collected Parameters can also be collected by a function, allowing the user to input any number of parameters to the function
```
def hello3(*collectedParams):
    return collectedParams
print hello3('foo', 'bar', 0)
```

Parameter types

Definition

Any kind of variable can be passed to a function (string, integer, float, list, dict, tuple, object). Your function must use these as the right type though.

Example

```python
def sortStudents(students):
    return sorted(students)

theStudents='John and Mary'
print sortStudents(theStudents)
theStudents=['John', 'Mary']
print sortStudents(theStudents)
```

Return values

Definition

Parameters are inputs to functions. Return values are outputs.

Multiple return values

To return more than one value, put them in a tuple

```
def hello():
    x=1
    y=2
    return (x,y)
foo=hello()
one,two=hello()
```

Function No-Nos

homework

Linguistic Examples

Resources

Functions

parameters

Return values

No-no

Don't print out stuff in functions (unless debugging)

```python
def hello():
    print "hello, world"
```

Yes-Yes

Do return stuff in functions and print later

```python
def hello():
    return "hello, world"
print hello()
# OR
sys.stdout.write(hello() + "\n")
```