

NLP Lab Session Week 1  
January 21, 2010

## Starting an NLTK Session

This session assumes that you have followed the directions for downloading NLTK, starting the Python IDLE window and tested the download.

You will have the IDLE window open for testing NLTK and a browser window open with these instructions. You may also want to have a separate tab or window open to the NLTK book: <http://www.nltk.org/book/>, where these examples are taken from Chapter 1.

In the following, examples for you to try are given following the Python Idle prompt of `>>>`. You can copy and paste the Python example into the Idle window, or you can type the example in.

Start by typing a couple of examples of arithmetic into the Python interpreter. For example:  
`>>> 1 + 2`

Note that if you want to type in a string of text, you surround the string with quotes.  
`>>> "hello"`

Next, you use the Python “import” statement to load the data used in the book examples into the Python environment:

```
>>> from nltk.book import *
```

This command loaded 9 of the text examples available from the corpora package (only a small number of them!). It has used the names `text1` through `text9` for these examples. These names are “variables” in Python. If you type the variable name, you get a description of the text:

```
>>> text1
```

## Searching Text

The text data structure has a number of functions to operate on text. One is called “concordance”, and it will search for any word that you give to the function and show you the occurrences and some surrounding context.

```
>>> text1.concordance("monstrous")
```

Observe the use of the arrow keys with the enter key to select and modify previous lines in Python, and try a similar example.

```
>>> text2.concordance("affection")
```

Another function is “similar” which finds all the words that are used in the same context as the one given, where the context is the word before and the word after.

```
>>> text1.similar("monstrous")
```

We can use this to compare how the same word is used differently in other texts.

```
>>> text2.similar("monstrous")
```

## Counting Vocabulary

Each text can be separated into a list of tokens. These consist of words and all the punctuation and other symbols occurring in the text. To further investigate text, we can count the occurrences of words.

We start by using the Python length function, “len” to tell us how many things are in a list. (Strictly speaking, each text variable is an object of type `nltk.text.Text`, which contains the text string and some other functions, but we’re trying not to explain much programming here.)

```
>>> len(text3)
```

```
>>> len(text4)
```

Now this is the total number of tokens, and we might also want to find out how many unique words there are, not counting repetitions. The Python “set” function removes the repetitions, and we can apply the “sorted” function to that, returning the resulted sorted list of tokens. When we type this, lots of words will flash by on the screen.

```
>>> sorted(set(text3))
```

Or we can just find the length of that list.

```
>>> len(sorted(set(text3)))
```

Now let’s compute the ratio of the total number of tokens to the number of unique tokens and we’ll get an average of how many repetitions there are for each word. First we get a division operator that uses real arithmetic (aka floating point) instead of integer and then we divide to get the ratio.

```
>>> from __future__ import division
```

```
>>> len(text3) / len(set(text3))
```

Now let’s search for and count occurrences of particular words and compare that to the total number of words.

```
>>> text3.count("smote")
```

Compute the fraction of the number of occurrences of the word compared with the total number of words and then multiply by 100 to get a percentage.

```
>>> 100 * text3.count('smote') / len(text3)
```

How does this compare with a more common word, such as the word “a”?

```
>>> 100 * text3.count('a') / len(text3)
```

## Independent Investigations

For this first wee, just try the examples individually and see what happens. Compare notes with the person next to you, and we will discuss at the end what you found.

1. How many times does the word “lol” occur in text5? What is the percentage of its occurrences in the text? [Warning: text5 is uncensored chat]

Think of another word to find occurrences and get the number of occurrences and its percentage in the text.

2. Go to the book chapter 1 and under the section Searching Text, try the dispersion plot example on the presidential Inaugural Addresses. Since these addresses occur in the text in order from 1789 to the present, the horizontal axis represents time, and you can see the difference in word usage across time.

Try the dispersion plot with text2, Sense and Sensibility, with the four main characters, Elinor, Marianne, Edward, and Willoughby. What can you observe about the different roles played by the males and females in this novel? Can you identify the couples?

3. (Optional, for those interested in learning more about Python and programming NLTK) Go to the NLTK API page and look for the class for `nlk.text.Text` in the list in the lower left pane. Look at the available functions. What other functions look interesting? Note that the constructor wants a list of string tokens, and we’ll talk more about that next time.