

# 684.02 Data Intensive Computational Linguistics

## Assignment 2

Due 5:00pm 24 Jan 2002

Please review the first NLTK tutorial.

**Accessing NLTK.** You should be able to access Python2.2 using Emacs on the Linguistics Unix system or direct from the command line. If you choose to do the work on your own machine (Windows or Linux) it is your responsibility to ensure that your code also works identically on the departmental system.

**Grading.** Points will be awarded for well-structured and well-documented code. Points will be deducted for late submissions. Assignments must be strictly original work.

## 1 Language Identification

This is a practical exercise in language identification.

Copy into your own filesystem the code in:

`/public_html/2002/winter/684.02/code/langid/language_id.py` This defines a set of functions that implement the language identification technology that we talked about in the class. You'll also need the data in the files `train.e`, `train.f`, `test.e`, `test.f` in the same directory. Copy all these files into a directory in your own space, along with the Python code. We'll call the directory `/home/cbrew/snlp2` but you should use another name.

## 2 What you should do

- Bring up a terminal.
- Within that terminal, type `cd /home/cbrew/snlp2`.
- Run emacs by typing `emacs language_id.py`.
- Start the Python interpreter (using C-c !)
- Load the code from the provided files (using C-c C-c)
- Switch to the Python window and try running the command

```
tryit(8)
```

This trains and tests language identification models for French and English, using a test suite consisting of strings of length 8. You should see roughly the following. (Timings may vary)

```
>>> tryit(8)
684.02 Language identifier
Training took 12 seconds
Building test suites. Chunks of size 8
0.914118 success rate for French test corpus
```

```
0.882583 success rate for English test corpus
It all took 23 seconds
>>>
```

Now try the same with a larger string length.

```
>>> tryit(12)
684.02 Language identifier
Training took 11 seconds
Building test suites. Chunks of size 12
0.956777 success rate for French test corpus
0.938533 success rate for English test corpus
It all took 31 seconds
```

Note how allowing a larger test string improves the performance.

Next, examine the code, and work out what to change in order to work with different texts. Acquire, some different texts, and test the algorithm. You should test the effects of training corpus size (use a substring of the corpus), size of test string (we saw that above), order of N-grams used. Keep notes of what works and what doesn't. A good report satisfies the assignment.

For extra personal satisfaction, improve the code to work better. (for example, if working with the Hansard data I would be tempted to try something about proper names, which are noise for purposes of language identification) If you do this you should really have a training corpus, a development corpus that you use a a test corpus while you are trying to improve the code, then a final test corpus that you use to demonstrate that your alleged improvements were general enough to also improve things when applied to data that you didn't consider until the improvements were made. Peeking at the test corpus before you are done is bad practice, and gives you a false impression of how well your ideas are working.

**Submission.** I need, by the due date, a report on the performance of the language identification technology on a pair of languages, neither of which should be either English or French. In order to do this you will have to adapt the code which I provided.

Please email your completed assignment as a *single text file* to me [cbrew@ling.ohio-state.edu](mailto:cbrew@ling.ohio-state.edu), in the body of the message. Please use plain ASCII (no zip files or attachments) and ensure that no lines are longer than 80 characters. Use `language_id.py` as a basis, but include any changes to the code that you need in order to work with your new languages, and provide a description of what you tried, what worked well and what you learnt by replacing the

```
"""
=====
PUT DESCRIPTIVE COMMENT HERE.
=====
"""
```

with your own comments (thus)

```
"""
Name: Chris Brew
Languages used: Klingon and Vulcan
```

NB. This report is just a silly example. All false

I found it really hard to get reliable native speaker intuitions about these languages, but I found a corpus which is available in my home directory under `/home/cbrew/data/klingon` and `/home/cbrew/data/vulcan`.

The technology worked fine. order-4 ngrams were better than order-3 or order-5. Test strings had to be at least 8 characters long. If the training corpus is less than 500 characters performance degrades considerably.

And so on.

""

Don't write more than you need, but do make this a proper scientific report on what you did. Choose interesting languages if possible.

Have fun.