

## Exercise: Web as Corpus – The BootCat Tool

### 1 Task

Create a Corpus from the Web, based on a list of keywords ('seeds'). As additional tasks apply some basic lexical statistics (see exercise on basic Unix tools, 31.7.2006) and create a part-of-speech tagged corpus.

### 2 Getting Started

- Move in the BOOTCAT directory: `cd /home/alumno/ESSLLI/BOOTCAT`

The directory contains a set of perl programs which make up the BootCat Tool. We won't use them but work with an online version of BootCat instead.

- **Seeds**

Before you start the browser, look at the 'seeds' file. It contains five keywords which are the starting point for your collection of web pages: `less seeds`

The seed words are expected to be representative for the domain under investigation, e.g. the domain 'Spanish wine and sherry'.

- **N-tuple construction**

The tool will create a list of n-tuples (n being a predefined number, e.g. 3-tuples) out of the seed words by randomly combining them. These n-tuples are used as keywords in the google search.

To give you an idea how this looks like create a 3-tuples list yourself: `build_random_tuples.pl -n3 -l10 seeds`

'build\_random\_tuples.pl' is one of the BootCat perl programs.

'-n3' defines the tuple size (here '3')

'-l10' defines the number of tuples created (here '10')

### 3 Bootcat WWW version

1. Start mozilla: `mozilla &`
2. Go to `http://corpora.fi.muni.cz/bootcat/`
3. Scroll down the page and click on '**Advanced Search**'
4. Update seed words in a plain text file: 'Examinar'  
click on (ESSLLI > BOOTCAT >) 'seeds'
5. Type in (better: copy and paste) your google api key. (If you don't have your own key ask the instructors for assistance).
6. Keep language: English
7. Do **not** tag the corpus (this you can do later, see section 4).
8. Choose a name for your corpus, e.g. 'test' or 'SpanishWineSherry'.
9. You don't need to give your email address. We will build a small toy corpus only.
10. Do **not** click on 'build a corpus' yet.
11. Scroll down to 'Advanced Options'.
12. Change the default setting of **Max. URLs per query to 5.**

13. Change the default setting of **Max. page size (in kB) to 1000**.
14. Keep the other settings.
15. Now click on **build corpus!**

While the system creates your corpus take a break or click on ‘publications’ at the bottom menu and have a look at Baroni et al. (2006).

When your corpus is created.

- Look at it with the online concordancer: the **Sketch Engine Concordancer** is based on cqp and offers a comfortable keyword in context search:
  1. Click on the URL of your corpus
  2. Concordance: type in some keyword then click on ‘new search’, scroll through the result.
  3. Corpus: ‘word list’, set number of items to 1000. It shows you a frequency list of the corpus (cf. Monday’s exercise).
  
- **Download** your corpus.
  1. Choose ‘text download raw’
  2. You can edit the suggested name if you want, e.g. call it ‘test.raw.txt’ or ‘SpanishWineSherry.raw.txt’
  3. Save your corpus in the BOOTCAT directory.
  
- **Inspect** your corpus
  1. Go back to the terminal window. Less <your corpus>, e.g. `less test.raw.txt`
  2. Look through the corpus. Each saved webpage starts with “CURRENT URL: http://...” . Are the texts about the expected topics? Is there extra-textual material like html code?

## 4 Additional Tasks

### 1. Lexical statistics

Create a frequency list, a rank frequency profile and a frequency spectrum of your corpus. Does it show the same ‘Zipfian distribution’ as the corpus on Monday?

### 2. Create a **tagged corpus**

Repeat the Bootcat procedure but choose ‘tagged corpus’ in step 3.7 and save the vertical format in step ‘Download-1’. Look at the results. How does the tagger deal with extra-textual parts?

## Online resources

BootCat tools: <http://sslmit.unibo.it/~baroni/bootcat.html>

BootCat www version: <http://www.corpora.fi.muni.cz/bootcat/>

WaCky project: <http://wacky.sslmit.unibo.it/>

## References

M. Baroni, A. Kilgarriff, J. Pomikalek and P. Rychly. 2006. [WebBootCaT: Instant domain-specific corpora to support human translators](#). Proceedings of EAMT-2006 Poster Session. 247-252.