UW Sociolinguistics Lab Corpus Access: Command Line Interface

Last updated March 2019

Based on a guide produced by John McCranie (CLI developer)

This guide will assist you in using the command line interface for accessing the Socio Lab's corpus holdings. In order to access the corpora (COCA, COHA, CMSW, and GLOWBE), you will need the ability to connect to the lab's server remotely from your own computer. The process for doing this will be different depending on the operating system you use. In the steps below, commands intended to be typed at the terminal shell start with a dollar sign prompt:
($ cd ~, $ ls)
You do not need to type the dollar sign; instead, type (or copy & paste) all of the information that follows the $.

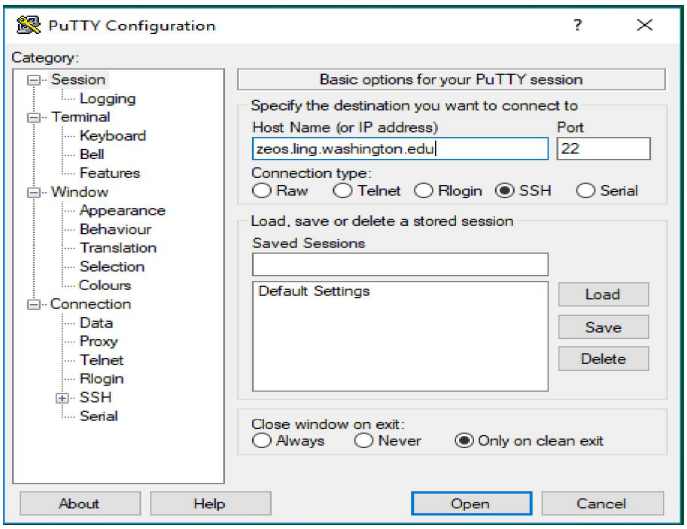
Mac:
You computer should already have an application installed called 'terminal.' Launch terminal and type in the following command:
$ ssh corpsearch@zeos.ling.washington.edu

You will then be prompted for the password:
$ corpsearch@zeos.ling.washington.edu's password:

Enter the password that was given to you by the lab staff.

Windows:
For Windows, you will need to download a separate program that allows SSH access to a server. Some options include PuTTy (www.putty.org) or MobaXterm (https://mobaxterm.mobatek.net/). If using PuTTy, we would recommend also downloading a graphical SFTP client such as WinSCP (https://winscp.net/eng/index.php), which will make downloading your search results easier (this interface is included in MobaXterm).

In your SSH emulator, you will need to access the address of zeos.ling.washington.edu at port 22 (screen shots are from a PuTTy interface):



Once you open the connection, you will be prompted with the following:
login as:

Here, enter corpsearch which will lead to the following:
corpsearch@zeos.ling.washington.edu's password:

Enter the password that was given to you by the lab staff.

From this point forward, the directions are the same for both Mac and Windows operating systems.

1. Starting SearchMatrix, the Corpus Command Line Interface (CLI) Program:

You should now have something similar to the following prompt in your terminal:
`[corpsearch@zeos ~] $`

To launch the corpus search tool, type searchmatrix in at the prompt and specify a file for saving your searches by adding --session-file myFile.txt (where you replace myFile with a name that helps you find, identify, and retrieve your search output). Hit enter:
`[corpsearch@zeos ~] $ searchmatrix --session-file myFile.txt`

Please note: if you do not specify an output file using the --session-file command you still be able to perform searches but you will not have a method of saving searches unless you copy and paste them from the terminal (which we would recommend against).

This will launch the tool, and you should see the following in your terminal:

[corpsearch@zeos ~]$ searchmatrix --session-file myFile.txt
writing session output to: myFile.txt

corpus server command line client, version 0.2

sm>

The sm> is the command line prompt for the SearchMatrix tool, much like the $ from the previous section. You do not want to enter or copy & paste the sm> from the commands that follow; instead, you will want to enter everything that follows the sm> prompt.

Before conducting any searches, we want to make sure that the tool is working correctly. To do this, we will check the server connection by typing in “server” at the sm> prompt. After typing “server” and hitting enter, you should see the following output:

sm> server
    name: /home/corpsearch/project/CURRENT/trunk/src/bin/snowmutt_server.py
    version: 0.2

If you do not see this output, or you are forced out of SearchMatrix (the prompt in the terminal changes from sm> back to $), contact lab staff at linglab@uw.edu or lxlabsa@uw.edu for further assistance.

2. SearchMatrix Basic Commands

The client is just a simple command loop: it reads a command, executes it, prints the output, and goes on to the next prompt. You can use the up and down arrow keys to scroll through the history of commands used in the current session (if you log out and log back in, the commands will no longer be available to you).

To view the commands that may be entered to run searchmatrix, type "help" at the sm> prompt. Add the command name after "help" to view some information about that command.

You should be able to copy and paste the commands below, and get the same output.

```
sm> help


Documented commands (type help <topic>):
========================================
EOF      corpora  exit  pass    search  set    show
commands  corpus   help  results  server  shell


Examples:

sm> help exit

        quit the program

sm> help corpus

        corpus <clabel>

        print the details about the given corpus
```

3. Corpora Available in SearchMatrix

The SearchMatrix CLI currently allows access to four corpora, broadly outlined below:

1-Corpus of Contemporary American English

corpus label: coca

fully available searches: lemmas (2M), lexicon (5.4M), pos (8k), sources (240k)

partially available: kwic, 460M rows subset (out of 620M rows, is in the process of being loaded)

2-Corpus of Historical American English

corpus label: coha

available searches: lemmas (600k), lexicon (3.9M), pos (8k), sources (115k), kwic (470M)

3-Corpus of Global Web-based English

corpus label: glowbe

fully available searches: lemmas (725k), lexicon (13.6M), pos (10k), sources (170k)

partially available: kwic, 1M rows subset (out of 2G, rest will be loaded after COCA done)

4-Corpus of Modern Scottish Writing

corpus label: cmsw

available searches:
search label: kwic (5M)

If you want to see the corpora available in SearchMatrix, or want to see the labels associated with them in order to perform searches, type in "corpora" at the sm> prompt:
sm> corpora

you should see the output:

label: coca
name: Corpus of Contemporary English
description: the full COCA data set

label: coha
name: Corpus of Historical English

description: the full COHA data set

label: cmsw
name: Corpus of Modern Scottish Writing
description: test run of CMSW

label: glowbe
name: GLOWBE
description: the full GLOWBE data set

The corpus labels we will use for future requests (searches) are: coca, coha, glowbe, cmsw

4. Details and Searches available for each corpora

To get detailed information about each corpus available in SearchMatrix, type in “corpus <corpusLabel>” at the prompt:

sm> corpus cmsw

corpus

label: cmsw
name: Corpus of Modern Scottish Writing
description: test run of CMSW
search

label: kwic
name: Keyword in Contex Search
description: search the words in context
fields: source_id, year, genre, word, word_prev01, word_follow01, word_prev02, word_follow02, word_prev03, word_follow03, word_prev04, word_follow04, word_prev05, word_follow05, word_prev06, word_follow06, word_prev07, word_follow07, word_prev08, word_follow08

This tells us that the cmsw corpus has 1 search available, named "kwic", and that will show us the fields named source_id, year, genre, and so on.

To see the details of coha:
sm> corpus coha

corpus

label: coha
name: Corpus of Historical English
description: the full COHA data set

search

label: kwic
name: Keyword in Contex Search
description: search the words in context
fields: source_id, genre, year, word, lemma, pos, word_follow01, word_prev01, lemma_prev01, lemma_follow01, pos_prev01, pos_follow01, word_follow02, word_prev02, lemma_prev02, lemma_follow02, pos_prev02, pos_follow02, word_follow03, word_prev03, lemma_prev03, lemma_follow03, pos_prev03, pos_follow03, word_follow04, word_prev04, lemma_prev04, lemma_follow04, pos_prev04, pos_follow04, word_follow05, word_prev05, lemma_prev05, lemma_follow05, pos_prev05, pos_follow05

search

label: lemmas
name: Lemma Search
description: search the lemmas
fields: id, name

search

label: lexicon
name: Lexicon Search
description: search the lexicon
fields: id, word, lemma, pos

search

label: pos
name: Part of Speech Search
description: search the part of speech list
fields: id, pos

search

label: sources
name: Sources Search
description: search the source set
fields: id, num_word, genre, year, title, author, publisher

This tells us that there are 5 searches available for coha. The keyword in context search, named "kwic", is similar to the one available for cmsw. in addition, there is a search named "lemmas", which has fields id, name. There is a search named "lexicon", which has fields id, word, lemma, pos. There is a search named "pos", that has fields id and pos. Finally there is a search named "sources", that has fields id, num_word, genre, year, title, author, publisher.

To see the details about coha and glowbe:

sm> corpus coca

sm> corpus glowbe

This will provide the list of available searches for them. All have the same names and fields as the ones available for coha.

5. Performing searches in SearchMatrix

There is a basic structure to the searches performed in SearchMatrix, brackets [ ] indicate required elements that vary by search/corpus; curly brackets {} indicate optional elements:

sm> search [corpus] [label*] {field**} {restrictor} {searchTerm}

*Anything from the search label portions of the corpus in question; see the previous section for the available labels in each of the corpora.
**Equals the contents of the search fields portions of the corpus search label in question.

The restrictor serves to constrain the search that you are performing. Any field that is a string of characters, or text field, can be tested for these five conditions:

1. equality, =, restrictor is "eq",
2. inequality, or =/=, restrictor is "neq",
3. equality ignoring case, restrictor is "eqic",
4. database like operator restrictor is "like",
5. regular expression pattern match, restrictor is "match".

Note that restrictors 4 and 5 are a bit more involved (and powerful) search options. There is detailed information on these two options later in the guide (Section 8).

The searchTerm is the lexical item which you are interested in.

For example, if we were interested in a kwic (keyword in context) search of 'dog' in the Corpus of Modern Scottish Writing, we could enter the following search:

sm> search cmsw kwic word eq dog

The above code is telling SearchMatrix to perform a search in the cmsw corpus based on the kwic. What is being searched for is a word, and the word that we want are all instances of dog (lower case).

More information about the search labels can be found in the following section.

More information about the restrictors can be found in the Appendix. Please note that the current release of the CLI only allows for one restrictor per search.

Returning to performing searches, we can do a search for “anymore” as a lemma in the Corpus of

Historical English using the following search:

```
sm> search coha lexicon lemma eq anymore
        status: success
```

Note that once a search is complete, it will return a status of "success." Some searches may take longer; currently, the CLI does not offer a "processing" visualization; you will know that the search is complete when you get the status: success return and again have access to the sm> prompt.

You will notice that, while the search has been completed, you are not seeing any results. In order to see the results of the previous search, you will use the "results" command. The "results" command will only print the results of the most recent search; there is currently not a method for printing the results of earlier searches. When you invoke the "results" command, you have the following options:

```
sm> results 100      ***
sm> results all
sm> results file
```

***You can use any number here; SearchMatrix will then print to the screen the number of rows retrieved for your search up to that number (less if there were fewer rows).

Please note that "results ##" and "results all" only print your results to the screen. If you want to save your results, you will need to use "results file", which will write your output to the --session-file that you specified when you launched SearchMatrix.
An example (**and important note on using "results file"**):

Let's say that we are interested in all contexts in which we find the word "mouse" in the Corpus of Historical English. We could use the following command in SearchMatrix:

```
sm> search coha kwic word eq mouse
        status: success
```

After performing the search, we decide that we want to verify that it captured what we were interested in so we will look at the first 10 lines that it returned and get the following result:

```
sm> results 10
8694 fic 1810 nor the stirring of a mouse .So chilly -- Mar nor the stir of a mouse .so chilly -- mar cc at jj io at1 nn1 yrr jj z npm1_vv0
8694 fic 1810 snivelling marquises , with a mouse &#x27;s soul , a lady snivel marquise , with a mouse &#x27;s soul , a lady jj@_nn1@ nn2 y iw at1 nn1 ge nn1 y at1 nn1
8637 fic 1811 there as still as a mouse ; &quot; For Mahomet curse there as still as a mouse ;&quot; for mahomet curse rl rg rr csa at1 nn1 y&quot; if np1_nn1 vv0
7433 fic 1813 In the jaws of a mouse , and the sly little in the jaw of a mouse ,and the sly little ii at nn2 io at1 nn1 ycc at jj
```

jj
8564 fic 1814 in that manner -- a mouse could not have pass 'd in
that manner -- a mouse could not have pass have ii dd1 nn1 z at1 nn1
vm xx vhi nn1 vhn
8613 fic 1815 cravat Peeping out like a mouse from acheese With shoes cravat
peep out like a mouse from acheese with shoe nn1 vvg rp ii at1 nn1 ii
at1 nn1 iw nn2
8717 fic 1815 themselves , nor ketch a mouse :I wonder he did
themselves , nor ketch a mouse : i wonder he do ppx2 y cc vv0
at1 nn1 y ppis1 vv0 pphs1 vdd
8861 fic 1818 about . Just like a mouse in awire trap , about . just
like a mouse in awire trap , rp@ y rr ii_vv0@ at1 nn1 ii at1 nn1
nn1_vv0 y
8706 fic 1819 be grown Just fit to mouse , and gnaw a bone be
grow just fit to mouse ,and gnaw a bone vbi vvn rr vv0_jj ii nn1 ycc
vv0 at1 nn1
8988 fic 1819 you . Not e'en a mouse could stir ' twixt one
you . not e'en amouse could stir ' twixt one ppy y xx jj_nn1
at1 nn1 vm vvi ge nn1 mc1

Because it is printing to the screen, this is not easily human-readable, but it looks like we are getting instances of the word mouse in context. Now we want to write all of the results to our output file (perhaps to work with it in R or some other program for processing texts). We would assume that we could just do the following:

```
sm> results file
rows written to file: 0
```

If we do this nothing will be written out to the output file. This is somewhat counter-intuitive from a user perspective, but you will need to perform the search again and then write your results to the output file:

```
sm> search coha kwic word eq mouse
        status: success

sm> results file
rows written to file: 4,205

sm>
```

You can run multiple searches in a single session and have them all write out to the same session file; each time you use the command “results file” it will append each new search to the existing file (and not overwrite your previous searches). **However: if you close SearchMatrix and start a new session using the session file name, your data will be lost. Make sure to retrieve you session file to prevent it from being overwritten by another user.** Please note that searches that produce a large number of results will take longer to write to your file.

When you are done performing searches, enter the command “exit” at the sm> prompt:

sm> exit
[corpsearch@zeos ~] $

6. Other Search Options (from the 'Labels' section of each corpus listing)

Currently, the Corpus of Modern Scottish Writing (cmsw) only allows for the kwic (keyword in context) search type. The other three corpora (COCA, COHA, and GLOWBE) also have a lexicon list, and subset lists for lemmas and pos (Part-of-Speech) tags. Furthermore, these corpora have sources for the data, which can also be used as a search function.

If, for example, we wanted to see all of the sources that were used in compiling GLOWBE, we could run the following search:

```
sm> search glowbe sources
        status: success

sm> results 1
2       237     US      G       http://www.sheetmusicplus.com/title/There-Has-to-Be-a-Song/19495585
        There Has to Be a Song Sheet Music by Andrea Ramsey | Sheet
```

We could then use these sources to constrain our searches:

```
sm> search glowbe sources id eq 2
```

This would only return data that came from that specific source.

We could also see all of the part-of-speech (pos) tags used in COCA by running the following search:

```
sm> search coca pos
```

If we wanted to know the possible lemmas which we could use to search in COHA, we would run the following:

```
sm> search coha lemmas
```

Once we have this information, we can run constrained searches on the corpora to get instances of a specific lemma or part-of-speech. Currently, we do not have the option to combine searches so that you could get all instances of a word (such as 'meeting') and specify its part-of-speech (such as 'nn'). Instead, we would recommend that you perform a search for the word and then take your output to another program (such as R) where you could further subset the data by part-of-speech.

7. Retrieving your searches (Downloading the session-file)

To download your session-file (where your “results file” command(s) stored all of your search results),

you have several options:

1. If using a graphical SFTP client such as WinSCP or FileZilla (https://filezilla-project.org/), you can use the interface to copy the file from Zeos onto your local machine.
2. If you are comfortable using the command line to download your file, you can use the scp command (if you are not already familiar with doing this, we would recommend installing either WinSCP or FileZilla, connecting to corpsearch@zeos from there, and using (1) above).

When done, type exit into the terminal to end your connection to Zeos and/or close your SFTP client.

Please make sure that you download your output (session-file) files; over time, we will need to remove these from the server and cannot guarantee that another user may (inadvertently) append additional files to your session-file.

8. Advanced Searches: 'like' and 'match'

The search restrictors 'like' and 'match' allow for the user to do more complex and advanced searches in SearchMatrix.

'**like**' allows for searching the corpus for a string that either begins or ends with the characters specified to the left or right (but *not both*- for such searches you will use 'match') of the % character. For example, if you wanted to find all examples of words in COCA that have the prefix octo-, you could use the following search:

```
sm> search coca lexicon word like octo%
      status: success
```

We can then view these results; for this example, you can check if you get the same number of lines written to your session file:

```
sm> results file
rows written to file: 299
```

If you wanted to search COHA for the number of words that end with the suffix -ment, you could use the following search:

```
sm> search coha lexicon word like %ment
     status: success

sm> results file
rows written to file: 9,913
```

Note that there are some notable shortcoming to the 'like' search option: while we got 9,913 instances of words ending in -ment, we did *not* get plural forms. For example, this search will return judgment, but not judgment. In order to do such a search, we would need something more powerful, like a regular expression (regex) search. This can be accomplished with the 'match' search option.

‘**match**’ allows for (most) regular expression searches on the corpora available in SearchMatrix. There is an important caveat here: any regex search that begins with * or ^* will cause the program to crash and exit. This is a known bug that is in the process of being addressed. Note that searches that begin with .* will work.

Let’s say that we were interested in all of the lemmas contained in COCA that had t as the third letter, and would be found in the first half of the dictionary. We could use the following search:

```
sm> search coca lemmas word name ^[a-m].t
    status: success

sm> results file
rows written to file: 67,907
```

If you encounter any regex searches that cause SearchMatrix to quit or crash, please send the search to linglab@uw.edu and lxlabsa@uw.edu .

Appendix

Restrictors

Below are some example restrictors for performing your searches:

```
sm> search glowbe lemmas name eq any
Returns all lemma items with a name equal to any (spelling must be lower case)

sm> search glowbe lemmas name neq any
Returns all lemma items where the name is not equal to any

sm> search glowbe lemmas name eqic any
Returns all lemma items named any, irrespective of case.

sm> search glowbe lemmas name like anym%
Returns all lemmas that begin with 'anym'

sm> search glowbe lemmas name match .*anym.*
RegEx search in lemmas for the glowbe corpus.
```

Restricting Sources (GLOWBE)

Now we can restrict glowbe sources,

```
sm> corpus glowbe

search
        label: sources
        name: Sources Search
```

description: search the source set

fields: id, words, country, genre, url, title

integer fields, like ids, can be restricted to numeric targets:

sm> search glowbe sources id eq 100

sm> results all

100 316 CA G http://lasvegasattractions.org/ Las Vegas Attractions | Discover What to Do In Las Vegas

sm> search glowbe sources title like Las%
status: success

sm> results file

rows written to file: 999
sm>

Multiple Constraints (Not Currently Supported)
It is hoped that a future version of the CLI will allow for multiple constraints in a single search, such as:
search cmsw kwic word eq the and word_follow01 eq big and word_follow02 eq boat