

Command-line Magic

Or Why I Can't Abandon Perl (Just Yet)

Na-Rae Han
09/17/2016

Agenda

- ▶ Text-processing on the fly, using unix utilities
- ▶ Unix utilities:
 - ◆ cd, ls, cat, more
 - ◆ Piping: |, >, >>, <
 - ◆ tr, sed
 - ◆ grep --color, grep -P
 - ◆ wc
 - ◆ uniq -c, sort
 - ◆ head, tail
 - ◆ perl
 - ◆ for loop (Bash shell)

Na-Rae's environment

▶ My setup:

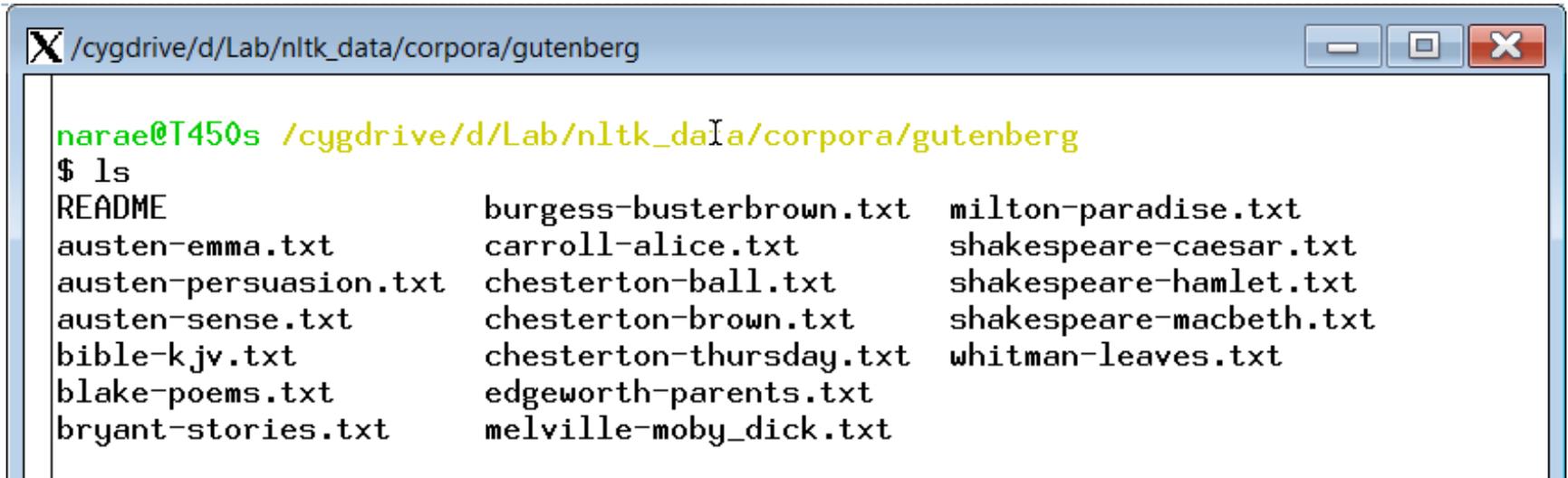
- ◆ Cygwin 1.7 running on Windows 7
 - ◆ <http://www.cygwin.com/>
- ◆ X11 windows manager
 - ◆ <http://x.cygwin.com/>
- ◆ Bash shell
 - ◆ [http://en.wikipedia.org/wiki/Bash \(Unix shell\)](http://en.wikipedia.org/wiki/Bash_(Unix_shell))

words file (Unix)

- ▶ [https://en.wikipedia.org/wiki/Words \(Unix\)](https://en.wikipedia.org/wiki/Words_(Unix))
 - ◆ words is a standard file on all Unix and Unix-like operating systems, and is simply a newline-delimited list of dictionary words.
 - ◆ It is usually found as: `/usr/share/dict/words`

- ▶ Alternatively, you can get the file via:
 - ◆ <http://www.pitt.edu/~naraehan/ling1330/words> (download & save)
 - ◆ **PyLing members with a PSC account** can access the file at:
 - ◆ `/home/naraehan/words`

Gutenberg Corpus



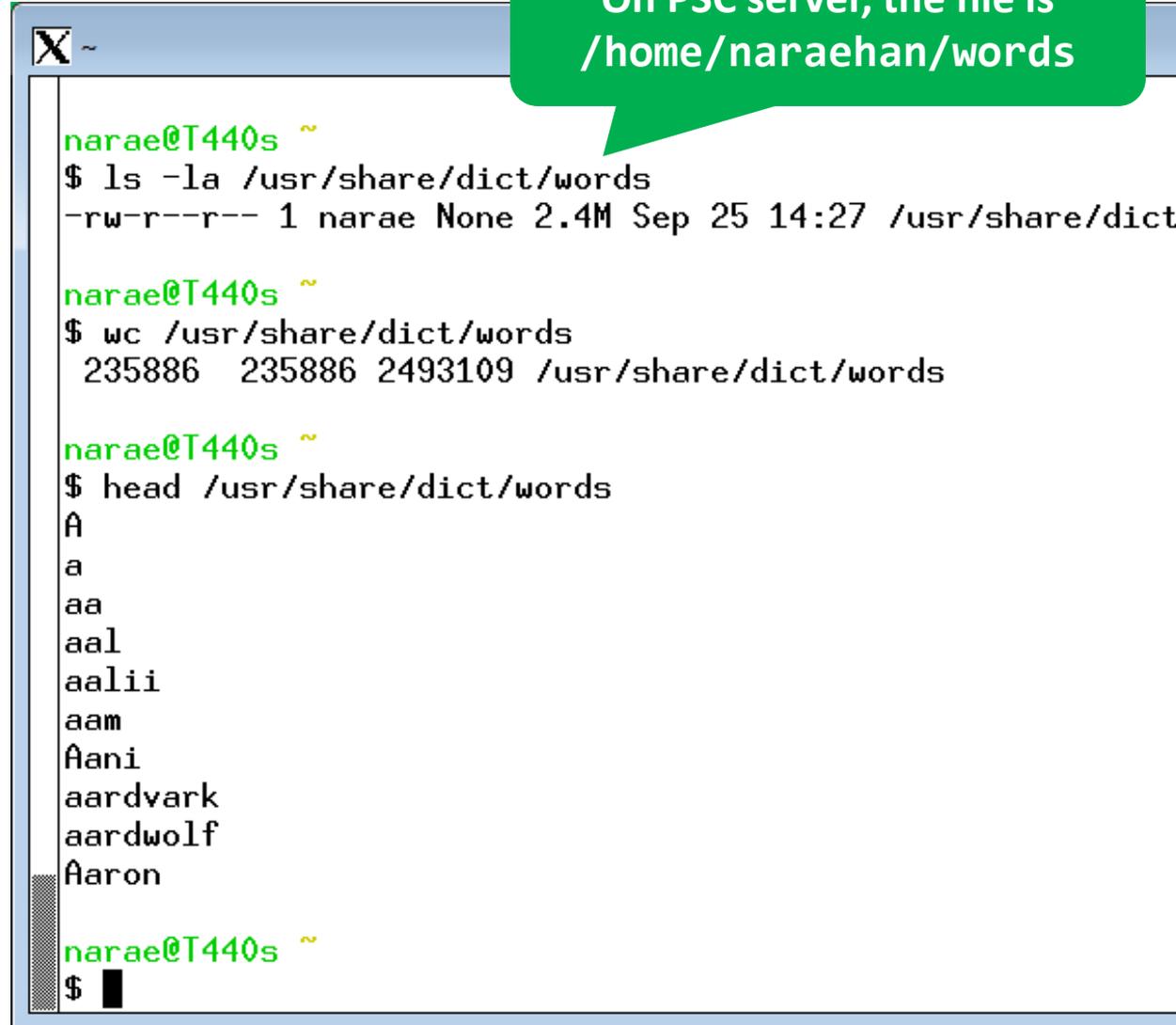
```
X /cygdrive/d/Lab/nltk_data/corpora/gutenberg
narae@T450s /cygdrive/d/Lab/nltk_data/corpora/gutenberg
$ ls
README                burgess-busterbrown.txt  milton-paradise.txt
austen-emma.txt       carroll-alice.txt        shakespeare-caesar.txt
austen-persuasion.txt chesterton-ball.txt      shakespeare-hamlet.txt
austen-sense.txt      chesterton-brown.txt     shakespeare-macbeth.txt
bible-kjv.txt         chesterton-thursday.txt  whitman-leaves.txt
blake-poems.txt       edgeworth-parents.txt
bryant-stories.txt   melville-moby_dick.txt
```

"Project Gutenberg Selections"

- ▶ Includes public-domain texts in their entirety
 - ◆ Jane Austen's *Emma*, the Bible, *Hamlet*, *Moby Dick*...
- ▶ Distributed as an NLTK data package
 - ◆ Download from: http://www.nltk.org/nltk_data/
- ▶ **PyLing members with a PSC account** can find the files at:
 - ◆ /usr/share/nltk_data/corpora/gutenberg

Examining a text file

- ▶ **ls -la**
 - ◆ Displays file info
- ▶ **WC**
 - ◆ Displays line count, word count, and character count
- ▶ **head -n**
 - ◆ Displays initial n lines
- ▶ **tail -n**
 - ◆ Displays last n lines



```
X ~
narae@T440s ~
$ ls -la /usr/share/dict/words
-rw-r--r-- 1 narae None 2.4M Sep 25 14:27 /usr/share/dict/words

narae@T440s ~
$ wc /usr/share/dict/words
235886 235886 2493109 /usr/share/dict/words

narae@T440s ~
$ head /usr/share/dict/words
A
a
aa
aal
aalii
aam
Aani
aardvark
aardwolf
Aaron

narae@T440s ~
$ █
```

On PSC server, the file is
/home/naraehan/words

grep

▶ grep

- ◆ Searches each line in text for regular expression match

▶ grep -P

- ◆ Accepts perl-style regular expressions
- ◆ Perl-style == Python-style

```
X /usr/share/dict
narae@T440s /usr/share/dict
$ grep '^z.*z$' words
zizz

narae@T440s /usr/share/dict
$ grep '^a.*z$' words
abuzz
adz
alveloz
aveloz

narae@T440s /usr/share/dict
$ grep -P '[aeiou]{5,}' words
cadiueio I
Chaouia
euouae
Guauaenok

narae@T440s /usr/share/dict
$ █
```

Words with 5+ consecutive "vowel"s

grep -i, -v

- ▶ **grep -i**
 - ◆ ignores case
- ▶ **grep -v**
 - ◆ prints lines that DO NOT match

```
X /usr/share/dict
narae@T440s /usr/share/dict
$ grep -i 'q' words | grep -v 'u'
Iraq
Iraqi
Iraqian
miqra
nastaliq
Pontacq
Q
q
qasida
qere
qeri
qintar
Qoheleth
qoph
Saqib
shoq
Tareq

narae@T440s /usr/share/dict
$ █
```

Words that contain 'q'
but with no 'u'

Pipelines and I/O redirections

|

- ◆ Pass the output of one command to another for further processing

>

- ◆ Redirect the command-line output to a file. (Overwrites any existing file.)

>>

- ◆ Append the output to the end of an existing file.

<

- ◆ Read from a file and feed the content as the command-line input.

grep and pipelines

X /usr/share/dict

```
narae@T440s /usr/share/dict
$ grep '^un.*able$' words | wc -l
1119
```

Pipe into wc -l to count

```
narae@T440s /usr/share/dict
$ grep '^un.*able$' words > able.txt
```

Write out to a file

```
narae@T440s /usr/share/dict
$ tail -5 able.txt
unwordable
unworkable
unwoundable
unwrinkleable
unwritable
```

Take a look at the last 5 lines of file

```
narae@T440s /usr/share/dict
$ grep '^in.*able$' words >> able.txt
```

Append new search result to file

```
narae@T440s /usr/share/dict
$ tail -5 able.txt
inviolable
ininvitable
invitrifiable
invocable
invulnerable
```

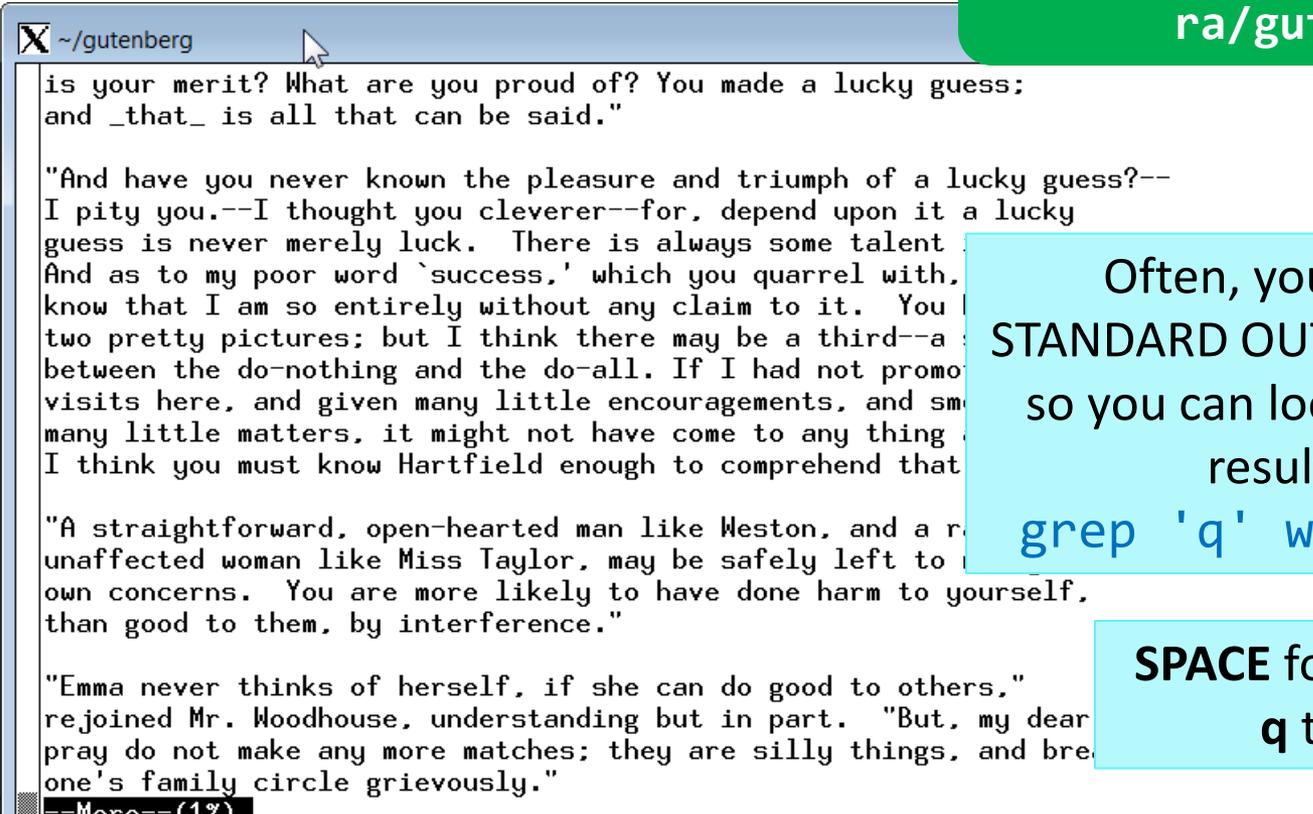
Take a look at the last 5 lines of file

more

- ▶ **more** pages through a text file content, one screen-full at a time. Press **SPACE** for next page, **q** to quit.

```
narae@T440s ~/gutenberg  
$ more austen-emma.txt
```

On PSC server, the directory is
`/usr/share/nltk_data/corpora/gutenberg`



```
~/gutenberg  
is your merit? What are you proud of? You made a lucky guess;  
and _that_ is all that can be said."  
  
"And have you never known the pleasure and triumph of a lucky guess?--  
I pity you.--I thought you cleverer--for, depend upon it a lucky  
guess is never merely luck. There is always some talent.  
And as to my poor word `success,' which you quarrel with,  
know that I am so entirely without any claim to it. You  
two pretty pictures; but I think there may be a third--a  
between the do-nothing and the do-all. If I had not prom  
visits here, and given many little encouragements, and sm  
many little matters, it might not have come to any thing  
I think you must know Hartfield enough to comprehend that  
  
"A straightforward, open-hearted man like Weston, and a r  
unaffected woman like Miss Taylor, may be safely left to  
own concerns. You are more likely to have done harm to yourself,  
than good to them, by interference."  
  
"Emma never thinks of herself, if she can do good to others,"  
rejoined Mr. Woodhouse, understanding but in part. "But, my dear  
pray do not make any more matches; they are silly things, and bre  
one's family circle grievously."  
--More-- (1%)
```

Often, you **pipe** your
STANDARD OUTPUT into **more**,
so you can look through the
result, e.g.,
`grep 'q' words | more`

SPACE for next page
q to quit

cat

- ▶ **cat** concatenates text file content and prints on the standard output.
 - ◆ Often used as the first step of piping.
 - ◆ Also useful in concatenating multiple file contents.

```
narae@T440s ~/gutenberg
```

```
$ ls *.txt
```

```
austen-emma.txt          burgess-busterbrown.txt  melville-moby_dick.txt
austen-persuasion.txt    carroll-alice.txt        milton-paradise.txt
austen-sense.txt         chesterton-ball.txt      shakespeare-caesar.txt
bible-kjv.txt            chesterton-brown.txt     shakespeare-hamlet.txt
blake-poems.txt          chesterton-thursday.txt  shakespeare-macbeth.txt
bryant-stories.txt       edgeworth-parents.txt   whitman-leaves.txt
```

```
narae@T440s ~/gutenberg
```

```
$ cat *.txt | grep 'purely'
```

```
enemies: 1:25 And I will tu e, and purely purge away
rhetoric and the rhetoric o a purely verbal
"To my friends--I myself should always prefer weapons purely intellectual.
purely intellectual conspiracy would soon threaten the very
purely. I came up with all speed (taking breath). Our pie is in the
purely, and to find you up so hearty--
```

Concatenating all
text files (*.txt)
into grep

Typos? Can't remember stuff?

- ▶ There are built-in shortcuts and help.
 - ▶ **Command-line history**
 - ◆ BASH shell remembers all commands you used in a session.
 - ◆ **↑** (UP arrow: brings up **previous command**)
 - ◆ **↓** (DOWN arrow: brings up **next command**)
- ← When you make a mistake and get an error message, don't re-type the whole thing! Bring up the previous command and then edit it.
- ← Need to repeat a command? Again, recycle with **↑** and **↓**.

```
narae@T450s /usr/share/dict
$ grep '^un.*able$' words | wcc -l
bash: wcc: command not found
```

Oops, typo

```
narae@T450s /usr/share/dict
$ grep '^un.*able$' words | wcc -l
```

↑, then fix it

Typos? Can't remember stuff?

▶ **TAB completion**

- ◆ Nobody can remember long commands or file names.
- ◆ Type in the first few characters, then hit **TAB**.
 - ◆ If a unique match is found, it auto-completes. Saves precious typing!
 - ◆ If not, you hear "beep". Hit TAB again, and all matching entries are shown to you.

```
narae@T450s ~/gutenberg
$ more austen-
austen-emma.txt          austen-persuasion.txt  austen-sense.txt
```

TAB x 2
Shows all 3 files starting with 'austen-'

```
narae@T450s ~/gutenberg
$ more austen-emm
```

TAB now completes to 'austen-emma.txt'

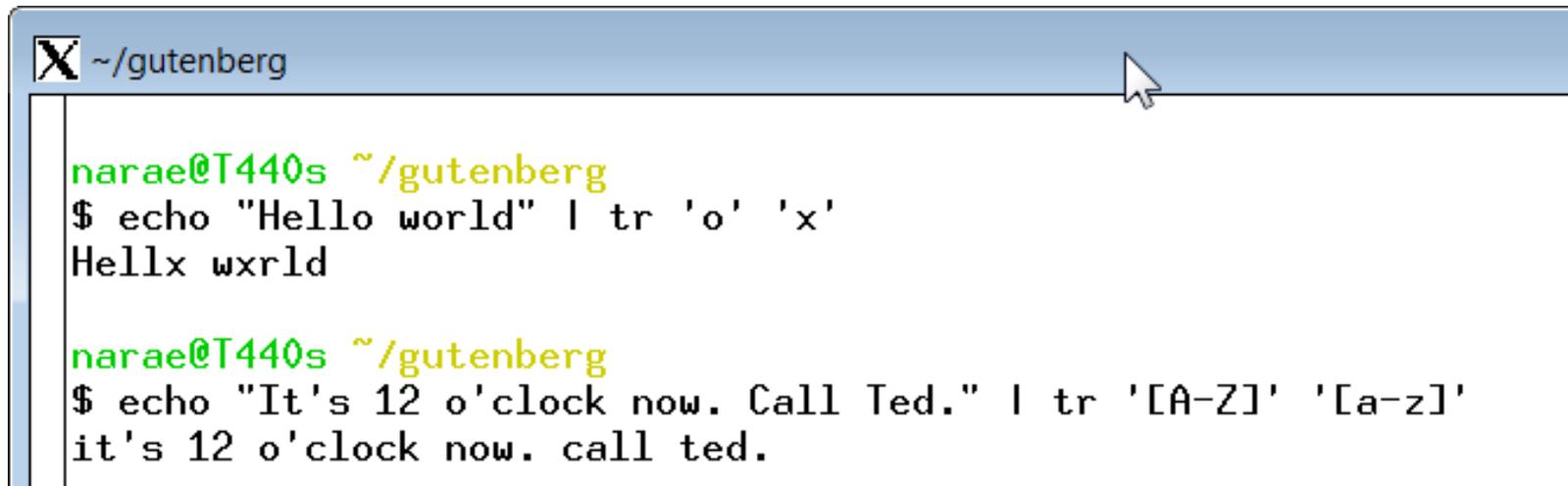
▶ **Ctrl + C** to the rescue

- ◆ Stuck? Control + C quickly gets you out and back to shell prompt.

tr

▶ tr

- ◆ Translates a **character** to another.
- ◆ Can also specify two sets of characters.

A terminal window with a blue title bar containing a window icon and the text '~ /gutenberg'. The terminal shows two command-line sessions. The first session shows the command 'echo "Hello world" | tr 'o' 'x'' resulting in the output 'Hellx wxrld'. The second session shows the command 'echo "It's 12 o'clock now. Call Ted." | tr '[A-Z]' '[a-z]'' resulting in the output 'it's 12 o'clock now. call ted.'.

```
X ~ /gutenberg
narae@T440s ~ /gutenberg
$ echo "Hello world" | tr 'o' 'x'
Hellx wxrld

narae@T440s ~ /gutenberg
$ echo "It's 12 o'clock now. Call Ted." | tr '[A-Z]' '[a-z]'
it's 12 o'clock now. call ted.
```

Lowercases all capital letters

sed

▶ sed

- ◆ short for 'stream editor'
- ◆ transforms a string into another

▶ sed -r

- ◆ enables regular expressions

```
X ~/gutenberg
narae@T440s ~/gutenberg
$ echo 'Ooooooh, what a wonderful book.' | sed -r 's/o+/_/g;'
O_h, what a w_nderful b_k.

narae@T440s ~/gutenberg
$ echo "It's 12 o'clock now...! Call Ted." | sed -r 's/\W+/ /g'
It s 12 o clock now Call Ted
```

Putting it all together

► Tokenizing *Emma* in one-line

```
narae@T440s ~/gutenberg
$ wc austen-emma.txt
16823 158167 887071 austen-emma.txt
```

16823 lines, 158167 words,
887K characters

```
narae@T440s ~/gutenberg
$ tr '[A-Z]' '[a-z]' < austen-emma.txt | sed -r 's/\W+/\n/g' | grep -P '\S'
> austen-emma.toks
```

```
narae@T440s ~/gutenberg
$ head austen-emma.toks
emma
by
jane
austen
1816
volume
i
chapter
i
emma
```

(1) `tr` folds case
(2) `sed` replaces non-
word chars with an
empty new line
(3) `grep` filters in
non-empty lines

Command-line perl can do it all

▶ perl -npe

- ◆ -e for command-line script, -n for line-by-line processing, -p for printing
- ◆ tr syntax in perl: `tr/x/y/;`
- ◆ sed syntax in perl: `s/abc/def/g;` (g for 'global replacement')

```
narae@T440s ~/gutenberg
$ perl -npe 'tr/[A-Z]/[a-z]/; s/\W+/\n/g;' austen-emma.txt | grep -P '\S' >
austen-emma.toks2
```

```
narae@T440s ~/gutenberg
$ wc -l austen-emma*
161983 austen-emma.toks
161983 austen-emma.toks2
 16823 austen-emma.txt
340789 total
```

Two token files are exactly the same!

```
narae@T440s ~/gutenberg
$ diff austen-emma.toks austen-emma.toks2
```

```
narae@T440s ~/gutenberg
```

Obtaining frequency counts

- ▶ First, you sort the word list using `sort`...

```
$ head -10 austen-emma.toks  
emma  
by  
jane  
austen  
1816  
volume  
i  
chapter  
i  
emma
```

```
$ head -10 austen-emma.toks | sort  
1816  
austen  
by  
chapter  
emma  
emma  
i  
i  
jane  
volume
```

First 10 tokens of Emma,
for illustration

Obtaining frequency counts

- ▶ And then `uniq -c` collapses adjoining identical lines while counting...

```
narae@T440s ~/gutenberg
$ head -10 austen-emma.toks | sort | uniq -c
  1 1816
  1 austen
  1 by
  1 chapter
  2 emma
  2 i
  1 jane
  1 volume
```

First 10 tokens of Emma,
for illustration

Obtaining frequency counts

- ▶ And finally, sort by reverse numerical order using `sort -nr`.

```
narae@T440s ~/gutenberg
$ sort austen-emma.toks | uniq -c | sort -nr | head -10
 5239 to
 5201 the
 4896 and
 4291 of
 3178 i
 3129 a
 2528 it
 2469 her
 2398 was
 2340 she
```

Top 10 word frequency counts!

Instead of `head -10`, use `more` to page through the entire list (SPACE for next page, q to quit)

Producing bigrams, on the fly

- ▶ Use `head` and `tail` to produce word1 and word2 token lists, and then `paste` the two into tab-separated bigram lines.

```
narae@T440s ~/gutenberg
$ wc -l austen-emma.toks
161983 austen-emma.toks
```

```
narae@T440s ~/gutenberg
$ head -161982 austen-emma.toks > austen-emma.toks.w1
```

```
narae@T440s ~/gutenberg
$ tail -n +2 austen-emma.toks > austen-emma.toks.w2
```

```
narae@T440s ~/gutenberg
$ paste austen-emma.toks.w1 austen-emma.toks.w2 > austen-emma.bigrams
```

```
narae@T440s ~/gutenberg
$ head -5 austen-emma.bigrams
emma      by
by        jane
jane      austen
austen    1816
1816     volume
```

`tail -n +2`
all lines starting
from the 2nd line

Bigram frequencies, on the fly

- ▶ Then, produce bigram frequencies:

```
narae@T440s ~/gutenberg
```

```
$ sort austen-emma.bigrams | uniq -c | sort -nr | head -10
```

```
607 to      be
566 of      the
448 it      was
446 in      the
395 i       am
332 she     had
328 she     was
308 had     been
299 mr      knightley
299 it      is
```

Tokenizing every file

- ▶ Use for loop in Bash shell.

```
narae@T450s ~/gutenberg
$ for file in *txt
> do
> perl -npe 'tr/[A-Z]/[a-z]/; s/\W+/\n/g;' $file | grep -P '\S' > $file.toks
> echo $file finished.
> done
austen-emma.txt finished.
austen-persuasion.txt finished.
austen-sense.txt finished.
bible-kjv.txt finished.
blake-poems.txt finished.
bryant-stories.txt finished.
burgess-busterbrown.txt finished.
carroll-alice.txt finished.
chesterton-ball.txt finished.
chesterton-brown.txt finished.
chesterton-thursday.txt finished.
edgeworth-parents.txt finished.
melville-moby_dick.txt finished.
milton-paradise.txt finished.
shakespeare-caesar.txt finished.
shakespeare-hamlet.txt finished.
shakespeare-macbeth.txt finished.
whitman-leaves.txt finished.
```

```
for x in file-list
do
command1 $x
command2 $x
command3 $x
done
```

Every file in the directory with .txt extension is processed, token list is saved as \$file.toks

Frequency counts for the entire corpus

- ▶ Easy. Concatenate all .txt.toks files, and then pipe into the routine.

These are the token files

```
narae@T440s ~/gutenberg
$ ls *.txt.toks
austen-emma.txt.toks          chesterton-brown.txt.toks
austen-persuasion.txt.toks    chesterton-thursday.txt.toks
austen-sense.txt.toks         edgeworth-parents.txt.toks
bible-kjv.txt.toks           melville-moby_dick.txt.toks
blake-poems.txt.toks         milton-paradise.txt.toks
bryant-stories.txt.toks      shakespeare-caesar.txt.toks
burgess-busterbrown.txt.toks  shakespeare-hamlet.txt.toks
carroll-alice.txt.toks       shakespeare-macbeth.txt.toks
chesterton-ball.txt.toks     whitman-leaves.txt.toks
```

cat *.txt.toks concatenates all token files.

```
narae@T440s ~/gutenberg
$ cat *.txt.toks | sort | uniq -c | sort -nr | head
133583 the
 95442 and
 71267 of
 48057 to
 33960 a
 33580 in
 30265 i
 28798 that
 25857 he
 22303 it
```

Windows Users: Install Cygwin

- ▶ An introduction to Cygwin

 - ◆ <http://physionet.org/physiotools/cygwin/>

- ▶ YouTube Tutorials

1. Installation: <http://www.youtube.com/watch?v=XQKOQiiivyY>

2. Introduction to Cygwin

<http://www.youtube.com/watch?v=ISuP11gmFDA>

3. Basic BASH Practical Tutorial

<http://www.youtube.com/watch?v=68PEBV9NpL8>

4. More basic BASH

<http://www.youtube.com/watch?v=gia47XIY1oI>

Learn Unix

▶ Unix Tutorial for Beginners

- ◆ <http://www.ee.surrey.ac.uk/Teaching/Unix/index.html>

▶ Unix for Mac OS X Users

- ◆ A video tutorial on Lynda.com
- ◆ <http://www.lynda.com/Mac-OS-X-10-6-tutorials/Unix-for-Mac-OS-X-Users/78546-2.html>
- ◆ You can log in with your Pitt account
- ◆ Most of the content also applies to Cygwin