CS 124/LINGUIST 180
From Languages to Information

Unix for Poets (in 2013)
Christopher Manning
Stanford University
Unix for Poets
(based on Ken Church’s presentation)

• Text is available like never before
  • The Web
  • Dictionaries, corpora, email, etc.
  • Billions and billions of words

• What can we do with it all?
• It is better to do something simple, than nothing at all.
• You can do simple things from a Unix command-line
• DIY is more satisfying than begging for “help”
Exercises to be addressed

1. Count words in a text
2. Sort a list of words in various ways
   1. ascii order
   2. “rhyming” order
3. Extract useful info from a dictionary
4. Compute ngram statistics
5. Work with parts of speech in tagged text
Tools

- grep: search for a pattern (regular expression)
- sort
- uniq –c (count duplicates)
- tr (translate characters)
- wc (word – or line – count)
- sed (edit string -- replacement)
- cat (send file(s) in stream)
- echo (send text in stream)
- cut (columns in tab-separated files)
- paste (paste columns)
- head
- tail
- rev (reverse lines)
- comm
- join
- shuf (shuffle lines of text)
Prerequisites

• ssh into a corn
• cp /afs/ir/class/cs124/nyt_200811.txt
• man, e.g., man tr (shows command options; not friendly)
• Input/output redirection:
  • >
  • <
  • |
• CTRL-C
Exercise 1: Count words in a text

• Input: text file (nyt_200811.txt)
• Output: list of words in the file with freq counts
• Algorithm
  1. Tokenize(tr)
  2. Sort(sort)
  3. Count duplicates (uniq –c)
Solution to Exercise 1

- tr -sc ’A-Za-z’ \n’ < nyt_200811.txt | sort | uniq -c

- 25476 a
- 1271 A
- 3 AA
- 3 AAA
- 1 Aalborg
- 1 Aaliyah
- 1 Aalto
- 2 aardvark
Some of the output

- tr -sc ’A-Za-z’ ’\n’ < nyt_200811.txt | sort | uniq -c | head –n 5
  25476 a
  1271 A
  3 AA
  3 AAA
  1 Aalborg

- tr -sc ’A-Za-z’ ’\n’ < nyt_200811.txt | sort | uniq -c | head

- Gives you the first 10 lines
- tail does the same with the end of the input
- (You can omit the “-n” but it’s discouraged.)
Extended Counting Exercises

1. Merge upper and lower case by downcasing everything
   • Hint: Put in a second tr command

2. How common are different sequences of vowels (e.g., ieu)
   • Hint: Put in a second tr command
Sorting and reversing lines of text

- `sort`
- `sort -f` Ignore case
- `sort -n` Numeric order
- `sort -r` Reverse sort
- `sort -nr` Reverse numeric sort
- `echo "Hello" | rev`
Counting and sorting exercises

- Find the 50 most common words in the NYT
  - Hint: Use sort a second time, then head

- Find the words in the NYT that end in “zz”
  - Hint: Look at the end of a list of reversed words
Lesson

- Piping commands together can be simple yet powerful in Unix
- It gives flexibility.

- Traditional Unix philosophy: small tools that can be composed
Bigrams = word pairs counts

- Algorithm
  1. tokenize by word
  2. print $word_i$ and $word_{i+1}$ on the same line
  3. count
Bigrams

- tr -sc ’A-Za-z’ ’\n’ < nyt_200811.txt > nyt.words
- tail –n +2 nyt.words > nyt.nextwords
- paste nyt.words nyt.nextwords > nyt.bigrams
- head –n 5 nyt.bigrams

KBR said
said Friday
Friday the
the global
global economic
Exercises

• Find the 10 most common bigrams
  • (For you to look at:) What part-of-speech pattern are most of them?

• Find the 10 most common trigrams
grep

• Grep finds patterns specified as regular expressions
• grep rebuilt nyt_200811.txt

Conn and Johnson, has been rebuilt, among the first of the 222 move into their rebuilt home, sleeping under the same roof for the the part of town that was wiped away and is being rebuilt. That is to laser trace what was there and rebuilt it with accuracy," she home - is expected to be rebuilt by spring. Braasch promises that a the anonymous places where the country will have to be rebuilt, "The party will not be rebuilt without moderates being a part of
grep

- Grep finds patterns specified as regular expressions
  - globally search for regular expression and print

- Finding words ending in –ing:
  - `grep ’ing$’ nyt.words | sort | uniq -c`
grep

• grep is a filter – you keep only some lines of the input
• grep gh  
  keep lines containing “gh”
• grep ’^con’  
  keep lines beginning with “con”
• grep ’ing$’  
  keep lines ending with “ing”
• grep –v gh  
  keep lines NOT containing “gh”

• grep -P  
  Perl regular expressions (extended syntax)
• grep -P ’^[A-Z]+$’ nyt.words | sort | uniq –c  
  ALL UPPERCASE
Counting lines, words, characters

• `wc nyt_200811.txt`
  140000 1007597 6070784 nyt_200811.txt

• `wc -l nyt.words`
  1017618 nyt.words
grep & wc exercises

• How many all uppercase words are there in this NYT file?
• How many 4-letter words?
• How many different words are there with no vowels
  • What subtypes do they belong to?
• How many “1 syllable” words are there
  • That is, ones with exactly one vowel

Type/token distinction: different words (types) vs. instances (tokens)
**sed**

- sed is a simple string (i.e., lines of a file) editor
- You can match lines of a file by regex or line numbers and make changes
- Not much used in 2013, but
- The general regex replace function still comes in handy

- `sed 's/George Bush/Dubya/' nyt_200811.txt | less`
sed exercises

• Count frequency of word initial consonant sequences
  • Take tokenized words
  • Delete the first vowel through the end of the word
  • Sort and count

• Count word final consonant sequences
awk

- Ken Church’s slides then describe **awk**, a simple programming language for short programs on data usually in fields
- I honestly don’t think it’s worth learning awk in 2013
- Better to write little programs in your favorite scripting language, be that Python, or Perl, or groovy, or ...
shuf

- Randomly permutes (shuffles) the lines of a file

- Exercises
  - Print 10 random word tokens from the NYT excerpt
    - 10 instances of words that appear, each word instance equally likely
  - Print 10 random word types from the NYT excerpt
    - 10 different words that appear, each different word equally likely
cut – tab separated files

- `cp /afs/ir/class/cs124/parses.conll .`
- `head –n 5 parses.conll`

1. Influential _ JJ JJ _ 2 amod _ _
2. members _ NNS NNS _ 10 nsubj _ _
3. of _ IN IN _ 2 prep _ _
4. the _ DT DT _ 6 det _ _
5. House _ NNP NNP _ 6 nn _ _
cut – tab separated files

• Frequency of different parts of speech:
  
cut -f 4 parses.conll | sort | uniq -c | sort –nr

• Get just words and their parts of speech:
  
cut -f 2,4 parses.conll

• You can deal with comma separate files with: cut –d,
cut exercises

• How often is ‘that’ used as a determiner (DT) “that man” versus a complementizer (IN) “I know that he is rich” versus a relative (WDT) “The class that I love”
  • Hint: With grep –P, you can use ‘\t’ for a tab character

• What determiners occur in the data? What are the 5 most common?