Chapter 1. First Class of Linux and Lexical analysis (Borrow Alex’s case)

Jingbo Xia
College of Informatics, HZAU

http://www.tecmint.com/13-basic-cat-command-examples-in-linux/

Linux command cat:

The cat (short for “concatenate“) command is one of the most frequently used command in Linux/Unix like operating systems. cat command allows us to create single or multiple files, view contain of file, concatenate files and redirect output in terminal or files. In this article, we are going to find out handy use of cat commands with their examples in Linux.

http://www.linuxcommand.org/man_pages/tr1.html

Linux command tr:

```
        tr - translate or delete characters
SYNOPSIS
        tr [OPTION]... SET1 [SET2]
DESCRIPTION
        Translate, squeeze, and/or delete characters from standard
input, writing to standard output.

        -c, --complement
                first complement SET1
        -d, --delete
                delete characters in SET1, do not translate
        -s, --squeeze-repeats
                replace each input sequence of  a  repeated  character
that  is  listed in SET1 with a single occurrence of that character
        -t, --truncate-set1
                first truncate SET1 to length of SET2
```

http://www.thegeekstuff.com/2013/04/sort-files

Sort command is helpful to sort/order lines in text files. You can sort the data in text file and display the output on the screen, or redirect it to a file. Based on your requirement, sort provides several command line options for sorting data in a text file.

Sort Command Syntax:

```
$ sort [-options]
```

Case study:

Text complexity comparison of plain text and biomedical text

Corpora are borrowed from DSG group. http://dsg.ctl.cityu.edu.hk

Linux commands for lexicon

```
$cat pubmed-a.txt |tr -cs "[:alnum:]" "\n" |tr [:upper:] [:lower:]
>pubmed.a.pure.txt
$wc pubmed.a.pure.txt
1032975 1032975 6634213
$sort pubmed.a.pure.txt |uniq |wc
31638   31638  282238
```

So the token/word ratio is 31638/1032975 = 3.06%.

```
$cat BROWN_A.txt |tr -cs "[:alnum:]" "\n" |tr [:upper:] [:lower:]
>BROWN_A.pure.txt
$wc BROWN_A.pure.txt
91064   91063  515866
$sort BROWN_A.txt |uniq |wc
11927   11926   95306
```

So the token/word ratio is 11927/91064 =13.09%.

But this is an unfair comparison.

Linux commands for lexicon

```
$cat pubmed-a.txt |tr -cs "[:alnum:]" "\n" |tr [:upper:] [:lower:] |split -
1000
$ wc xaa
1000     999     5935 xaa
$sort xaa |uniq |wc
458     457     3307

$cat pubmed-a.txt |tr -cs "[:alnum:]" "\n" |tr [:upper:] [:lower:] |split -
2000
$wc xaa
2000     1999    11759 xaa
$ sort xaa |uniq |wc
782      781     5614
```

Assignment:

Currently, you know how to compute token/word ratio of a text and evaluate its lexical complexity.

Select texts with your interests sizes, analyze their differences, and do a comparison.

Find something novel and prepare a 10 minutes talk in next class. Two members win the points.