

Exercise: Basic Unix Tools and Corpus Frequencies

1 Task

Handle (plain) text files by means of basic Unix tools. Create a frequency list of the word types comprised in the corpus (see end of today's slides).

Additional tasks Create a rank / frequency profile and a frequency spectrum (see end of today's slides).

2 Tools

In this section you'll find an overview of the commands that are used in this exercise. All commands are to be typed in a shell and finished by pressing the 'enter' key.

There are three general ways of getting help (on Unix/Linux)

- `man <command>`
Displays the online reference manual of a command, e.g. 'man less' displays the manual of less.
- `<command> --help`
Displays the usage of a command, e.g. 'less --help' displays the usage of less.
- `info <command>`
Displays the full manual.

Command

Example

cp

Copies files and directories.

```
cp <source> <target>
```

```
cp grep10a-plain.txt dickens.txt
```

less

Paging through text files one screenful at a time. Pressing the 'space key' gives you the next page; 'q' makes you quit.

```
less <file>
```

```
less dickens.txt
```

Command	Example
<p>cat Concatenates files and prints them (line by line) to the standard output. cat <file> Useful Options: -n number all output lines -b number non-blank output lines -s squeeze more than one single blank line.</p>	<p>cat text1 text2 text3 cat -ns dickens.txt</p>
<p>pipe ' ' Combines sequences of commands. Output of the first command is 'piped' to the second command. <command1> <command2></p>	<p>cat dickens.txt less</p>
<p>print '>' Prints to a file. <command> > <output.file></p>	<p>cat dickens.txt > copy.txt</p>
<p>tr Translates characters defined in set 1 to corresponding character in set 2 and writes to standard output. tr <set1> <set2> Examples: Translate 'space' to 'newline' (= print a file one word per line) Translate lower-case 'abc' to upper-case Translate all lower-case characters to upper-case characters. Useful Options: -d deletes characters in <set1>, does not translate. Example: Delete all punctuation.</p>	<p>cat dickens.txt tr ' ' '\n' tr abc ABC tr a-z A-Z cat original.txt tr -d [:punct:]</p>
<p>sort Sorts lines of text files. sort <file></p>	<p>cat dickens.txt tr ' ' '\n' sort</p>

Command	Example
<p>sort: Useful options</p> <ul style="list-style-type: none"> -n compare according to numerical value -r reverse the result of comparison -k# sorts according to content in column # <p>Example:</p> <p>Sort list of numbers in reverse order.</p> <p>Sort list of numbers in column 1 in reverse order.</p>	<pre>sort -nr numbers.txt sort -k1 -nr numbers-column</pre>
<p>uniq</p> <p>Removes duplicate lines from a sorted file.</p> <p>Useful options:</p> <ul style="list-style-type: none"> -c prefix lines by the number of occurrences -i ignore differences in case when comparing 	<pre>cat dickens.txt \ tr ' ' '\n' sort uniq ... uniq -ci</pre>
<p>wc</p> <p>Print the number of bytes, words, and lines in files.</p>	<pre>wc dickens.txt</pre>
<p>gawk</p> <p>A (powerful) pattern scanning and text processing language.</p> <p>We use it only for extracting part of an input line.</p> <p>Example:</p> <p>Print content of column 1 to standard output.</p>	<pre>gawk cat <input file> \ '{print \$1}'</pre>

3 Data

The starting point is file 'gexp10a-plain.txt'. It is derived from an EText file from Project Gutenberg (<http://www.gutenberg.org>): Charles Dickens: "Great Expectations". It is a stripped down version of the original EText which included a header that is saved in an extra file (gexp10-info.txt) and also punctuation marks.

4 Procedure

1. **Copy** file 'gexp10a-plain.txt' to a new file named 'dickens.txt':

```
cp gexp10a-plain.txt dickens.txt
```
2. Page through file 'dickens.txt':

```
less dickens.txt
```
3. **Create a list of tokens**
Convert 'dickens.txt' to one-word-per-line format and print it to a new file 'dickens-tokens':

```
cat dickens.txt | tr ' ' '\n' > dickens-tokens
```


Check the content of the new file:

```
less dickens-tokens
```
4. **Create an alphabetically ordered list of tokens**
Open 'dickens-tokens' and sort it alphabetically and print it to a new file 'dickens-tokens-sorted':

```
cat dickens-tokens | sort > dickens-tokens-sorted
```


Check the content of the new file:

```
less dickens-tokens-sorted
```
5. Do the same thing again but sort the list in **reverse order**:

```
cat dickens-tokens | sort -r > dickens-tokens-sorted.
```


Check the result.
6. **Create a list of types**
Open 'dickens-tokens-sorted', remove duplicate lines and print the output to a new file 'dickens-types':

```
cat dickens-tokens-sorted | uniq > dickens-types
```


Check the result:

```
wc dickens-types: 13079 13078 110368 dickens-types
```
7. **Create a frequency list**
Open 'dickens-tokens-sorted', remove duplicate lines, count the number of occurrences and print the output to a new file 'dickens-freq-list':

```
cat dickens-tokens-sorted | uniq -c > dickens-freq-list
```


Format:
<frequency> <type> (ordered alphabetically according to type; 13079 types)
8. Create a **rank/frequency profile**

```
cat dickens-freq-list | gawk '{print $1}' | sort -nr | cat -b  
> dickens-rank-freq-profile
```

Format:

<rank> <frequency 'tokens'> (ordered according to frequency (starting with highest frequency; 13079 entries).

Look at the top of the list. Rank/frequency profiles are useful to study the properties of **high frequency items**.

9. Create a frequency spectrum

How many different frequency values are there?

```
cat dickens-freq-list | gawk '{print $1}' | sort -nr | uniq  
|wc: 266 different frequencies
```

```
cat dickens-freq-list | gawk '{print $1}' | sort -n | uniq -c  
|\  
sort -k2 -n | gawk '$1 $2 {print $2"\t"$1}' > dickens-freq-spectrum
```

Format:

<frequency> <occurrence of frequency> (ordered according to frequency; starting with lowest frequency; 266 entries).

Frequency spectra are useful to study the properties of **low frequency items**.

5 Alternative Tools

- Unix tools for windows: <http://www.XXX>
- CygWin (requires XXX): <http://www.XXX>

6 References

- Charles Dickens: "Great Expectations". Project Gutenberg (<http://www.gutenberg.org>), EText-No. 1400, Release-date: 1998-07-01, file: grexp.10.txt