Lecture 9 Practical exercises with linguistically annotated data 21 Nov 2012

We work with the file parlamentocorpus.100k.vrt, a sample from the CRPC corpus. The file has a column format, the first column is a token or word form, the second its automatically predicted POS-tag and the third the lemma of the token.

NB. The columns in the corpus file are not separated by spaces but by tabs.
To type a tab character on the command line: CTRL+V followed by the key TAB.

1. Have a look at the contents of the corpus file.
1a. What commands can you use to look at the contents on the screen?

`$ head parlementcorpus.100k.vrt` → *gives you the first 10 lines*

`$ tail parlementcorpus.100k.vrt` → *gives you the last 10 lines*

`$ cat parlementcorpus.100k.vrt` → *gives you the full text at once*

`$ more parlementcorpus.100k.vrt` → *allows to scroll through the text ('q' to stop)*

`$ less parlementcorpus.100k.vrt` → *allows you to scroll through the text*

$ cut -f n[,m] parlementcorpus.100k.vrt → *allows to see only column n to m of the text*

2. Do some first counts on the corpus.
2a. How many texts are there in the corpus?

```
$
$
```

As '<' is a special character on the command line, it means 'read from input', we need to use double quotes “ around the first argument of the command 'grep' to tell grep that we use '<' literally as normal plain text and not as a special character.

Alternatively we can use 'grep -c 'that stands for “count the matches”.

```
$
```

2b. How many sentences are there in the corpus?

```
$
```

Or

```
$
```

2c. How many words are there in the corpus?

Get all lines that contain a tab and count them:

```
$
```

Or count the lines that do not contain XML markup. Use “grep -v” to get non-matching lines.

```
	$
```

3. Make a sorted frequency list of the determiners. In other words: a list of all words that have as part-of-speech tag "DA". (We do this step by step.)

3a. How do you get only those lines that contain a determiner ?

Use grep to get the lines contain the POS-tag "DA".

```
	$
```

3b. How many word forms (tokens) have “DA” as POS-tag?

```
	$
```

Why do we use wc -l and not wc -w ?

The command wc -w just counts every item separated by delimiters. The file DA_list also contains the POS_tags and lemmas, which will also be counted. However, the file only contains one word per line so we can use wc -l to count correctly.

3c. How do you make a frequency list with only determiners that looks like this?

```
 203 A      DA     a
  15 Ao     PREP+DA      a+o
  72 As     DA     a
   1 DAS    PREP+DA      de+a
  19 Da     PREP+DA      de+a
   5 Das    PREP+DA      de+a
  18 Do     PREP+DA      de+o
   4 Dos    PREP+DA      de+o
.....
```

```
	$
```

3d. How many word forms (types) have DA as POS-tag, in other words, how many lines does your sorted frequency list count?

```
	$
```

3e. The current list contains besides word forms also the columns with the POS and lemma information. How can we get a frequency list with only one column of word forms that looks like this?

```
 203 A
  15 Ao
  72 As
   1 DAS
  19 Da
   5 Das
  18 Do
   4 Dos
...
```

```
	$
```

3f. How do you make a numerical sort so that you can sort the list made in 3e. on its frequency rank like this?

```
…
 575 no
 640 os
 693 dos
1385 da
1492 do
1511 o
1776 a
```

Use sort but with the parameter "-n" which stands for 'numeric sort'.

```
	$
```

or all together:

```
	$
```

3g. How would you reverse the ordering of the sorting: highest frequent determiner first like this?

```
1776 a
1511 o
1492 do
1385 da
 693 dos
 640 os
 575 no
…
```

```
	$
```

4. The list of determiners contains also the words that are contracted word forms of a determiner and a preposition, for example 'da	PREP+DA'.
4a. How can you only keep the pure determiners and exclude the contracted word forms?
(Hint: To type a tab character: CTRL+V followed by the key TAB.)

```
	$
```

4b. Other solution: How do you convert the tabs to spaces?

```
	$
```

5. The list of determiners now contains words in lowercase and words with uppercase letters.
5a. How can you transform all letters to lower case ?

```
	$
```

6. Can you now combine different commands to reproduce the following frequency list?

```
1978 a
1920 o
 703 os
 567 as
```

```
        $
```

get only the pure determiners | *cut to keep the first column* | *make all lower case*| *sort* | *keep only unique copy* | *and then sort on the frequency value.*

7. Making your own concordances

You want to look at particular words in their context to study their usage in the text.
In this example we pretend to be linguists studying conjunctions in Portuguese, and today we want to study the behavior of the word 'como'.   The final result of exercise 7 is a list of concordances of the conjunction 'como' with 3 words left and right, and we will do this step by step.

7a. Make a frequency list of the word 'como' to get a first impression of the occurrences of this word in the corpus (use the command of 3c.).

```
        $
```

7b. How can you retrieve the context of a word?
You can use `grep` to take the lines before and after a match:

```
        $ grep -An  : grep with n lines after the match
        $ grep -Bn: grep with n lines before the match
```

Use this to get 3 lines before and after ‘como’.

```
        $
```

7c. How do you get 3 words before and after ‘como’? How do you get word forms only?

```
        $
```

7d. Can you think of a way to get one concordance of the word 'como' per line, like in this example:

```
 e elucidativo , como completo e bem
  falibilidade humana , como também se ressentem
  poderá aspirar , como supremo anelo da
  de direito , como frisou a Câmara
```

(Hint, this one is difficult, think of the command ‘tr’ .)

What separates the lines that you get in 7c. from the format that you see in the concordances 7d. ? The words are separated by newlines in 7c. and by spaces in 7d. so you need to convert the newlines into spaces.

What character separates the concordances from each other in 7c? The '-' character.

```
        $
```

Every concordance was separated from the others with two '-'. So using tr '-' '\n' means that every concordance is now separated by two newlines. How to get only one newline between every concordance?

With tr -s: Squeeze multiple occurrences of a character into one.

```
        $
```