Data analysis

Goals

- Be able to carry out simple text analysis tasks in the command line
- **Be able to create a corpus in TXM**
- Learn some of the basic functionalities of TXM for textual analysis: cooccurrences, specificity index, create partitions and subcorpora

Required software

- TXM and the TreeTagger extension

Importing files in TXM

Task: Create a corpus in TXM using as source the ELTEC-eng corpus available in the folder “datasets/tei”

1. Modify the file `ELTeC-eng_metadata.tsv` accordingly, so the metadata can be read in TXM
    - It must be a CSV file, with a comma as delimiter and named `metadata.csv`
    - The first column must be named “`id`” and contain the filenames (without extension)
2. Import the corpus with the option XML/w + CSV

Goals

- **Be able to carry out simple text analysis tasks in the command line**
- Be able to create a corpus in TXM
- Learn some of the basic functionalities of TXM for textual analysis: cooccurrences, specificity index, create partitions and subcorpora

Count

Command:

```
wc [OPTION] [INPUT]
```

- It is used to find out number of lines, word count, byte and characters count in the files specified in the input arguments.
- By default, it displays four-columnar output.
    - 1 = number of lines
    - 2 = number of words
    - 3 = number of characters
    - 4 = filename

Count

In the folder “datasets/plain-text/disco-19th” run the following commands:

1. `wc *`
2. `wc * > ../count.csv` (to print the results in a file)
3. `wc -w *` (to just count the number of words per file)
4. `wc -l *` (to just count the number of lines per file)

Find strings

Command:

```
grep [OPTION] [PATTERN] [INPUT]
```

- Searches for a pattern in a file.

Find strings

In the folder “datasets/plain-text/disco-19th” run the following commands:

1. `grep "amor[a-z]" *`
2. `grep "\sroj[ao]" *`
3. `grep "\Sroj[ao]" *`
4. `grep -l "\Sroj[ao]" *` (to just output the filename where the string occurs)
5. `grep "^yo" *`
6. `grep -y "^yo" *` (to ignore the case)
7. `grep -v -y -n "s" disco140n.txt` (`-v` displays the lines that do not match the pattern; `-n` adds the line number)

Get unique words

Commands:

`uniq [OPTION] [INPUT[OUTPUT]]`

- It reports and filters out the repeated lines in a file.

`sort [INPUT FILE]`

- It sorts a file, arranging the records in a particular order.

Get unique words

In the folder “datasets/plain-text/word-list” run the following commands:

1. `sort mrs-dalloway_woolf.txt | uniq > mrs-dalloway_unique-words.txt`
2. `sort mrs-dalloway_woolf.txt | uniq -i > mrs-dalloway_unique-words.txt`
3. `sort mrs-dalloway_woolf.txt | uniq -ic > mrs-dalloway_unique-words-count.txt`
4. `sort mrs-dalloway_woolf.txt | uniq -id | uniq -i > mrs-dalloway_repeated-words.txt`

Goals

- Be able to carry out simple text analysis tasks in the command line
- Be able to create a corpus in TXM
- **Learn some of the basic functionalities of TXM for textual analysis: cooccurrences, specificity index, create partitions and subcorpora**

CQL

- Corpus Query Language = enables the search for complex grammatical or lexical patterns by accessing the properties annotated in each token
  - It supports regular expressions
- TXM offers help building CQL expressions, so you can carry out many queries without knowing the language.

CQL syntax

- Criteria for each token must appear between a pair of square brackets with the format:
  - `[attribute="value"]`
- To search for a phrase, each token must appear in its own pair of square brackets:
  - `[lemma="add"][word="salt"][word="to"][pos="NOUN"]`
- Square brackets `[ ]` stand for ‘any token’. Curly brackets `{ }` are used for repetition of the preceding token.
  - `[lemma="refill"] [ ] [lemma="teapot"]`
  - `[lemma="have"] [ ]{2,4}[lemma="opinion"]`

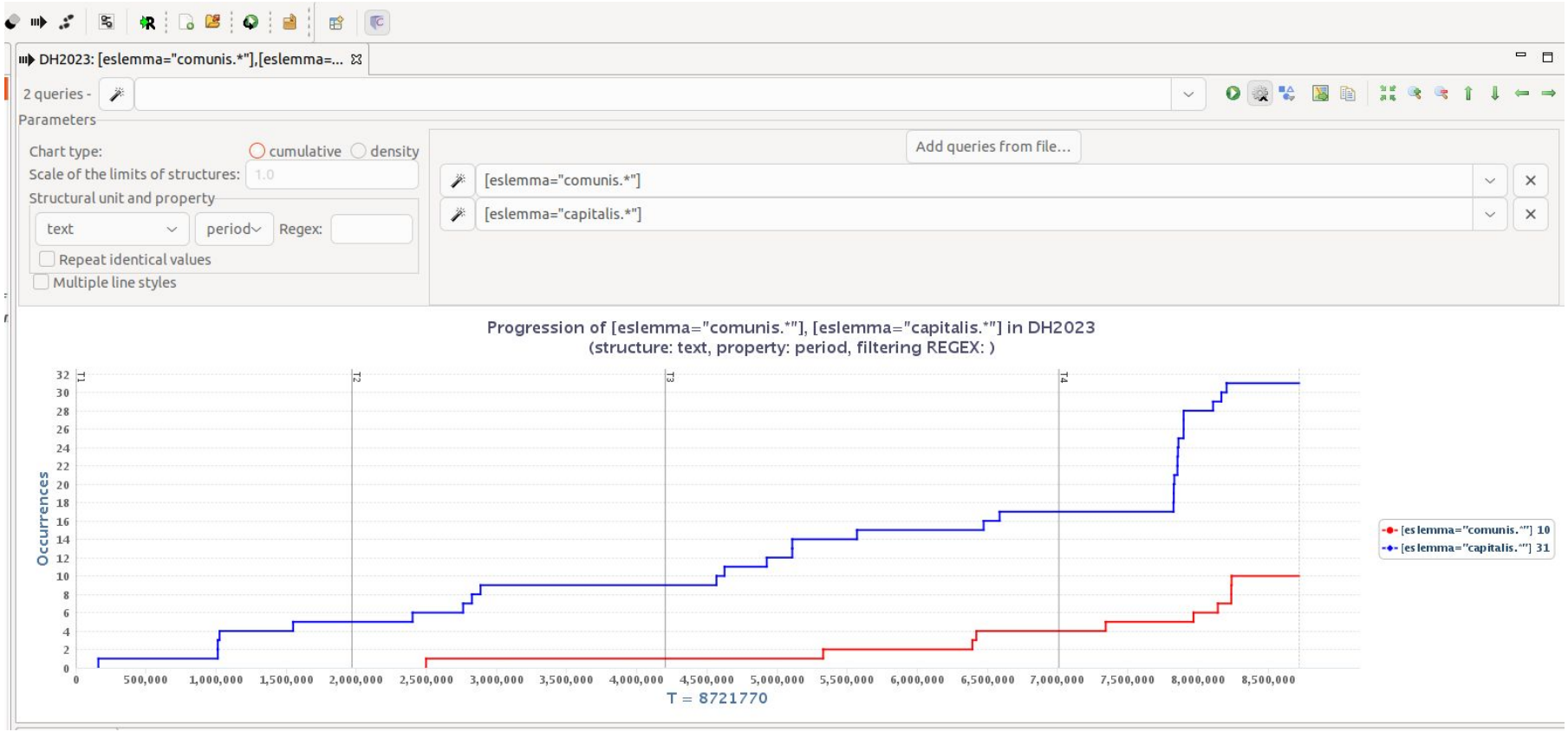
CQL syntax

- Combine different criteria with `&` (and) and `|` (or). E.g.:
  - `[lemma="power" & pos="VERB"]`
  - `[lemma="refill"] [] [lemma="kettle|teapot"]`
- Negate with `!` (not):
  - `[lemma="power" & pos!="V.*"]` or `[lemma="power" & !pos="V.*"]`

Progression analysis

- Global visualisation of a development (corpus analysed as a continuum). It enables the quantitative evolution of a word/collocation, for instance, through.
- Results presented as a cumulative chart (need to interpret the slope of the curve)

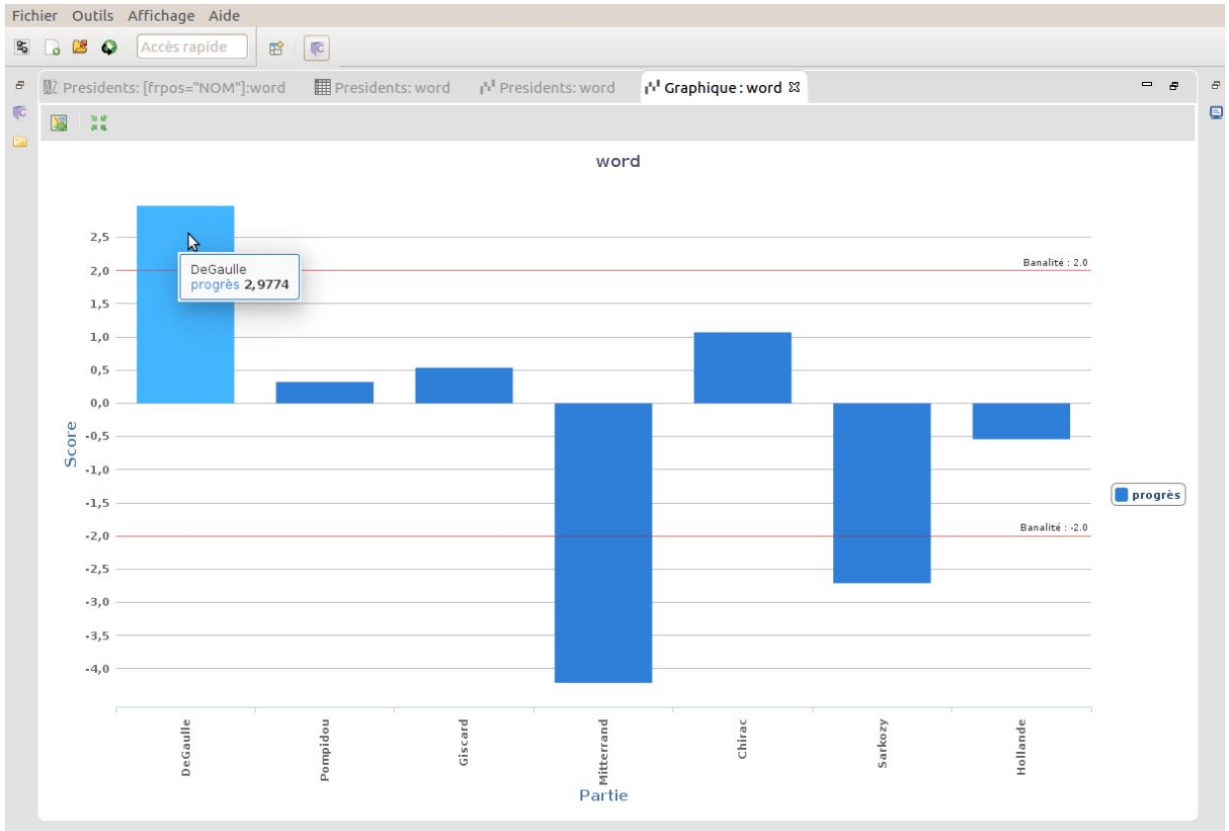
Progression analysis



Specificity index

- Based on the formula by Lafon (1984):
  - Lafon, P. (1984). *Dépouillements et statistiques en lexicométrie*. Genève, Paris: Slatkine.
- It can be used to show the specificity of a cooccurrence or to characterise a corpus partition

Specificity index



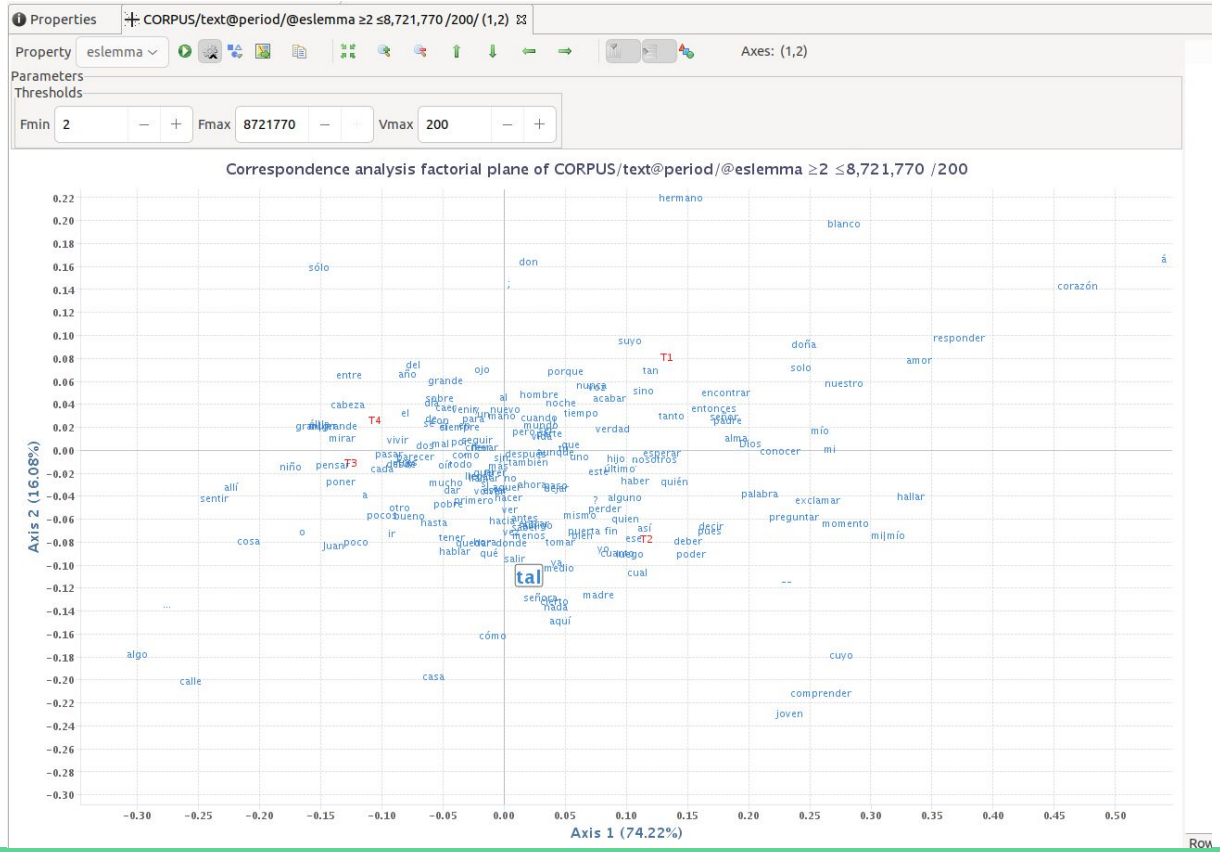
Task

1. Create a partition: one with the period T1 and another with the period T4
2. Calculate the specificities of PoS tags
3. Create a chart with the results

Factorial analyses

- Cartographic representation of the distribution of a specific property across partitions

Factorial analyses



To learn more

- Very fast introduction to TXM key concepts in 90 minutes: PDF
- Import of Europresse articles in XML/w+CSV to TXM
- List of manuals and resources in the TXM portal