TXM-Tutorial

Erasmus+ Lectures

Kraków, March/April 2016

Christof Schöch
CLiGS - University of Würzburg

GEFÖRDERT VOM
Bundesministerium
für Bildung
und Forschung

Overview

1. Individual functions
2. Several search modes
3. Advanced search
4. Wildcards and Symbols
5. Structuring a Corpus
6. Comparative Queries
7. Importing texts

What is TXM?

- A text analysis tool developed at the ENS Lyon
- Designed for linguistic or literary text analysis
- Indebted to and designed for "textométrie"
- Based on Unicode and XML, supports TEI
- Supports linguistic annotation and metadata
- Using the CQP corpus query processor (from Open Corpus Workbench)
- Flexible, free, open and in active development

Sample text collections

- 12 British novels by Arthur Conan Doyle
- Written in two different decades
- Belonging to four different subgenres: detective, adventure, horror, historical novel

1. Individual functions

Lexicon

List of all terms (which can be word forms, lemmas, POS, etc.) which appear in the corpus.

- Can be sorted alphabetically or by frequency
- Useful to find most frequent and very rare words

Index

Function to find out the frequency of a certain term in the corpus

- Gives a list of found terms and their frequency
- It is possible to set the property of the term that should be shown (word, lemma, POS or combination)
- Important conceptual distinction: property of the query vs. property displayed
- Further tools can be started from here

Concordance

Search which displays the context of a certain term

- This function shows the hits in context, with some words appearing before and some words following
- Includes a reference to the document in which each hit was found
- Again, the property of the hits to be shown can be customized

Cooccurrences

Search for terms which typically appear together with a target term (many options!)

- `cooccurrent`: the term cooccurring with the target term
- `frequency`: absolute frequency of the cooccurring terms;
- `cofrequency`: absolute frequency with which target term and found term occur together
- `score`: measure for the degree of specificity of a cooccurrence (most relevant)
- `mean distance`: mean distance between the target terms and the found term

2. Several search modes

(Depending on the annotations present)

Word forms

Search for graphical word forms

- `"worked"` - the simplest mode; finds the number of times "worked" occurs
- `[word="worked"]` - same, but this query can be enhanced more easily (see below)

Lemmas

Search for all terms which correspond to a specific lemma (base form)

- `[enlemma="mouse"]` – finds all terms corresponding to the lemma "mouse" in its different forms: "mouse", "mice"
- The label used is language dependent: `enlemma`, `pllemma`, `frlemma`, `eslemma`, etc.

Part-of-Speech

Search for all terms which belong to a certain grammatical category.

- `[enpos="VV"]` - finds any verb: "was", "is", "had", etc.
- `[enpos="JJ"]` - finds any adjective: "other", "great", "little".
- The inventory of POS-Tags depends on the language and tagger used; the label is language-dependent: `frpos`, `plpos`, etc.

3. Advanced search queries

Sequence of several search termsss

One search term followed by another search term. (See also the Query Editor!)

- `[word="she"][word="did"]` - finds the number of hits for the sequence "she did". It is possible to combine several search modes
- `[enpos="JJ"][enlemma="house"]` - finds hits in which an adjective is followed by any word form based on the lemma "house": for example, "great house", "empty house" and "old house".

Search terms in a certain distance

Several search terms with a minimal (and maximal) distance between them:

- `[enpos="VV"][]{0,5}[enlemma="money"]` - finds all combinations of a verb, followed by 0 to 5 words of any kind, followed by the lemma "money": for example, "take his money" and "add to the money" and "do to raise the money".

Combination of criteria for one term

Combines several criteria of different types for one single term, for example criteria concerning the word for and the grammatical category

- `[word="dog" & enpos="VV"]` - finds all terms which are written "dog" AND are a verb (to dog = to follow someone). This notably excludes all cases in which "dog" is a noun

4. Wildcards and Symbols

Wildcard and quantifiers:

- . dot = wildcard: any word character (but not a whitespace)
- * asterisk = the character or term immediately preceding has to be present zero or several times
- + plus = ... has to be present at least once or several times
- ? question mark = ... has to be present either zero times or one time

Some more useful symbols

- `{x}` = numerical indicator. Indicates how many times the previous character or term should be present.
- `{x,y}` = numerical range. Indicates the minimum and maximum number the previous character or term should be present
- `%c` = ignore uppercase/lowercase distinction (add after the quotation marks)
- `%d` = ignore accents and diacritics (add after quotation marks). The combination `%cd` is also possible.

Examples:

- `[word="heaven"%c]` – finds: "heaven" (physical) und "Heaven" (metaphysical / or at sentence-initial position).
- `[word="the"][enpos="JJ"]{2}[enpos="NN"]` - wildcard on the level of terms; sequence of "the", then two adjectives, then one noun. Finds "the deep blue sky" and "the angry old man".

Search with alternatives

Find several search terms at a time (using the "|" pipe character)

- `[word="city|village"]` - alternatives concerning word forms; finds all passages containing either "city" or "village"
- `[enpos="DT|PP"]` - alternatives concerning the part-of-speech; finds all articles and personal pronouns: "the", "a", "it", "you", etc.

5. Structuring a corpus

Define a sub-corpus

Only a specific part of the corpus goes into the subcorpus

- Right-click on "DOYLE", choose `subcorpus`
- Give the subcorpus a name, for instance "historical"
- Stay in the `simple` tab and adapt the settings: structure="text", property="subgenre". Confirm.
- From the list of subgenres, select "historical"; done!
- (For more complex scenarios, use the assisted and advanced modes)

Partition a corpus

Add an internal structure to the corpus based on metadata

- Right-click on `DOYLE`, choose `partition`
- Give the partition a name, e.g. "decades".
- Stay in the `simple` tab and adapt the settings: structure="text", property="decade". Confirm.
- The corpus will be automatically subdivided according to decade.
- (Right-click on your partition and select `Dimensions` to find out about the proportions in your partition.)

6. Comparative analyses

Progression

- In Index oder Lexicon, right-click on a term
- Choose `Send to progression`
- Graph type: select `cumulative` or `density`
- Bandwith multiplier: set to "0.5" (or vary between 0.2 und 1.0)
- Structural unit: select "text"
- Property: select "subgenre" or "title"; Confirm

Specificities

- Click on a partition, then right-click
- Select `specificities`
- Define a `property` (for example "word")
- `Part focus`: leave empty to compare all partitions; confirm.

7. Import texts

Various import formats

- Using the functions provided in `File>Import`
- Directly from the clipboard (single text)
- From a folder with text files (with optional `metadata.csv` file)
- From a folder with XML/TEI files (with optional `metadata.csv` file)

Import options

- `Main language`: for linguistic annotation; make sure to select the right language (requirement: TreeTagger and relevant language parameter file)
- `Metadata preview`: check if you have a metadata.csv file. One column needs to be called `id` with entries corresponding to the file names.

Conclusion

Conclusion

- Now you know the basic functions of TXM!
- Let's move on and try this out in practice

T
X
M
}Textométrie
Fédération des recherches et développements
en textométrie autour de la création
d'une plateforme logicielle ouverte

Christof Schöch, 2016

http://www.christof-schoech.de