CRETA
CENTER FOR REFLECTED TEXT ANALYTICS

Handout

„From Text to Networks“ – Tutorial @ DH 2018, Montreal, August 8, 2017

Agenda

09.30 – 09.45	Lecture: Introduction
09.45 – 10.45	Hands-on: Parallel annotation and inter-annotator agreements
10.45 – 11.00	Discussion: Observations made during hands-on session
	Coffee Break
11.30 – 12.00	Lecture: Segment annotation
12.00 – 12.45	Hands-on: Segmentation and network visualization
12.45 – 13.00	Concluding discussion

Available Annotation Tools

WebAnno	https://webanno.github.io/webanno/
Slate	https://bitbucket.org/dainkaplan/slate/
SWAN	https://github.com/annefried/swan
CorA	https://github.com/comphist/cora
CATMA	http://catma.de (there is also a tutorial next door!)

Regular Expressions Cheat Sheet

Basics

Most characters match themselves: `/a/` matches the character “a”.
Sequences of characters match sequences of the characters: `/the/` matches “the”.

Special characters

The dot matches everything: `/./` matches every single character
In order to match a (real) dot, we need to escape it: `/1\./` to find “1.”
All symbols with special meaning (see below) can be escaped to find them literally.

“Quantifiers”

The **question mar**k makes the previous thing optional: `/them?/` matches both “the” and “them”
The **plus sign** matches the previous thing multiple time (but at least once: 1-n times): `/ab+/` finds “ab”, “abb”, “abbb”, …
The **Kleene**[1] **star** matches the previous thing zero or more times: `/ab+/` matches “a”, “ab”, “abb”, …

Alternations

Alternations are specified with `(…|…)` and allow specification of optional variants.
`/(good|better|best)/` matches all three adjective forms.
`/great(er|est)?/` matches all three adjective forms of “great”

Character classes

Square brackets are used to create classes of characters:
`/[aeiou]/` matches all vowels
`/[Tt]he/` matches upper and lower case forms of “the”
`/[a-z]/` matches all lowercase characters, `/[0-9]/` all digits

[1] Named after the mathematician Stephen Cole Kleene, 1909-1994.