Building and Exploring (Web) Corpora [2]

EMLS 2008, Stuttgart

23-25 July 2008

Pavel Rychlý

pary@fi.muni.cz

NLPlab, Masaryk University, Brno

Outline

- Regular expressions, CQL, CorpusBuilder
  - Regular expressions tutorial
  - Corpus Query Language
    - defining word sketches
  - CorpusBuilder
    - building corpora from your own texts
- Exercises 2
  - RE examples
  - Create own corpus using CB
  - Define simple sketch grammar

Regular expressions

- Wikipedia: In computing, a **regular expression** is a string that is used to describe or match a set of strings, according to certain syntax rules.
- regular expressions tutorials
  - http://gnosis.cx/publish/programming/regular_expressions.html
  - http://www.zvon.org/other/PerlTutorial/Output/index.html
- regular expression exercises
  - http://www.itri.brighton.ac.uk/ARCHIVE/courses/MScLex/exercises/re

CQL

- Corpus Query Language
- all queries created by filling-in the concordance search form can be expressed in CQL
  - the form is there just for convenience
  - conc_description link

CQL syntax by examples

- find all occurrences of the word *play*
  - [word="play"]
- all words which have *play* as lemma
  - [lemma="play"]
- lemma *play* as noun
  - [lemma="play" & tag="N.*"]

CQL syntax by examples (2)

- verb *fight* followed by
  - any preposition
    - [lemma="fight" & tag="V.*"] [tag="PR.*"]
  - preposition *for*
    - [lemma="fight" & tag="V.*"] [word="for" & tag="PR.*"]
- verb *fight* preceded by a noun
  - [tag="N.*"] [lemma="fight" & tag="V.*"]
- verb *fight* followed by the noun *independence* (window 5)
  - [lemma="fight" & tag="V.*"] []{0,4} [lemma="independence"]

CQL in depth

- **attribute expression** is enclosed in square brackets and matches single position in a corpus
  - [word="play"]
  - [word="play.*"]
  - [word="(foot|volley|basket)ball"]
  - [lemma="play" | lemma="drama"]
  - [(lemma="play" | lemma="drama") & tag="N.*"]

CQL in depth

- **CQL query** is a regular expression over tokens
  - [tag="AV0" | tag="AJ."]***** [tag="N.*"]**{1,}**
  - [word="`"] [word!"'"]**{0,10}** [word="'"]
- it is important to understand that RE can be used on two levels:
  - for matching strings in attribute expressions
    - [word="[Tt]hank.*"]
  - for matching token sequences
    - [lemma="look"] [tag="PR."]**?** [tag="N.*"]**{1,}**

CQL excercises

1. any verb in a past tense (it's tag is *VBD*, *VDD*, *VHD* or *VVD*)
2. *fast* as a noun or a verb
3. any noun followed by a verb
4. any verb followed by two or more prepositions
5. any noun phrase (may be simplified)
6. *like* <noun phrase> *so*/*very much*

CQL readings

- http://trac.sketchengine.co.uk/wiki/SkE/CorpusQ
- http://www.fi.muni.cz/~thomas/corpora/CQL/

Sketch Grammar

- definition of word sketch grammatical relations
- set of queries with keyword and collocation labels
  - **1:** keyword
  - **2:** collocation
- special options for symmetric and dual relations

Corpus Builder

- web interface for getting your own texts into the Sketch Engine
- main features
  - associating uploaded texts with metadata
  - POS-tagging of uploaded texts (currently only for English, TreeTagger)
  - build word sketches and thesaurus
- demo

Create your own corpus

- from template
  - English, TreeTagger+WS
- if you don't have any texts handy download some web pages in plain text format, e.g. Wikipedia pages
- upload your texts
- POS-tag and lemmatize
- merge
- encodevert

Create your own word sketches

- create a sketch grammar
  - you can copy an existing grammar from a same language corpus
- compile word sketches
  - upload the grammar
- recompute scores in ws
- test word sketches
- (compile thesaurus)

Summary: What have you learned?

- Regular Expressions
- Using corpus manager Sketch Engine
  - searching for concordances using CQL
- Using Corpus Builder
  - build a tagged corpus from your own texts
  - define own grammatical relations