Voces

R-based Dashboard for Lexical Semantics

Krzysztof Nowak

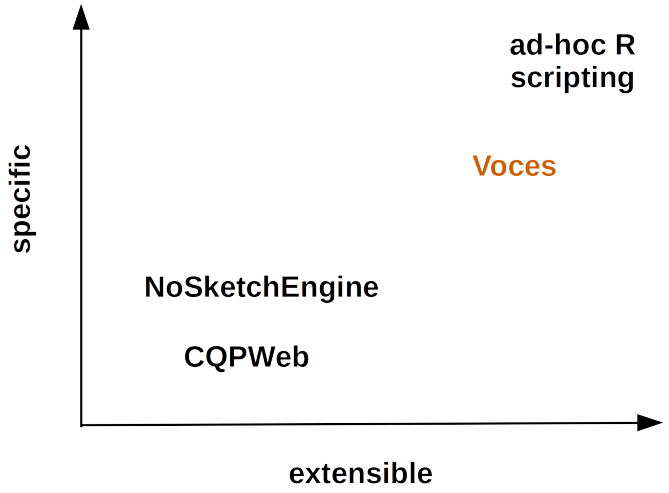
Lexicon Mediae et Infimae Latinitatis Polonorum
Institute of Polish Language
Polish Academy of Sciences

8/10/2017

Why?

Motivation

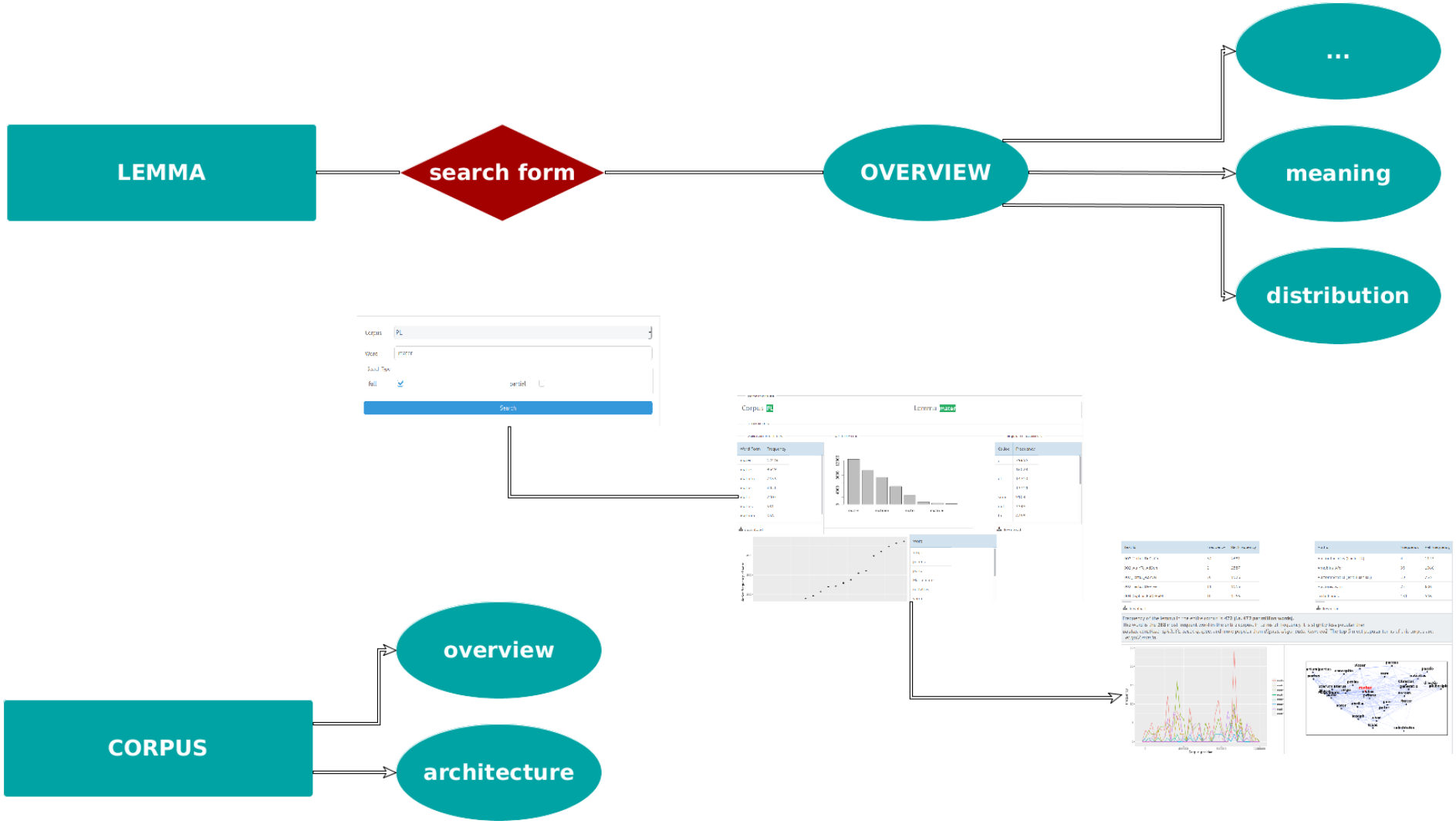
- domain: lexical studies (use case: Medieval Latin lexicology)
- goals
  - quick inspection of word's properties
  - integration of R code for flexible presentation
  - searching for best model
- target: between fully fledged corpus query systems and *ad-hoc* scripts



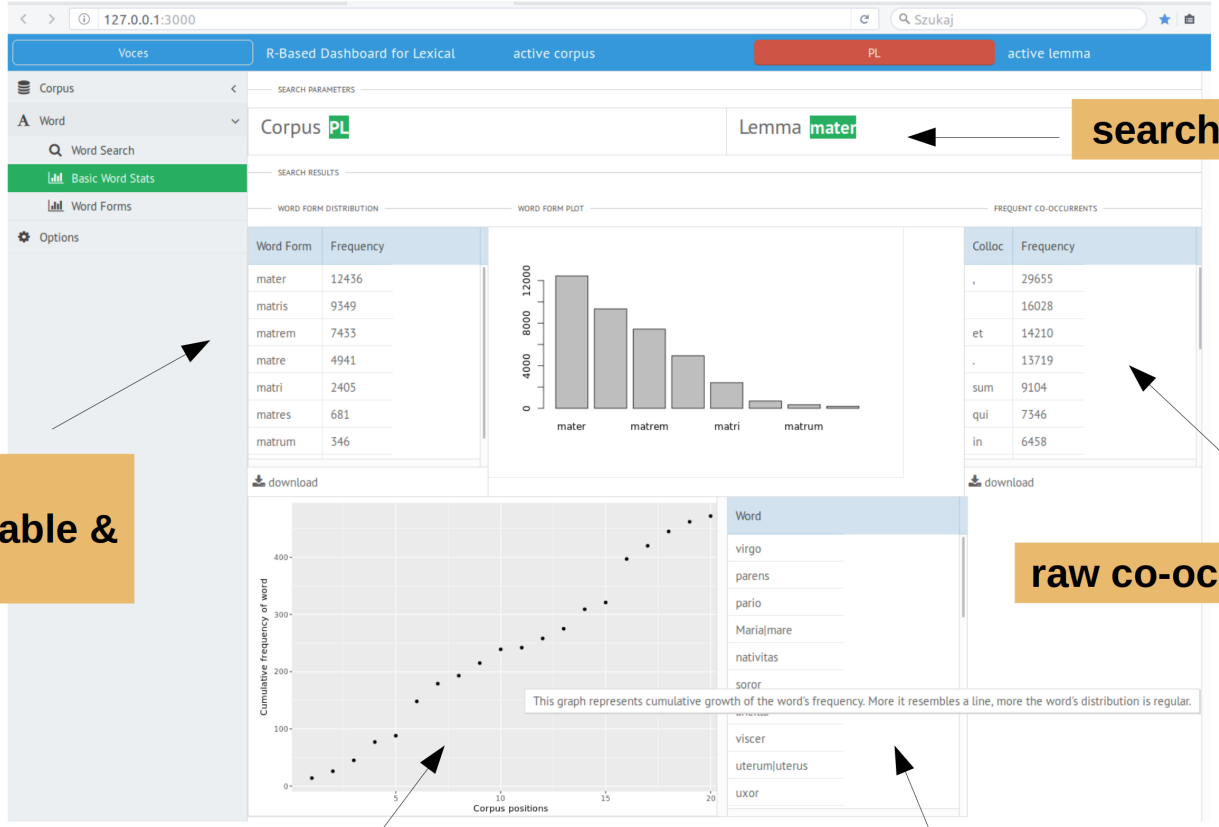
Dashboards (Few 2013)

- information at a glance
- visual display
- specific objectives
- fits on a single screen

Overview


LEMMA
search form
OVERVIEW
...
meaning
distribution
CORPUS
overview
architecture

General Lemma View



sidebar

wordform distribution table & barplot

search params

raw co-occurrences

cumulative frequency

similar terms

In-depth Lemma View

word distribution by authors and texts

basic prose description of the word distribution

wordform frequency by corpus position

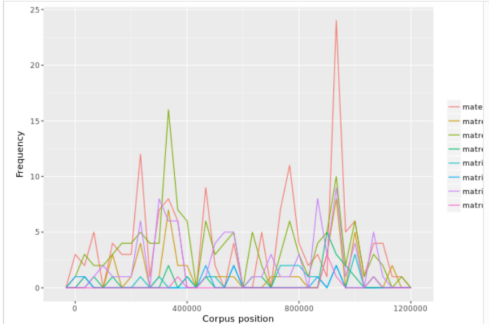
Text Id	Frequency	Rel
002_Tertul_DeCaCh	34	2681
002_AulnTe_AdSen	2	2587
002_Tertul_AdvVal	20	1926
002_Tertul_DeViVe	11	1513
004_CypCar_EpDeExM	10	1266

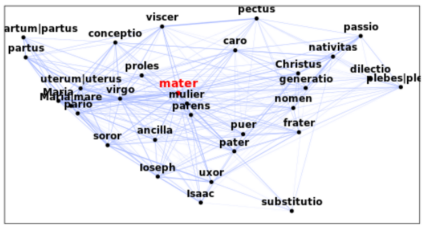
download

Author	Frequency	Rel
Auctor incertus (Lucius I?)	4	1153
Arnobius Afer	93	1066
Auctor incertus (Tertullianus?)	10	753
Auctores varii	23	606
Tertullianus	181	546

download

Frequency of the lemma in the entire corpus is **472 (i.e. 472 per million words).**
The word is the **288** most frequent word in the entire corpus. In terms of frequency it is slightly less popular than *paulus, constituo, spiritalis, sacer, quippe*, and more popular than *dignus, uirgo, puto, item, eo2*. The top 5 most popular terms of this corpus are: *.,et,qui2,sum,in.*





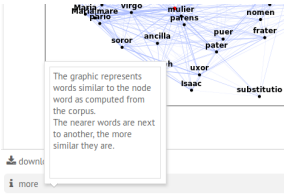
word similarity chart (pkg: wordspace)

understand

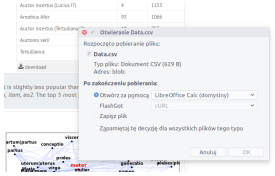
Author	Frequency	Rel Frequency
Auctor incertus (Lucius I?)	4	1153
Arnobius Afer	93	1066
		753
Auctores varii	23	606
Tertullianus	181	546

Relative frequency represents the average number of occurrences (per million words)

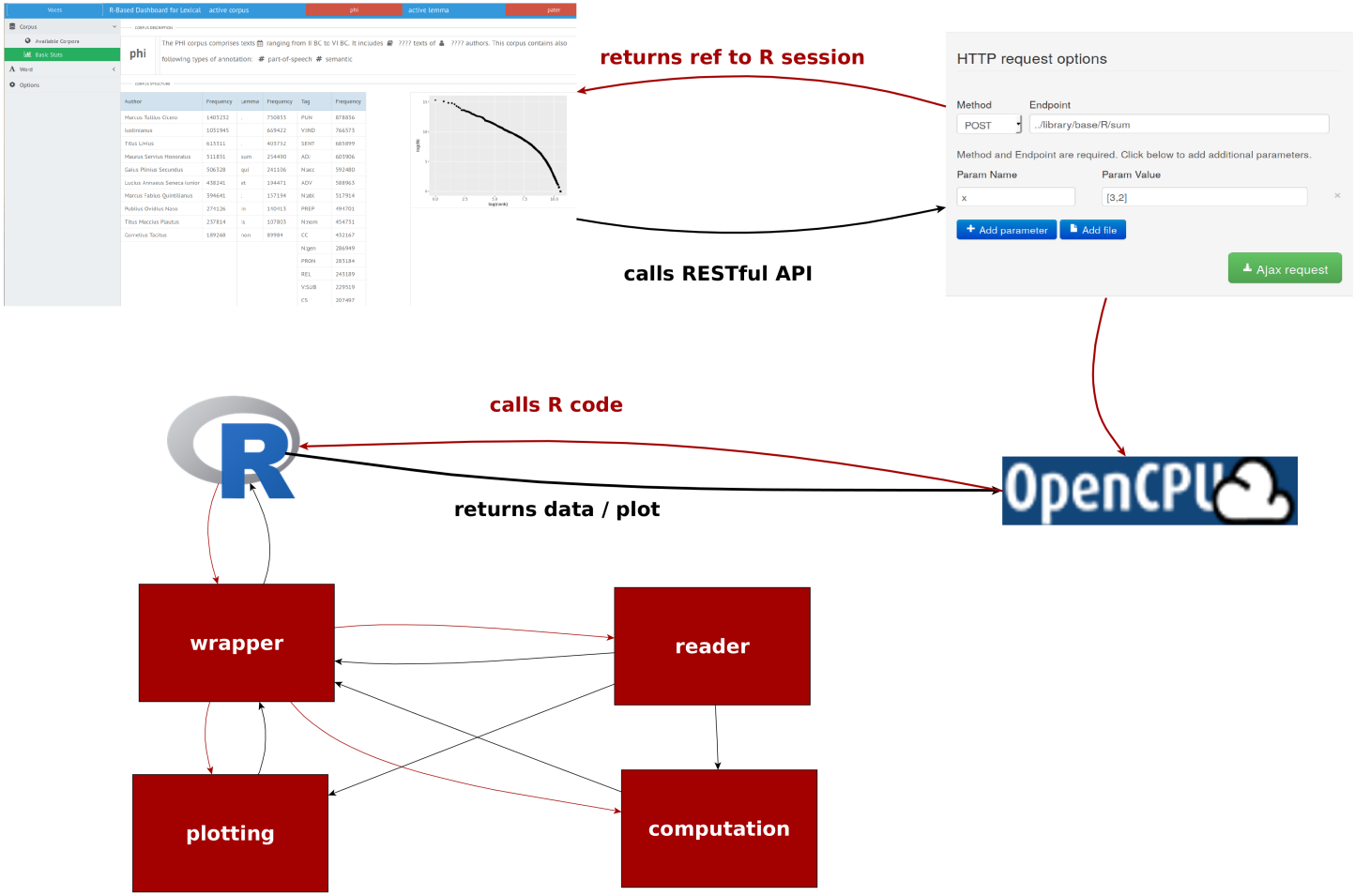
interpret



reuse



How does it work: architecture



How does it work: R code

wrap

- process AJAX calls
- dispatch call to a relevant function
- return results back

import

- read output of CQS (implemented: NoSKetchEngine, to do: CWB)
- prepare R objects for further processing

prepare data and plots

- possibly simple
- operate mostly on `text.tables`

Typical scenario

1. Ajax call to OCPU
2. OCPU calls wrapper fn
3. wrapper fn dispatches call to:
   - data fn or
   - plotting fn
4. data / plotting fn operates on read-in `data.table` or calls import fn
5. data / plot are sent back to OCPU
6. OCPU serializes output as JSON string

Perspectives

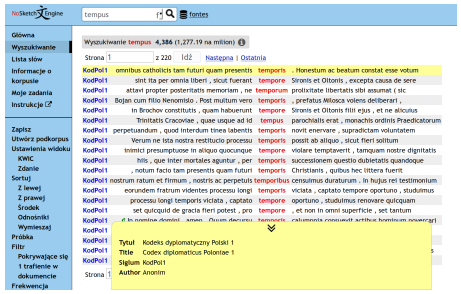
- analytic interface for the Corpus of Polish Medieval Latin
  - handling user input
  - storing search routines
- going beyond
  - two- and more lemma comparison
  - diachronic analysis
- optimization
- towards model of development

Questions?

krzysztof.nowak@ijp.pan.pl
Department of Medieval Latin
Institute of Polish Language, Kraków
(Polish Academy of Sciences)

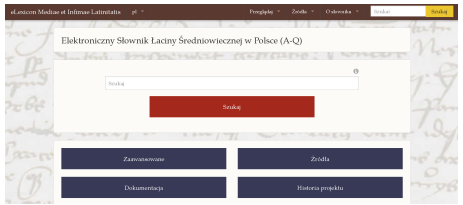
Our projects

eCorpus of ML



scriptores.pl/en/efontes

eDictionary of ML



scriptores.pl/elexicon