Turning Research Code into a Webservice with CLAM

Matteo Romanello (@mr56k)

Deutsches Archäologisches Institut / DARIAH-DE

Summary

CLAM is a Python tool developed in the context of CLARIN-NL which aims to simplify the process of transforming code into webservices.
In this demo I show a concrete example of applying CLAM to the code I've written for my doctoral research project.

Forschungskontext

- PhD in Digital Humanities Research
- King's College London (2009-2015)
- Supervisors: Willard McCarty and Shalom Lappin

Goal

- automatic extraction of canonical references
- with applications to: semantic indexing, information retrieval, network analysis

The Data: *L'Année Philologique*

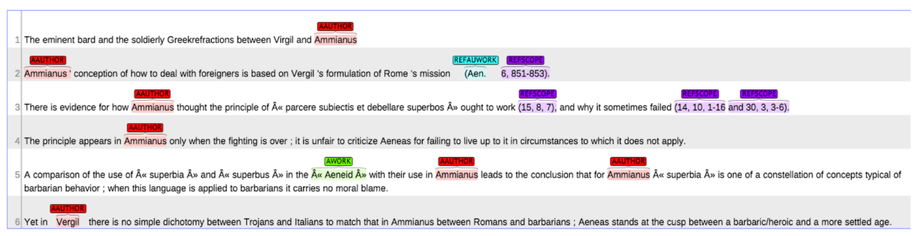
- bibliographic resource
- indexes every year anything published in Classics
- 85 volumes: 1 (1924) to 85 (2014)
- multilingual: FR, IT, DE, EN, ES

The Extraction Pipeline

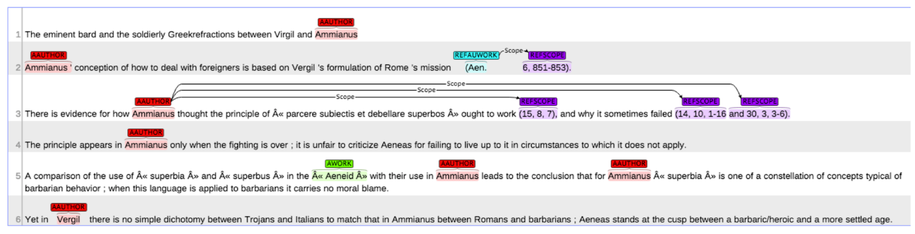
Pre-processing

1 The eminent bard and the soldierly Greekrefractions between Virgil and Ammianus
2 Ammianus ' conception of how to deal with foreigners is based on Vergil 's formulation of Rome 's mission (Aen. 6, 851-853).
3 There is evidence for how Ammianus thought the principle of Â« parcere subiectis et debellare superbos Â» ought to work (15, 8, 7), and why it sometimes failed (14, 10, 1-16 and 30, 3, 3-6).
4 The principle appears in Ammianus only when the fighting is over ; it is unfair to criticize Aeneas for failing to live up to it in circumstances to which it does not apply.
5 A comparison of the use of Â« superbia Â» and Â« superbus Â» in the Â« Aeneid Â» with their use in Ammianus leads to the conclusion that for Ammianus Â« superbia Â» is one of a constellation of concepts typical of barbarian behavior ; when this language is applied to barbarians it carries no moral blame.
6 Yet in Vergil there is no simple dichotomy between Trojans and Italians to match that in Ammianus between Romans and barbarians ; Aeneas stands at the cusp between a barbaric/heroic and a more settled age.

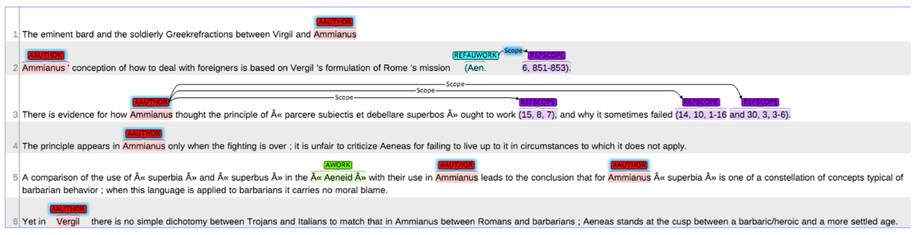
Named Entity Recognition



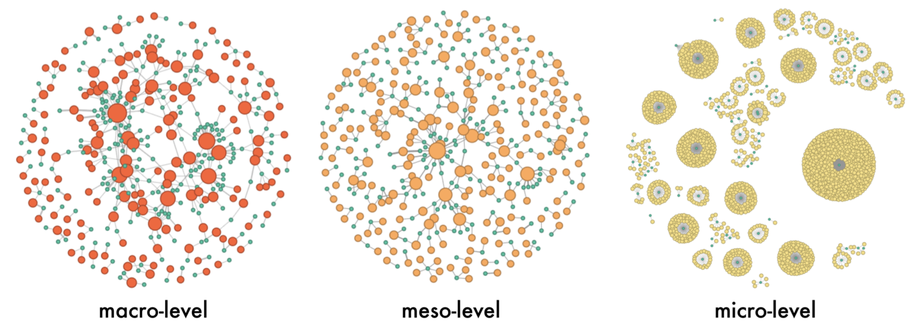
Relation Extraction



Named Entity Disambiguation



Outputs



Evaluation

Table 1: Evaluation results for the named entity extraction.

Algorithm	Precision	Recall	F1 Score
CRF	**79.24%**	69.62%	**73.88%**
MaxEnt	75.29 %	66.75%	70.43%
SVM	74.44%	**70.21%**	71.93%

Table 2: Evaluation results for the relation detection.

True Pos	False Pos	False Neg	Precision	Recall	F1 Score
350	25	31	93.33%	91.87%	92.60%

Table 3: Evaluation results for the disambiguation of `aauthor` and `awork` entities and `scope` relations.

Matching Type	Precision	Recall	F1-Score
Exact	58.33%	62.88%	60.52%
Approximate (threshold=4)	**61.04%**	90.94%	**73.05%**
Approximate (threshold=7)	58.94%	**94.76%**	72.67%

Software/Code

1 Citation Extractor

Lorem ipsum Verg. *Aen.* 12.10 f. sit amet, consectetur adipiscing elit. Duis facilisis lacinia ipsum id pharetra.

Verg. *Aen.* 12.10 f.

Verg. *Aen* + 12.10 f.

2 Citation Matcher

Verg. Aen. — lookup against KB → urn:cts:latinLit:latinLit:phi0690.phi003

12.10 f. — through CitationParser → 12.10-12.11

urn:cts:latinLit:latinLit:phi0690.phi003:12.10-12.11

3 Knowledge Base

Authors → Name variants

Text Passages → Unique identifiers

Works → Title variants

(1,500 authors; 5,500 works)

4 Citation Parser

12.10 f. / 12,10 s. / XII 10 s. →

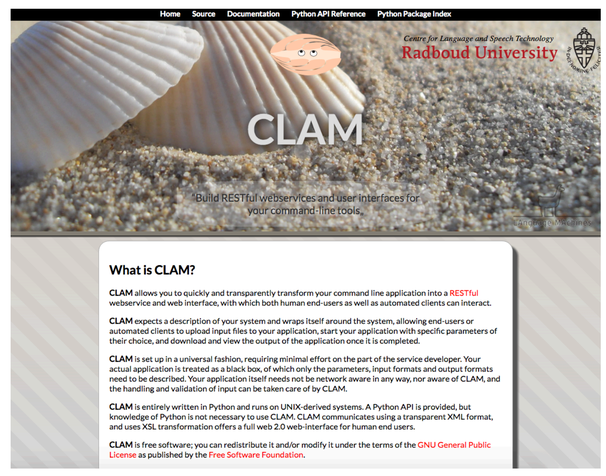
```
{
scope:
  {
   start: [12, 10]
  , end: [12, 11]
  }
}
```

(Context-Free Grammar)

The Computational Linguistics Application Mediator (CLAM)

About CLAM:

- `http://proycon.github.io/clam/`
- Open Source Python library
- written by Maarten van Gompel
- developed within CLARIN-NL
- rich documentation + video tutorials



CLAM: what for?

- CLAM simplifies the task of transforming existing code into webservices
- creates a wrapper for complex pieces of software as well as commandline scripts
- is based on the concept of input and output templates

What CLAM generates:

- a project skeleton (with plenty of inline documetation!)
- a GUI (basic + customisable with templates)
- a REST API
- service documentation at `http://<webserviceurl>/info/`

What CLAM needs:

- some settings in the configuration file
  - input template(s)
  - output template(s)
  - global parameters (optional)
- a wrapper that calls your code/script for each input file

...and now: Demo Time

Let me show you how the Canonical Reference Extractor Web-service works!

Contacts and Acknowledgements

- thanks Maarten for having created such an awesome library!
- thanks to Eric Rebillard and Adam Chandler for their help with annotatins the APh data