Code aus einem Forschungskontext mit CLAM in einen Web-Service verwandeln

Matteo Romanello (@mr56k)

Deutsches Archäologisches Institut / DARIAH-DE

Zusammenfassung

CLAM ist ein Python Tool, das darauf abzielt, die Transformation von Code aus einem Forschungskontext in einen Web-Service zu vereinfachen. Während der HandsOn Session wird die CLAM Anwendung am Beispiel des für meine Promotion entwickelten Programms vorgeführt.

Forschungskontext

- Promotion in Digital Humanities Research, am King's College London (2009-2015)
- Doktorväter: Willard McCarty und Shalom Lappin

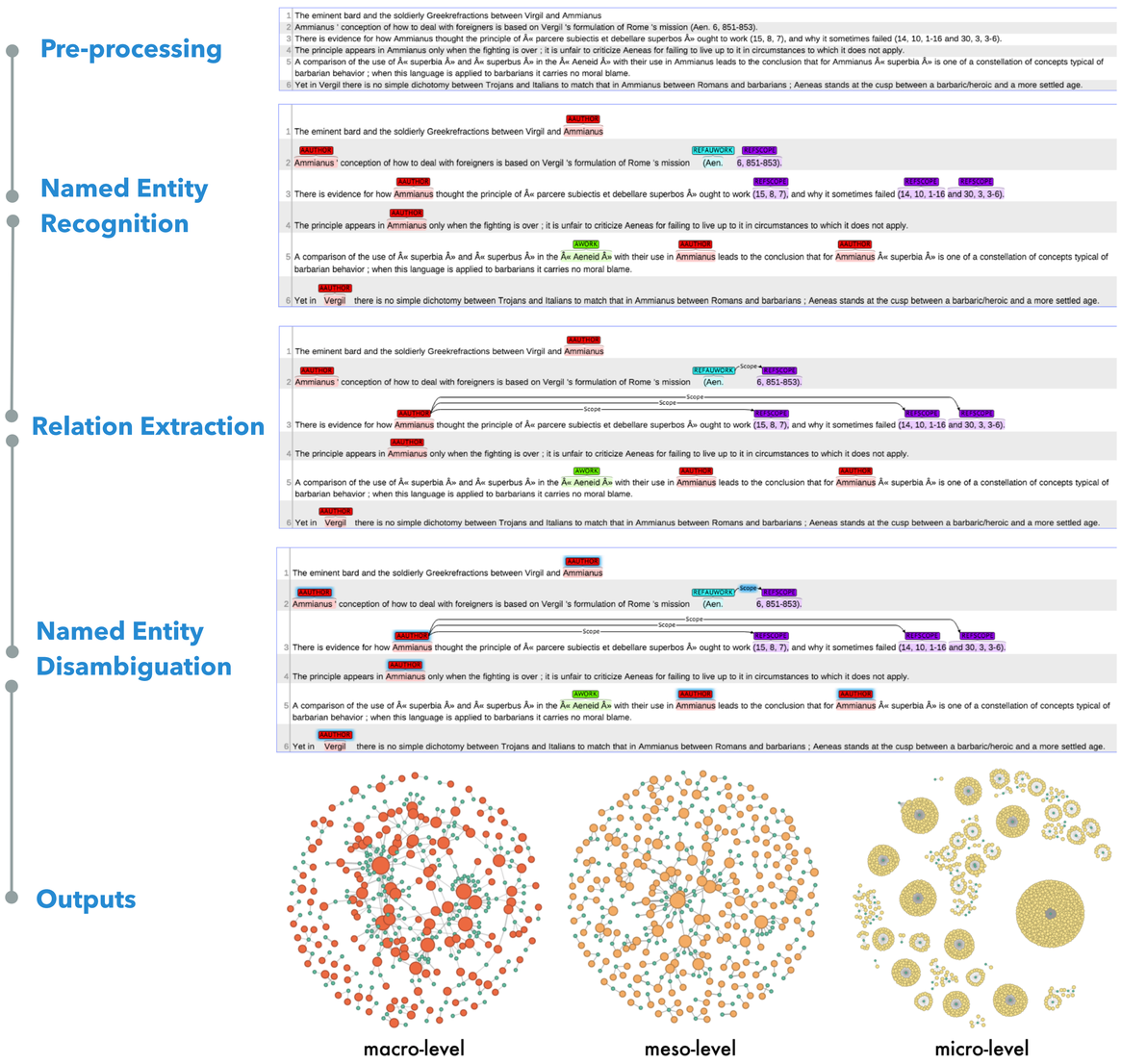
Ziel

- automatische Extraktion von Textreferenzen auf Primärquellen
- mit Anwendungen für: semantische Indexierung, Information Retrieval, Netzwerkanalyse

Die Daten: *L'Année Philologique*

- bibliographischer Katalog von Publikationen in den Altertumswissenschaften
- 85 Bände: 1. (1924) bis zum 85. (2014)
- mehrsprachig: FR, IT, DE, EN, ES

Die Extraktions-Pipeline



Auswertung

Table 1: Auswertung der Ergebnisse der 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%

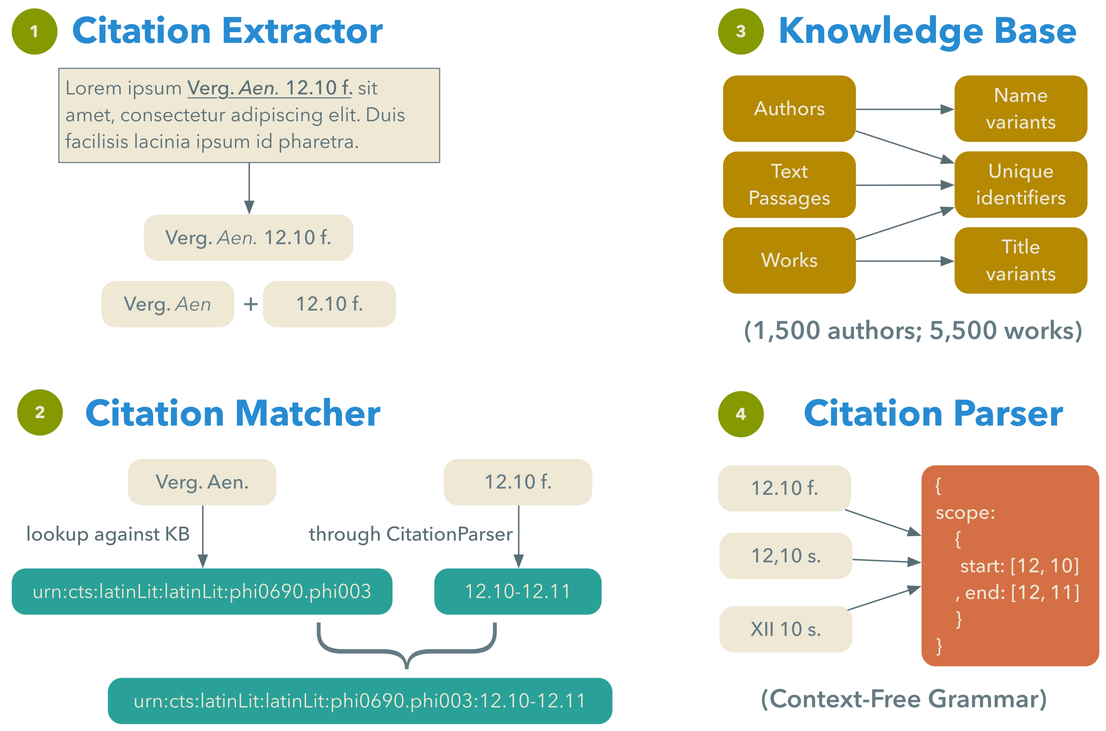
Table 2: Auswertung der Ergebnisse der Relation Detection.

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

Table 3: Auswertung der Ergebnisse der Disambiguierung von `aauthor` und `awork` Entitäten und `scope` Beziehungen.

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%

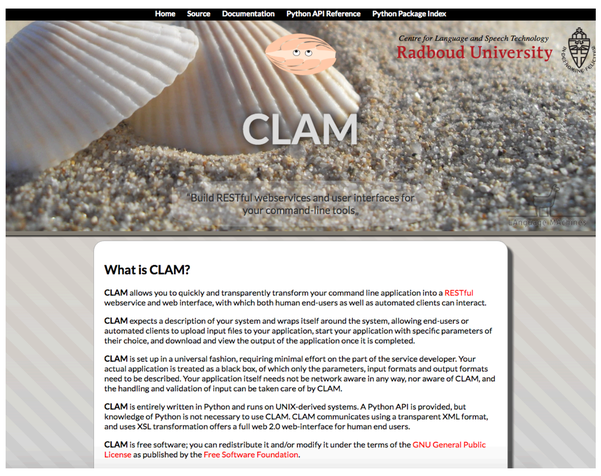
Software/Code



Der Computational Linguistics Application Mediator (CLAM)

About CLAM:

- `http://proycon.github.io/clam/`
- Open Source Bibliothek im Python
- im Rahmen von CLARIN-NL entwickelt von Maarten van Gompel
- reichhaltige Dokumentation + Video-Tutorials



CLAM: wofür?

- CLAM vereinfacht die Transformation von Code aus einem Forschungskontext in einen Web-Service
- erstellt einen Wrapper für komplexe Software-Bausteine sowie Befehlszeilen-Scripts
- basiert auf dem Konzept von Input- und Output-Templates

Was CLAM bietet:

- eine Projekt-Vorlage (mit reichhaltiger Dokumentation!)
- eine Benutzeroberfläche (grundlegend aber über Templates anpassbar)
- eine REST API
- Dokumentation des Web-Services unter `http://<webserviceurl>/info/`

Was CLAM braucht:

- Einstellungen in einer Konfigurationsdatei
  - Input Template(s)
  - Output Template(s)
  - globale Parameter (beliebig)
- ein Wrapper der Ihren Code für jede Input-Datei aufruft

...und jetzt: Demo Time

Lassen Sie mich Ihnen zeigen, wir der Canonical Reference Extractor Web-service funktioniert!

Danksagungen

- Dank an Maarten für eine solche wunderbare Python-Bibliothek!
- Dank an Eric Rebillard und Adam Chandler für die Hilfe bei der Annotation der APh-Daten