

# GitMA-Poster

## CATMA-Daten via Git abrufen und mittels Python-Bibliothek weiterverarbeiten

### Meister, Malte

meister@linglit.tu-darmstadt.de  
Technische Universität Darmstadt, Germany

### Vauth, Michael

vauth@linglit.tu-darmstadt.de  
Technische Universität Darmstadt, Germany

### Gerstorfer, Dominik

gerstorfer@linglit.tu-darmstadt.de  
Technische Universität Darmstadt, Germany

Etwas zu erinnern heißt nicht, es abzuspeichern, sondern auch, es abzurufen und weiter zu prozessieren. Denn nur im produktiven Anschluss erhält die Erinnerung eine Bedeutung. Diese Beobachtung trifft *a fortiori* auf technische Speichersysteme zu. Der Nutzen einer Software wird, gerade in den Digital Humanities, über die Möglichkeiten bestimmt, die erzeugten Daten zu exportieren, zu konvertieren, zu archivieren und in anderen Systemen weiterzuverarbeiten. In diesem Poster werden wir neue Möglichkeiten vorstellen, Daten aus CATMA (Gius et al. 2021) abzurufen und nachzunutzen.

CATMA (Computer Assisted Text Markup and Analysis) ist eine kollaborative Textannotations- und Analyse-Plattform, die in den Digital Humanities gut etabliert ist und von vielen Projekten aktiv genutzt wird. Annotationsexporte waren, besonders im XML-TEI Format, schon seit Version 3 ein wichtiger Bestandteil der CATMA-Software (Petris & Meister 2016; Petris 2017). Der Datenzugriff war aber bis einschließlich CATMA 5 nur über die graphische Benutzeroberfläche (GUI) möglich. Seit der Version 6.0 werden die von den Nutzer:innen erzeugten Daten in einem auf Git basierenden Backend gespeichert und versioniert.

Der genaue Aufbau der Datenstrukturen wird auf der CATMA Webseite dokumentiert (Petris 2020): Jedes Dokument, jede Annotation Collection einschließlich der Annotationen, sowie jedes Tagset einschließlich der zugehörigen Tags werden im Backend einzeln repräsentiert. Besonders wichtig für die Weiterverarbeitung der Annotationsdaten sind die Informationen, mit denen die einzelnen Annotationen repräsentiert werden:

- eine Referenz auf das entsprechende Dokument
- die genaue Platzierung der annotierten Textspanne (als sogenannte Start und End-Offsets, welche sich auf die Zeichen-Positionen im Dokument beziehen)
- eine Referenz auf das verwendete Tag (die Annotationskategorie) und das Tagset (eine benannte Kollektion von Tags) aus dem es stammt
- eventuell Properties (vordefinierte erweiternde Eigenschaften) und deren Werte
- Autor:in der Annotation
- Zeitpunkt der Annotation

Die Nutzer:innen können sowohl auf eigene als auch auf mit ihnen geteilte Daten in Form von Git Repositorien zugreifen. Diese stellen damit eine Art Programmierschnittstelle (API) zum Abruf von CATMA-Annotationen dar, welche auf den lokalen Rechner heruntergeladen oder in anderen Tools weiterverarbeitet werden können.

Im Fachbereich für Digital Philology an der TU Darmstadt ist außerdem eine Python-Bibliothek entstanden, die einen einfachen Zugriff auf die Git Repositorien zulässt. Sie ermöglicht die Weiterverarbeitung der Annotationen mit gängigen Python Datascience-Tools, zum Beispiel als Pandas DataFrame. Mit der Python-Bibliothek lassen sich unter anderem Berechnungen des Inter Annotator Agreement oder Visualisierungen zum Annotationsfortschritt und zur Annotationsexploration erstellen. Damit ermöglichen wir nicht nur die Annotationsauswertung, sondern auch die schnelle Identifizierung von Annotationsfehlern, die unmittelbar korrigiert werden können.

Insgesamt ist das zentrale Anliegen des Git Access, CATMA-Daten direkt verfügbar zu machen, damit Nutzer:innen nicht unbedingt an die schon in CATMA vorhandenen Funktionalitäten gebunden sind. Dadurch kann der Workflow zwischen Annotation, Annotationsauswertung und Annotationsüberarbeitung deutlich schneller werden. Das ist besonders für Nutzer:innen relevant, die sich – unter anderem im Rahmen von Forschungsprojekten – um die Organisation und Evaluierung von Annotationen kümmern.

Mit unserem Poster werden wir diesen Workflow detailliert darstellen. Das Poster soll also auch als eine Art Bedienungsanleitung für die Nutzung des CATMA Git Access fungieren und Best Practices zeigen. Dabei werden wir folgende Schritte abdecken:

1. Voraussetzungen für den Zugriff auf die CATMA GitLab API
2. Installation der CATMA Python Pakete (bzw. eines Docker Image, welches alle Erfordernisse abdeckt)
3. Clonen der Repositories
4. Zugriff auf die Daten mit Python
5. Beispiele für die Annotationsexploration und -auswertung

CATMA erscheint zum Beispiel im TAPoR Toolverzeichnis, sowie in „Digitale Werkzeuge zur textbasierten Annotation, Korpusanalyse und Netzwerkanalyse in den Geisteswissenschaften“ (Frey-Endres & Simon 2021).

## Bibliographie

**Frey-Endres, Marcel / Simon, Tobias** (2021): „Digitale Werkzeuge zur textbasierten Annotation, Korpusanalyse und Netzwerkanalyse in den Geisteswissenschaften“. In: *Digital Philology | Working Papers in Digital Philology* 02/2021. Darmstadt: TUPrints. URL: [https://tuprints.ulb.tu-darmstadt.de/17850/1/Digital\\_Philology\\_\\_Working\\_Papers\\_in\\_Digital\\_Philology\\_vol002.pdf](https://tuprints.ulb.tu-darmstadt.de/17850/1/Digital_Philology__Working_Papers_in_Digital_Philology_vol002.pdf) [letzter Zugriff 24. November 2021]

**Gius, Evelyn / Meister, Jan Christoph / Meister, Malte / Petris, Marco / Bruck, Christian / Jacke, Janina / Schumacher, Mareike / Gerstorfer, Dominik / Flüh, Marie / Horstmann, Jan** (2021): CATMA 6 (Version 6.3). Zenodo. DOI: 10.5281/zenodo.1470118. URL: <https://catma.de/> [letzter Zugriff 24. November 2021]

**Petris, Marco** (2017): „TEI Export Format“. In: *CATMA*. URL: <https://catma.de/documentation/tei-export-format/> [letzter Zugriff 6. Juli 2021].

**Petris, Marco** (2020): „Git Access“. In: *CATMA*. URL: <https://catma.de/documentation/git-access/> [letzter Zugriff 6. Juli 2021].

**Petris, Marco / Meister, Malte** (2016): „Technology and Versions“. In: *CATMA*. URL: <https://catma.de/documentation/technology-and-versions/> [letzter Zugriff 6. Juli 2021].