Nachlass Ludwig Wittgenstein: Softwaretechnologien und computerlinguistische Methoden der Software-Infrastruktur um die FinderApp WiTTFind

Hadersbeck, Maximilian
maximilian@cis.uni-muenchen.de
Ludwig-Maximilians Universität München

Babl, Florian
Florian.Babl@campus.lmu.de
Ludwig-Maximilians Universität München

Eisterhues, Marcel
Marcel.Eisterhues@campus.lmu.de
Ludwig-Maximilians Universität München

Röhrer, Ines
I.Roehrer@campus.lmu.de
Ludwig-Maximilians Universität München

Still, Sebastian
Sebastian.Still@campus.lmu.de
Ludwig-Maximilians Universität München

Ullrich, Sabine
sabine.ullrich@campus.lmu.de
Ludwig-Maximilians Universität München

Landes, Florian
florian.landes@kbl.badw.de
Bayerische Akademie der Wissenschaften, München

Lindinger, Matthias
matthias.lindinger@bsb-muenchen.de
Bayerische Staatsbilbliothek, München

Die Infrastuktur und das Projekt

Seit 2010 kooperieren das Wittgenstein Archiv der Universität Bergen und das Centrum für Informations- und Sprachverarbeitung der Ludwig-Maximilians Universität München in der Forschungsgruppe „Wittgenstein Advanced Search Group“ (WAST). Die Forschungsgruppe entwickelt Web-Frontends (FinderApps) und spezielle Suchwerkzeuge, die sich gut für die Forschung und Lehre im Bereich der Digital Humanities eignen. Ihre erste Suchmaschine, die FinderApp WiTTFind (wittfind.cis.lmu.de, siehe Abb. 1), die den von der UNESCO zum Weltkulturerbe (im Jahr 2017) erhobenen (Schmidt 2018) Nachlass von Ludwig Wittgenstein durchsucht, gewann im Jahre 2014 der EU-Open-Humanity Award. Der Preis zeichnet Gruppen aus, die herausragende Technologie im Bereich der Humanities entwickelt haben. Die in der Forschergruppe programmierte FinderApp WiTTFind erlaubt es, mit hochqualifizierten, computerlinguistisch orientierten Suchwerkzeugen Nachlasstrans-kriptionen zu durchsuchen. Die Transkriptionen entstammen der *Bergen Normalized Edition*, die die Grundlage der Wittgenstein Edition bildet. Neben den gefundenen Treffern der Suchmaschine, werden in den Suchergebnissen von WiTTFind die Faksimile-Extrakte aus den Originaldokumenten angezeigt. So kann der Nutzer die „Aura“ der gefundenen Textstelle im Original studieren und nicht nur den transkribierten Text sehen.

Abbildung 1: WiTTFind (http://wittfind.cis.lmu.de)

Damit derNutzer auch den seitenweisen Kontext des Suchtreffers im Original studieren kann, wurde am CIS eine weitere WEB-Applikation entwickelt, der doppelseitige Reader. Dieser Reader ermöglicht es, vom Suchtreffer direkt an die entsprechende Stelle im entsprechenden Dokument des Originals zu springen. Im doppelseitigen Lesemodus kann der Nutzer in den Faksimile des originalen Dokuments blättern. Eine symmetrische Autovervollständigung gibt während der Suchanfrage einen statistischen und lexikalischen Zugang zu den Wörtern, die in der Edition vorkommen. Im Zentrum der Suche steht die selbstprogrammierte C++ Suchmaschine wf, die mit Hilfe von Vollformlexika (WiTTlex), verbessertem POS-Tagging und weiteren Metainformationen regelbasiertes Suchen erlaubt. Zum Aufspüren semantisch ähnlicher Textpassagen in der Edition gibt es das NLP-Tool WiTTSim.

Die thematisch getrennten Aufgaben innerhalb der Infrastruktur der WAST-Tools (siehe Abb. 2) werden

über REST-API's von einzelnen Microservices realisiert, deren zentrale Datenhaltung über eine mongo Datenbank realisiert wird. Die Oberflächen der FinderApps werden mit HTML5, Javascript und Bootstraptechniken für WEB-Browser programmiert und möglichst browserunabhängig gehalten.

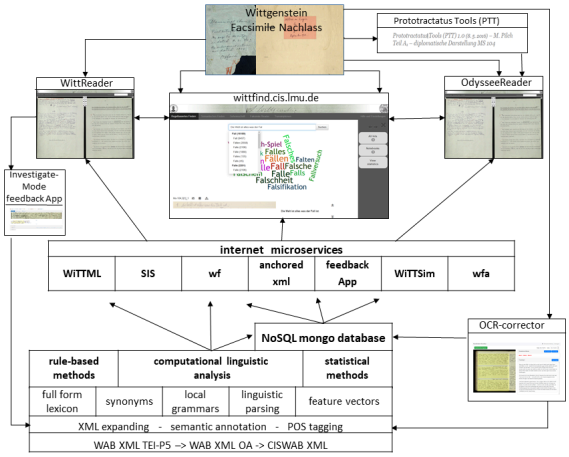

Abbildung 2: Infrastruktur der WAST-Tools (http://gitlab.cis.lmu.de)

Alle Programme, Schnittstellen und Entwicklungen werden dokumentiert (siehe Abb. 3) und Tutorials für Anschlussprojekte entwickelt. So ist gewährleistet, dass die Tools und Suchmaschinen nachhaltig verwendet und auch für die Forschung und Lehre eingesetzt werden können. Als Versionskontrollsystem wird git verwendet.

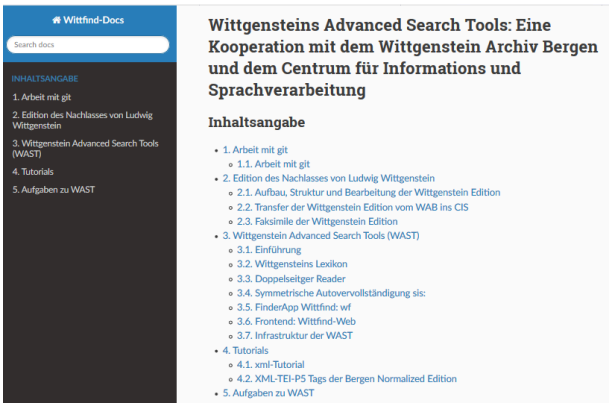

Abbildung 3: Dokumentation der WAST-Tools: http://wittfind.cis.uni-muenchen.de/wast/infrastruktur/index.html

Bei der Entwicklung der Infrastruktur der WAST-Tools wurden die strengen Vorgaben des EU-Open-Humanity Awards eingehalten: Forderungen nach Open-Source, interdisziplinäre Öffnung und Nachhaltigkeit. Diese Offenheit ermöglichte es weitere FinderApps für andere Wissenschaftsbereiche zu implementieren: GoetheFind (Faust-I und Faust-II Edition, Deutsches Textarchiv Berlin (XML-TEIP5, DTA Basis Format)), HistoFind (Briefwechsel Erzherzog Leopold Wilhelms an Kaiser Ferdinand III. aus dem Reichsarchiv Stockholm; Kooperation mit Historikern) und den OdysseeReader (Schreibprozess der zur Logisch-Philosophischen-Abhandlung führte; Kooperation mit Philosophen).

In diesem Workshop werden die verwendeten Softwaretechnologien und computerlinguistischen Methoden im konkreten Einsatz vorgestellt. Den Teilnehmer*innen wird ein Debian-10 Container mit allen notwendigen Programmen, Tools und Dokumentation der gesamten Softwareinfrastruktur zur Verfügung gestellt. Innerhalb dieses Containers können die Teilnehmer*innen die einzelnen Tools der WAST-Projektgruppe kennenlernen und bekommen von den Projektmitarbeiter*innen kleine Aufgaben gestellt, die sie dann mit ihnen bearbeiten. So können sie die Arbeitsweise der WAST Infrastruktur konkret kennenlernen.

Im Workshop werden folgende Datenformate, Tools und Programmierkonzepte vorgestellt und geübt

Gitlab Projektmanagement und Continuous Integration, XML TEI-P5 Edition CISWAB, Faksimilestrukturierung und Texterkennung, lexikalische Arbeit, WEB-Oberfläche der FinderApps und Einsatz mit Micorservices, doppelseitiger Faksimilereader mit MongoDB, NLP-Tools zur semantischen Ähnlichkeitssuche, Vorstellung und Programmierung einer regelbasierten Suchmaschine und die Erstellung eines Dokumentationssystems mit Sphinx.

Voraussetzungen an die Kursteilnehmer*innen

Programmierkenntnisse (Grundkenntnisse): LINUX (Arbeit mit der UNIX-Shell), Python, XML, HTML, git, javascript, POS-Tagging.

Da beim Workshop einige Entwickler der WAST-Tools anwesend sein werden, gibt es die Möglichkeit auch vertieft in die jeweilige Thematik einzusteigen.

Gitlab Projektmanagement und Continuous Integration (Hadersbeck, Still)

Im gesamten Projekt wird als Versionierungssystem git verwendet. Die Projektrepositories werden auf zwei unterschiedlichen Rechnern ausgerollt: Dem preview-Server für Tests und einem Projektserver für die offizielle Onlineversion. Es wird das in der Praxis

bewährte „git branching model“ kombiniert mit einer „continuous integration“ Technik eingesetzt. Mit einer Feedbackapp können Nutzer Fehler melden oder Implementierungswünsche äußern, die in Issues innerhalb der Projektrepositories bearbeitet werden.

XML TEI-P5 Edition CISWAB (Hadersbeck)

Als Datenbasis für das WiTTFind Projekt wird die „Bergen Nachlass Edition“ (BNE) verwendet, die sich an den Richtlinien der Text Encoding Initiative (TEI-P5) orientiert. Im Workshop werden die wichtigen TEI-XML-Elemente der BNE vorgestellt.

Faksimilestrukturierung und Erkennung (Eisterhues, Landes)

Da in den FinderApps neben den gefunden Textstellen auch die zugehörigen Faksimileextrakte aus der Edition dargestellt werden, sind Kenntnisse der Bildkoordinaten der Textstellen nötig. Diese Koordinaten werden mit Hilfe einer Kette von Bildverarbeitungstools ermittelt. Da bei Manuskripten und bei manuellen Änderungen in Dokumenten die automatische Zeichenerkennung unbrauchbare Ergebnisse liefert, wurden eigene Strategien entwickelt, die die Informationen aus der BNE nutzen. Im Workshop werden die eingesetzten Tools und Optimierungsstrategien vorgestellt.

Lexikalische Arbeit (Lokale Grammatiken, Semantik) (Röhrer)

Zur lemmatisierten Suche, Partikelverberkennung und semantischen Wortfeldern wurden spezielle Projektlexika entwickelt (Röhrer 2017). Die Lexika enthalten alle Wörter der zu durchsuchenden Edition und sind mit grammatischen Angaben und zum Teil mit zusätzlichen semantischen Informationen versehen. Diese Lexika und ein nachgestelltes optimiertes Part-of-Speech Tagging ist die Grundlage für die computerlinguistischen Methoden, die bei der regelbasierten Suche im Nachlass von Ludwig Wittgenstein eingesetzt werden.

Regelbasierte Suchmaschine (Babl)

Im Zentrum der FinderApps steht die Suchmaschine wf, ein multithreaded C++ Programm, das viele Anfragemöglichkeiten zur Suche implementiert: Einwort und Mehrwortsuche (mit internem Rankingverfahren) und reguläre Ausdrücke kombiniert mit linguistischen Anfragen (Morphologische Eigenschaften, POS-Tags, semantische und syntaktische Tags). Für das Rankingverfahren wird für jeden Suchtreffer die Relevanz zur Suchanfrage berechnet. Die Qualität für jeden Suchtreffer, die Distanz zwischen den einzelnen Wörtern und unterschiedlichen Belohnungs- und Bestrafungsparametern, gehen in die Berechnung der Relevanz ein. Die Treffer werden dann nach dieser sortiert und auf der Website ausgegeben. Durch dieses neuartige Ranking kann nun auch nach verschiedenen Wörtern gesucht werden, die im Text nicht direkt hintereinander stehen müssen.

NLP-Tool Semantische Ähnlichkeitssuche (Ullrich)

Zur Extraktion von semantisch ähnlichen Bemerkungen wurde das Analysetool WiTTSim (Ullrich 2018) entwickelt, welches anhand von semantischen und syntaktischen Features ähnliche Texte identifiziert. Da die enorm hohe Anzahl von etwa 100.000 Features in Kombination mit den zu vergleichenden 54.000 Bemerkungen eine effiziente Suche unmöglich macht, wurde ein semantisches Clustering-Verfahren vorgeschaltet (Ullrich 2019), welches durch Dimensionsreduktion und Gruppierung der Texte die Rechenzeit der Ähnlichkeitssuche um den Faktor 100 beschleunigt.

WEB-Oberfläche der FinderApps und Micorservices (Hadersbeck, Still)

Zur Arbeit mit WiTTFind wird dem User eine WEB-basierte FinderApp zur Verfügung gestellt, die über REST-APIs und „internet microservices“ mit den WAST-Tools kommuniziert. HTML5, Javascript und Bootstrap-css erlauben den Aufbau der WEB-page, die nahezu browserunabhängig die Schnittstelle zum Anwender darstellt.

Doppelseitiger Faksimilereader und MongoDB (Lindinger)

Der doppelseitiger Faksimilereader ist eine komplett eigenständige Anwendung mit Suchschlitz und Investigate Mode zur gleichzeitigen Betrachtung von Faksimile und Transkription. Außerdem gibt es zahlreiche weitere Features, die es den Nutzern sehr bequem erlauben, die gefunden Treffer der Suchmaschine im Kontext einer doppelseitigen Darstellung der Faksimile zu sehen und gleichzeitig durch die Dokumente der Forschungsdomäne zu blättern. Sämtliche Informationen bzgl. Edition und Faksimile sind in einer MongoDB gespeichert und werden über HTTP-Schnittstellen abgefragt.

Dokumentationssystem Sphinx (Babl) (siehe Abb.2)

Für jedes Teilprojekt der Wittgenstein Advanced Search Tools (WAST) wird im entsprechenden Gitlab Ordner eine README.md Datei erstellt, das in einer Dokumentation, die alle Projekte umspannt mithilfe der Software Sphinx zusammengefasst und online auf ansprechende Art und Weise darstellt. Die Dokumentation hilft, neuen Studierenden einen schnelleren Einstieg in das Projekt zu finden und ermöglicht es, das gesamte WAST-Projekt schnell nach bestimmten Fachbegriffen zu durchsuchen.

Programm des Workshops (ganztages Workshop)

Überblick/Einführung/Vorstellungsrunde

Digitaler Zugang zum Nachlass von Ludwig Wittgenstein, das Projekt WAST (Dr. Max Hadersbeck)
Fragen/ Diskussion/ gewünschte Schwerpunkte der Teilnehmer*innen des Workshops

WAST-Spezialthemen (jeweils ca. 15 Min. Theorie / 20 Min. Praxis)

- Gitlab Projektmanagement und Continuous Integration mit git production / testing server (Hadersbeck, Still)
- XML TEI-P5 Edition CISWAB (Hadersbeck): Bergen Normalized Edition und xslt-Transformationen und Investigate-Mode von WiTTFind
- Faksimilestrukturierung und OCR Erkennung (Eisterhues, Landes)
- Lexikalische Arbeit (Röhrer): Lemmatisierte Suche, Lexika, Lokale Grammatiken, Query Beispiele
- WEB-Oberfläche der FinderApps und Microservices (Hadersbeck, Still): Flask server, Javascript
- Doppelseitiger Faksimilereader und mongodb (Lindinger)
- NLP-Tool Semantische Ähnlichkeitssuche (Ullrich): NLP-Python Libraries, Funktionalitäten
- Regelbasierte Suchmaschine (Babl): Programmierung C++, make/cmake, client-server Programmierung mit C++
- Dokumentationssystem Sphinx (Babl): Markdown, Sphinx Installation, 2HTML, 2PDF

Arbeitsgruppen: Diskussionen/Spezialfragen

Je nach Interesse der Teilnehmer*innen unter der Leitung der einzelnen Dozent*innen.

Kurzbiographie der Dozent*innen

Florian Babl (CIS)

Bachelorarbeit: Entwicklung eines Rankingverfahrens der Suchtreffer für die FinderApp WiTTfind im Nachlass Ludwig Wittgensteins
Forschungsschwerpunkte: verschiedene Rankingalgorithmen und ihre Funktionalität mit dem Ziel der Rankingverbesserung.

Marcel Eisterhues (CIS)

Forschungsschwerpunkte: Der momentane Forschungsschwerpunkt ist die automatische Seitensegmentierung von handgeschriebenen Texten.

Max Hadersbeck (CIS)

Projektleiter und Dozent am CIS
Forschungsschwerpunkte: Digitaler Zugang zum Nachlass von Ludwig Wittgenstein, FinderApp WiTTFind, Wittgenstein Advanced Search Tools, Programmierung: C ++, Python, XML

Florian Landes (Kommission für bayerische Landesgeschichte bei der Bayerischen Akademie der Wissenschaften)

Bachelorarbeit: Optical Character Recognition (OCR) – Optische Zeichenerkennung (OZE) Ein Werkzeug zur Verknüpfung von digitaler Edition und Faksimile? Semiautomatische Ermittlung von Bildkoordinaten für WiTTFind
Forschungsschwerpunkte: OCR, OZE, Bavarikonprojekt Ortsnamen des Regierungsbezirks Schwaben

Ines Röhrer (CIS)

Masterarbeit: Lexikon, Syntax und Semantik - computerlinguistische Untersuchungen zum Nachlass Ludwig Wittgensteins
Forschungsschwerpunkte: Digitales Speziallexikon WiTTLex für den Nachlass von Ludwig Wittgenstein

Sebastian Still (CIS)

Masterarbeit: Ludwig Wittgenstein: 100 Jahre Traktatus. Der Odyssee-Reader, ein web-basiertes Tool zur textgenetischen Suche im Traktatus
Forschungsschwerpunkte: moderne Frontend Programmierung, NLP (Backend)

Sabine Ullrich (CIS)

Masterarbeit: Clustering zur Verbesserung der Performanz einer Ähnlichkeitssuche

Forschungsschwerpunkte: Natural Language Processing, Data Mining, semantische Ähnlichkeitserkennung im Nachlass von Ludwig Wittgenstein

Bibliographie

Babl, Florian (2019): *Entwicklung eines Rankingverfahrens der Suchtreffer für die FinderApp WiTTFind im Nachlass Ludwig Wittgensteins*. Bachelor‘s thesis. LMU.

Landes, Florian (2019): *Optical Character Recognition (OCR) – Optische Zeichenerkennung (OZE). Ein Werkzeug zur Verknüpfung von digitaler Edition und Faksimile? Semiautomatische Ermittlung von Bildkoordinaten für WiTTFind*, Bachelorarbeit, LMU.

Lindinger, Matthias (2013): *Highlighting von Treffern des Suchmaschinentools WiTTFind im zugehörigen Faksimile.* Bachelor‘s thesis, LMU.

Lindinger, Matthias (2015): *Entwicklung eines WEB-basierten Faksimileviewers mit Highlighting von Suchmaschinen-Treffern und Anzeige der zugehörigen Texte in unterschiedlichen Editionsformaten*. Master's thesis, LMU.

Pichler, Alois (2017): *Wittgenstein Archives at the University of Bergen (WAB): Open Access to Wittgenstein's Nachlass. XML based Interactive Dynamic Presentation (IDP) of WAB's Nachlass transcriptions.* 16. Mai 2017. http://wab.uib.no/transform/wab.php?modus=opsjoner [letzter Zugriff 20.09.2019].

Hadersbeck, Maximilian / Pichler, Alois / Fink, Florian / Gjesdal, Øyvind L. (2014): „Wittgenstein's Nachlass: WiTTFind and Wittgenstein advanced search tools (WAST)“ , in: *Proceedings of the First International Conference on Digital Access to Textual Cultural Heritage*, 91-96. ACM.

Hadersbeck , Maximilian / Pichler, Alois / Bruder, Daniel / Schweter, Stefan (2016): *New (re)search possibilities for Wittgenstein's Nachlass II: Advanced Search, Navigation and Feedback with the FinderApp WiTTFind*. http://wab.uib.no/alois/Hadersbeck_Pichler%20Kirchberg2016.pdf [letzter Zugriff 20.09.2019].

Röhrer, Ines / Ullrich, Sabine / Hadersbeck, Maximilian (2019): *Weltkulturerbe international digital: Erweiterung der Wittgenstein Advanced Search Tools durch Semantisierung und neuronale maschinelle Übersetzung*. multimedial multimodal. Abstracts zur Jahrestagung des Verbandes Digital Humanities im deutschsprachigen Raum, 25. - 29.03.2019 an den Universitäten zu Mainz und Frankfurt.

Röhrer, Ines (2017): *Musik und Ludwig Wittgenstein: Semantische Suche in seinem Nachlass.* Bachelor‘s thesis, LMU.

Schmidt, Alfred (2018): „Ludwig Wittgenstein’s Nachlass in the UNESCO Memory of the World register.“, in: *Nordic Wittgenstein Review* 7(2):209–213.

Ullrich, Sabine / Bruder, Daniel / Hadersbeck, Maximilian (2018): Aufdecken von „versteckten" Einflüssen: Teil-Automatisierte Textgenetische Prozesse mit Methoden der Computerlinguistik und des Machine Learning. Kritik der digitalen Vernunft. Abstracts zur Jahrestagung des Verbandes Digital Humanities im deutschsprachigen Raum, 26.02.-02.03. 2018 an der Universi tät zu Köln, veranstaltet vom Cologne Center for eHumanities (CceH).

Ullrich, Sabine (2019): *Boosting Performance of a Similarity Detection System using State of the Art Clustering Algorithms*. Master‘s thesis. LMU.