

# Wittgensteins Nachlass: Aufbau und Demonstration der FinderApp WiTTFind und ihrer Komponenten

Yuliya Kalasouskaya, Matthias Lindinger, Stefan Schweter, Roman Capsamun  
Y.Kalasouskaya1@campus.lmu.de, matthias.lindinger@campus.lmu.de,  
Stefan.Schweter@campus.lmu.de, r.capsamun@campus.lmu.de  
Centrum für Informations- und Sprachverarbeitung (CIS), LMU, München

## 1 EINLEITUNG

Das von uns erstellte Poster soll den Aufbau und Einsatz der FinderApp WiTTFind mit den zugehörigen WAST-Tools<sup>1</sup> als *open source* Tool vorstellen. Im Mittelpunkt stehen die optimierte Browseroberfläche, zugrunde liegende Texte der FinderApp, Faksimile mit OCR, Faksimile Reader und den Einsatz des Finders als open source Programm. Für Interessierte werden wir die FinderApp vorführen.

## 2 NEUERUNGEN DER FINDERAPP

Seit 2 Jahren wird mit unserem Finder in der Nachlassforschung von Ludwig Wittgenstein gearbeitet und die von uns entwickelten Programme werden stetig optimiert. Eine zusätzliche Motivation für die Weiterentwicklung und Erweiterung war auch die Verleihung des EU-AWARDS 2014, der vom EU Projekt Digitised Manuscripts to Europeana (DM2E) ausgeschrieben wurde. Neuerungen des Finders bestehen darin, dass die Weboberfläche optimiert wurde, mehrere Dokumente parallel durchsucht werden können und eine lemmatisierte symmetrische Vorschlagsuche sowie ein neuer Faksimile-Reader integriert wurden. Die wichtigste Neuerung bei unserem Finder ist jedoch, dass WiTTFind für andere Forschungsprojekte geöffnet wurde und als open source für andere Projekte der Digital Humanities einsetzbar sein wird. Die Applikation ist unter dem folgenden Permalink zu erreichen:

<http://wittfind.cis.uni-muenchen.de>:

## 3 DIE KOMPONENTEN DES FINDERS

### 3.1 BENUTZEROBERFLÄCHE

Als erstes wollen auf dem Poster wir die Gestaltung der neuen benutzerfreundlichen und interaktiven Hauptseite der FinderApp vorstellen. Um die Bedienung der Anwendung übersichtlich zu gestalten, werden den Nutzern verschiedene Suchumgebungen angeboten. In der neuen Version können mehrere Text-Ressourcen parallel durchsucht werden, weshalb die Applikation um eine *multidoc*-Struktur erweitert wurde. Die Darstellung der Treffer wird auf die ausgewählten Dokumente beschränkt. In der nächsten Abbildung ein Beispiel zur *multidoc* Oberfläche:



Die Antwort auf die Benutzeranfrage wird mit Hilfe des *LocalStorage* Konzepts im Browser gespeichert. Bearbeitet werden die Daten und die *multidoc* Funktionen nur auf der Client Seite (Web-Browser) ohne Server-Zugriff. Zur Interaktivität und Lebendigkeit der Seite tragen die modernen Techniken von *JQuery* und *HTML5* bei.

<sup>1</sup> Wittgenstein Advanced Search Tools

### 3.2 FAKSIMILE-READER

Das zweite Thema des Posters stellt den Faksimile-Reader (siehe Bild:Faksimile-Reader) vor, der es erlaubt komplementär durch die Faksimile der Edition zu blättern und gleichzeitig die gefundenen Textstellen im Bild hervorzuheben. Dieser ist in Javascript sowie den Bibliotheken *jQuery* und *turn.js* programmiert. Zum *Highlighting* der einzelnen Treffer wird eine Liste von Koordinaten verwendet, die im Javascript-eigenen JSON-Format vorliegt. Zur schnellen Darstellung der Faksimile werden immer nur die Seiten geladen, die der Anfrage des Benutzers entsprechen. Somit kann der Anwendungsnutzer das komplette Dokument in Faksimile-Form durchblättern. Weitere Features des Readers sind eine dynamische Anpassung der Ansicht an das Browserfenster und eine kurze Bedienungsanleitung, die beim Start angezeigt wird. Damit die Faksimile zusammen mit den gefundenen, farblich hervorgehobenen Treffern dargestellt werden können, müssen die Faksimile mit der open source OCR Software *tesseract* verarbeitet werden. Je nach Qualität der Faksimile müssen die extrahierten OCR-Ergebnisse manuell nachbearbeitet werden. Dazu haben wir spezielle Tools entwickelt.

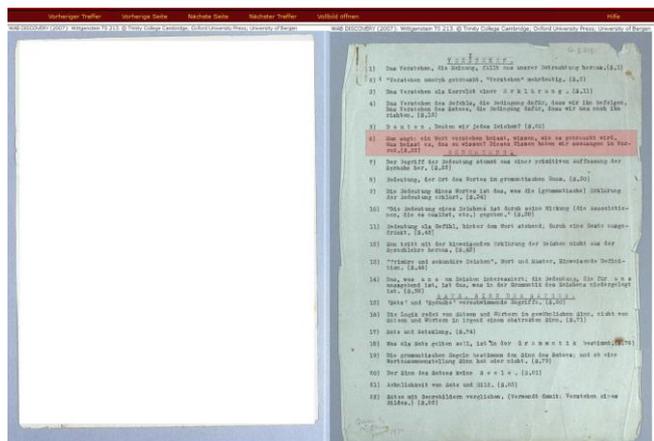


Bild:Faksimile-Reader

<sup>2</sup> Laboratoire d'Automatique Documentaire et Linguistique, Paris

## 4 UNSERE FINDERAPP FÜR ANDERE DIGITAL HUMANITIES PROJEKTE

In einem weiteren Thema des Posters geht es um eins der wichtigsten Ziele unseres Projekts: Die FinderApp und die WAST-Tools sollen plattformunabhängig einem breiten Forschungskreis zur Verfügung stehen.

### 4.1 DIE TEXTE DER EDITION

Die FinderApp findet Wörter, semantische Begriffe und Satzphrasen über mehrere Dokumente hinweg, sofern die Dokumente in dem XML-TEI-P5 Format vorliegen. Dieses XML-Format wird von uns CISWAB genannt und in einer eigenen *Document Type Definition (DTD)* beschrieben.

### 4.2 ELEKTRONISCHES VOLLFORMEN-LEXIKON

Zu den Texten einer Edition benötigt die FinderApp ein elektronisches Lexikon im DELA Format<sup>2</sup>. Das CIS verfügt über das größte deutsche Vollformenlexikon, das bei der Entwicklung eines eigenen „Editionslexikons“ herangezogen werden kann.

### 4.3 SOFTWARE-KOMPATIBILITÄT UNSERER FINDERAPP

Da bei unseren Programmen eine große Anzahl unterschiedlicher Programmiersprachen und Libraries im Einsatz sind, die von Standarddistributionen abweichen, setzen wir die quelloffene Containervirtualisierungssoftware namens *docker* ein. Bei dieser Technologie werden alle von WiTTFind benötigten Programme, Module und Libraries in einem Softwarecontainer zusammengefasst. Jeder Anwender, der auf seinem Rechner die *docker* Software<sup>3</sup> installiert hat, kann unseren Finder mit WAST-Tools lokal auf seinem Rechner einsetzen. Zur Entwicklungsverwaltung verwenden wir das

<sup>3</sup> <https://www.docker.com/>

Versionsverwaltungsprogramms Git und das web-basierte Versionsverwaltungs-Management-Werkzeug GitLab.

Am Posterstand werden wir auf Laptops mit verschiedenen Betriebssystemen unsere FinderApp WiTTFind und WAST-Tools vorstellen.

## **5 PUBLIKATION UND AWARD**

---

EU AWARD: <http://dm2e.eu/open-humanities-awards-round-2-winners-announced/>

Max Hadersbeck, Alois Pichler, Florian Fink, Øyvind Liland Gjesdal: Wittgenstein's Nachlass: WiTTFind and Wittgenstein advanced search tools (WAST). Madrid, DATECH 2014: 91-96 <http://wast.cis.uni-muenchen.de/tutorial>