# Projektplan

Veranstaltung: Praktikum „Sprachtechnologie“

Betreuender: Prof. Dr.-Ing. Torsten Zesch

Thema: BibTeX-Konverter

Teammitglieder: Lars Lafleur, David Konieczny, Constantin Schulz, Jürgen Bullinger

## Problemstellung

Mithilfe des Tools BibTeX können in LaTeX Literaturverzeichnisse erstellt werden. Dazu wird eine Datenbank erstellt (.bib-Datei), in der die Referenzen in einem für jeden der BibTeX-Eintragstypen spezifischen Struktur abgespeichert werden. Jeder Eintrag besteht hierbei aus einem Schlüsselwort und einem vom Eintragstyp abhängigen Mindestmaß an Tags (insgesamt weiter als BibTeX-Code bezeichnet). Die auf diese Weise erzeugte Datei wird in LaTeX eingelesen und nach der Kompilierung wird abhängig vom angegebenen Bibliography-Style ein fertig formatiertes Literaturverzeichnis erzeugt.

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

Hierbei kann BibTeX-Code für eine vorgegebene Referenz bereits im Web zugänglich sein oder muss manuell geschrieben werden.

## Zielbild

Was soll unser Endprodukt können?

Wie soll es für den Anwender aussehen?

Im Rahmen des Praktikums soll ein BibTeX-Konverter gebaut werden, der aus einem (formatierten) String den passenden BibTeX-Eintragstyp ermittelt sowie die zugehörigen Tags befüllt.

Der String soll sowohl per Copy-Paste als auch nach Erzeugung mithilfe von OCR aus einer Bilddatei dem Konverter übergeben werden können.

Abgekürzte Namen und Begriffe (z. B. die Namen von Journals) sollen dabei per Websuche ergänzt werden und in vollständiger Form im BibTeX-Code gespeichert werden.

## Lösungsskizze

Welche Technologien setzen wir ein?

* Aufbau / Definition einer einheitlichen Umgebung
* Datenspeicher: Git oder Cloud (DVC)
* Nutzung von Miniconda?
* Welche Bibliotheken anbinden?

Wie kommen wir zum Zielbild?

* + OCR-Schnittstelle? -> Task: OCR-Tool suchen
  + Online-Suche zur Vervollständigung von Abkürzungen? Aufbau einer Datenbank mit CiteDev? Datenstruktur aufbauen?
  + Beschreibung der Ansätze: Tokenisierung des Strings bzw. Repräsentation der Ein- und Ausgabedaten, Pipeline -> auch als Task

Visualisierung der Ergebnisse?

(Schaubild von Eingabe zur Ausgabe inkl. der verwendeten Technologien/Tools)

## Work Breakdown Structure

### Datensammlung und -vorverarbeitung

* Datenquellen bestimmen
  + API-Dokus für Extraktion von Daten aus Webseiten
  + LaTeX-/BibTeX-Doku zur Erzeugung von Daten
* Sammlung von Beispieldaten (Literaturverzeichnisse)
  + Ermittlung des Umfangs
  + Sicherstellung der Diversität der Daten
* Datenbereinigung und -formatierung
  + Erzeugung des Inputs für das Training des Modells
  + Erzeugung von Testdaten
  + Ablage in GitHub?
* Extrahierung von Metadaten (Titel, Autor, Jahr usw.)

### b) NLP-Modellentwicklung

- Auswahl eines NLP-Modells (z.B. PyTorch)

- Training des Modells mit den Beispieldaten

- Evaluation der Modellleistung

5. Implementierung der Konvertierungsfunktion

- Entwicklung des Algorithmus zur Konvertierung von Metadaten in BibTeX-Format

- Implementierung einer Benutzerschnittstelle (optional)

6. Testing und Qualitätssicherung

- Funktionstests des Konvertierungsalgorithmus

- Benutzertests durchführen

- Fehlerbehebung und Optimierung

7. Dokumentation und Abschluss

- Erstellen von Projektdokumentationen (Benutzerhandbuch, Entwicklerdokumentation)

- Abschlussbericht erstellen

- Abschlussdemo

## Team

### Aufgabenverteilung

|  |  |
| --- | --- |
|  | *Aufgaben* |
| *Constantin* |  |
| *Lars* |  |
| *Jürgen* |  |
| *David* |  |

### Way-of-Working

* Wöchentliche Meetings über Zoom mit Diskussion von Zwischenergebnissen und Problemen sowie Planung der nächsten Schritte und Aufgabenverteilung

## Milestones

## Datensammlung

### Umfang

Zum Trainieren und Testen des entwickelten Modells muss eine ausreichend große Datenbasis geschaffen werden. Prinzipiell werden 2-Tupel aus einer Referenz (String) und zugehörigem BibTeX-Code benötigt. Mehrere hundert Millionen von BibTeX-Einträgen können aus dem Web abgezogen werden (s. unten unter Methode).

### Methode

Abzug von Referenzen in BibTex und anderen Formaten aus Literatursuchmaschinen:

Beispiel: <https://aclanthology.org/>

Beispiel: <https://www.semanticscholar.org/>

* Über 200 Millionen Einträge
* Referenzen in verschiedenen Formaten:

Ein Bild, das Text, Screenshot, Schrift, Reihe enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Schrift, Reihe, Zahl enthält.

Automatisch generierte Beschreibung

Ein Bild, das Text, Schrift, Reihe, Screenshot enthält.

Automatisch generierte Beschreibung

* Abzug von Referenzen in bibtex per kostenloser API möglich
* Überführung in das gewünschte Zielformat z.B. per Python

Ein Bild, das Text, Screenshot, Schrift, Zahl enthält.

Automatisch generierte Beschreibung

* Auf diese Weise kann bereits eine ausreichende Menge an BibTeX-Code gesammelt werden.

Neben dem genannten Beispiel gibt es zahlreiche weitere Datenbanken, aus denen ggf. die Referenzen zusätzlich in den typischen Zitierweisen wie APA, MLA, ACM, etc. abgezogen werden können. Für die unterschiedlichen Zitierstile finden sich Beschreibungen unter <https://www.scribbr.de/richtig-zitieren/uebersicht-zitierstile/>.

Alternativen:

* Abzug von Referenzen in unterschiedlichen Zitierweisen über Web Scraping/Web Crawling
* Erzeugung von Referenzen in unterschiedlichen Zitierweisen mithilfe von LaTeX unter Anwendung der unterschiedlichen Bibliography-Styles (s. dazu <https://www.reed.edu/it/help/LaTeX/bibtexstyles.html>); auf diese Weise kann auch genügend Bildmaterial von Referenzen erzeugt werden, falls das OCR-Feature umgesetzt wird; durch die implizite Erzeugung von Daten mithilfe von LaTeX muss darauf geachtet werden, keinen Bias in das Modell einzubringen.

### Charakteristiken

Struktur: Als 2-Tupel aus einer Referenz (String) und zugehörigem BibTeX-Code (Dictionary-Typ unterschiedlichen Inhalts) sind die Daten semistrukturiert.

Volumen: Mehrere (hundert) Millionen Beispiele

Data Fit:

* Validität: Falls Referenzstring- und BibTeX-Code aus dem Web abgezogen werden, muss überprüft werden, ob beide Arten der Literaturangabe zusammenpassen. Falls die Referenzstrings aus LaTeX heraus erzeugt werden, muss geprüft werden, ob für den jeweiligen Eintragstyp die Pflichtfelder des BibTeX-Codes aus dem Web befüllt sind. Nach Anwendung von LaTeX auf validen BibTeX-Code sind die erzeugten Tupel automatisch valide.
* Reliabilität: Gegeben, da der Abzug aus dem Web bzw. die Erzeugung mit LaTeX immer die gleichen Datensätze liefert.
* Repräsentativität: Beim Abzug aus dem Web muss darauf geachtet werden, dass der finale Datensatz von jedem Eintragstyp ausreichend viele Beispiele enthält (Volume). Außerdem müssen die Referenzen in unterschiedlichen Zitierweisen vorliegen (Variety).

Data Integrity:

* Of known provenance: Teilweise gegeben; Quelle aus dem Web bekannt, aber die Algorithmen, die die unterschiedlichen Zitierweisen auf den Webseiten ineinander umwandeln, sind (meist) unbekannt
* Well-annotated: s. jeweilige API, LaTeX-Doku
* High Volume: Gegeben, siehe oben
* Complete: Gegeben durch Verwendung unterschiedliche Eintragstypen und Zitierweisen
* Timely: Gegeben durch Verwendung von Daten aus regelmäßig gewarteten Datenbanken
* Multivariate: Pro Datensatz werden nur zwei Attribute benötigt.
* Atomic: Gegeben
* Consistent: Abhängig von der Anzahl der Quellen; bei gleichem BibTeX-Code könnten die Referenzen auf unterschiedlichen Webseiten Unterschiede aufweisen
* Clear: Gegeben
* Dimensionally structured: Nicht benötigt

## Evaluation und Test

Metrik definieren (Ähnlichkeitsmaß) für Testdaten

## Aufbereitung und Visualisierung

**Example 3: Software Product Development:**

1. Project Management
   1. Requirements Gathering
   2. Project Planning
   3. Resource Planning
2. Product Design
   1. UX Design
   2. UI Design
   3. Prototyping
3. Development
   1. Front-end Development
   2. Back-end Development
   3. API Development
   4. Database Design
4. Quality Assurance
   1. Testing
   2. Bug Fixing
   3. User Acceptance Testing
5. Deployment and Maintenance
   1. Deployment Planning
   2. Release Management
   3. Maintenance and Support