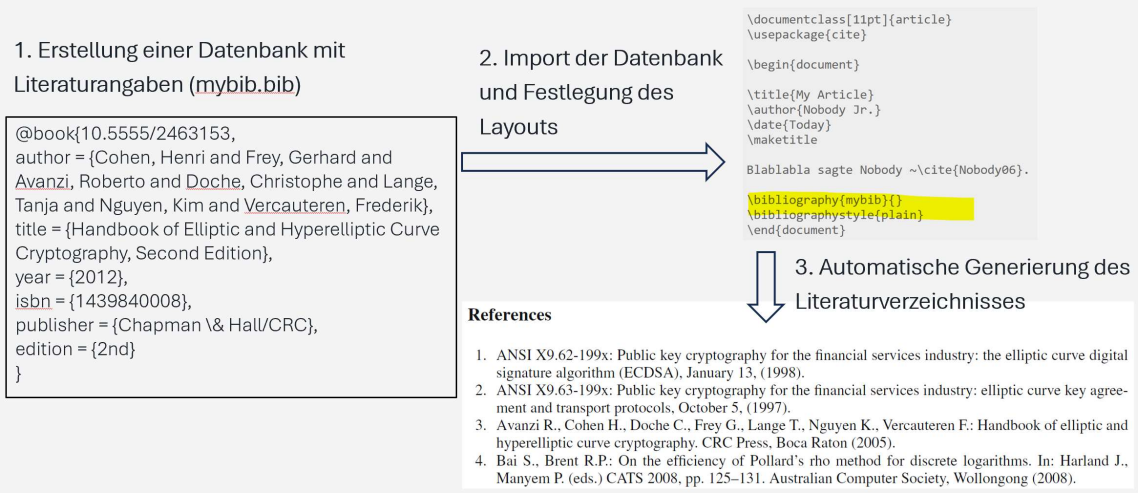
# Projektplan

| | |
|---|---|
| Veranstaltung: | Praktikum „Sprachtechnologie“ |
| Betreuender: | Prof. Dr.-Ing. Torsten Zesch |
| Thema: | BibTeX-Konverter |
| Teammitglieder: | Lars Lafleur, David Konieczny, Constantin Schulz, Jürgen Bullinger |

## 1. 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.




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

## 2. Lösungsskizze

### 2.1 Soll-Prozess

Der Anwender interagiert mit einem Webinterface, das in HTML, CSS und JavaScript geschrieben ist. Die Literaturangaben können dabei als String oder Bild in die Anwendung hochgeladen werden.

Wenn es sich um ein Bild handelt, wird es zuvor einem Modul OCR übergeben, das den Text extrahiert. Dafür kommen die Python-Bibliotheken OpenCV und Pytesseract zur Anwendung.

Anschließend werden die Literaturangaben dem Modul TextPreProcessing zugeführt. Dies wird mit der Bibliothek Spacy implementiert. Dort finden die Schritte

- Satzsegmentierung
- Tokenisierung
- Normalisierung
    - Entfernung von Stoppwörtern
    - Stammbildung und Lemmatisierung
    - Entfernung von bestimmten Interpunktionszeichen
    - Kleinschreibung
- POS-Tagging

statt.

Anschließend wird der vorverarbeitete String dem Modul Named-Entity-Recognition (NER) übergeben. Das ist das Kernmodul, das den bestimmten Bereichen des Strings die BibTex-Attribute

- Author
- Title
- Year
- Publisher
- Page
- ...

zuweisen soll. Die NER wird ebenfalls mit der Bibliothek Spacy umgesetzt.

Da bestimmte Namen abgekürzt sind oder Akronyme verwendet werden, sollen diese standardisiert ausgeschrieben werden. Dazu soll es ein Modul LookUp geben, das mithilfe der durch das NER erkannten Attribute und der Bibliothek scholarly nach Datensätzen sucht, die diesem BibTex-Eintrag entsprechen. Wenn ein gefundener Datensatz einen bestimmten Schwellenwert eines Ähnlichkeitsmaßes überschritten hat, wird dieser zur Datenanreicherung herangezogen.

Anschließend werden die generierten BibTex-Einträge auf einem zentralen Filesystem abgelegt.

## 2.2 Alternative Ansätze

Darüber hinaus sollen im Rahmen der agilen Arbeitsweise mögliche Alternativen wie das Fine-Tuning oder das Few-Shot-Prompting bei einem open-source LLM analysiert werden, falls sich der oben skizzierte Ansatz mit NER nicht umsetzen ließe.

# 3. Work Breakdown Structure

## 3.1 Projektinitiierung

- Projektziele und Scope festlegen

- Erfahrungen zum Thema NLP und Python im Team erfassen
- Tools festlegen
    - Kollaborationsplattform: GitHub
    - Kommunikationsplattform: Zoom, Discord
    - Entwicklungsumgebung: Anaconda
    - Programmiersprache und Bibliotheken analysieren und Dokumentationen sichten: Python, Spacy, OpenVC, Pytesseract, scholarly
    - Versionsverwaltung: GitHub
- Arbeitsweise festlegen: Scrum, Kanban
- Termine planen
- Projektplan erstellen

## 3.2 Analyse

- funktionale Anforderungen spezifizieren
    - Literaturverzeichnis in BibTex umwandeln
    - Webservice
    - Weboberfläche
    - Einzulesende Datenformate: Bilder, String
    - Ausschreiben von Akronymen
    - Ausschreiben von Abkürzungen
    - Vervollständigung fehlender Einträge mittels scholarly
- nichtfunktionale Anforderungen spezifizieren
    - Webserver
    - Betriebssystem
    - Datenbanken
    - Datencharakteristika
    - Performance: Mindestwerte Precession, Recall und F1-Score festlegen
- Technische Architektur
- Prozessbeschreibung
- Anforderungen priorisieren
- Glossar erstellen

## 3.3 Datensammlung

- Literaturangabe-Formate analysieren
- Datenquellen bestimmen
- Methodik: Web Scraping
- Repräsentativer Stichprobenumfang ermitteln
- Datenarchitektur aufbauen
    - Datenformat
    - Ort der Speicherung
    - Speichertechnologien
    - Datenzugriff / Datenschnittstelle

## 3.4 Explorative Datenanalyse / Datenqualität bestimmen

- Datencharakteristika bestimmen

- Datenattribute bestimmen
- Metrik zur Datenqualität bestimmen
    - Wann „fit for pupose“?
    - stichprobenartige Prüfung der Konsistenz
- Fehlertypen bestimmen
    - Duplikate
    - Fehlende Werte
- Erkenntnisse Visualisieren
    - Fehlerverteilung
    - Verteilung Attribute
    - Verteilung der Formate

## 3.5 Datenbereinigung

- Art von Datenbereinigungen (Imputation…) für Attribute analysieren
- Daten bereinigen

## 3.6 Datenvorbereitung

- Satzsegmentierung
- Tokenisierung
- Normalisierung
    - Entfernung von Stoppwörtern
    - Stammbildung und Lemmatisierung
    - Entfernung von bestimmten Interpunktionszeichen
    - Kleinschreibung
- POS-Tagging

## 3.7 Feature Engineering / Text Representation

    - Merkmalsextraktion
    - Merkmalsauswahl
    - Merkmalskonstruktion

## 3.8 NLP-Modellierung

- NLP-Modelle / NLP-Algorithmen analysieren
    - Named-entity recognition
    - LLM
- NLP-Modell auswählen
- Modelle trainieren / validieren
    - K-fold Cross-Validation
- Modelle evaluieren
    - Evaluations-Metriken: Precession, Recall, F1-Score
    - Benchmarking

### 3.9 Deployment

### 3.10 Abschlusspräsentation erstellen

### 3.11 Dokumentation erstellen

## 4. Team

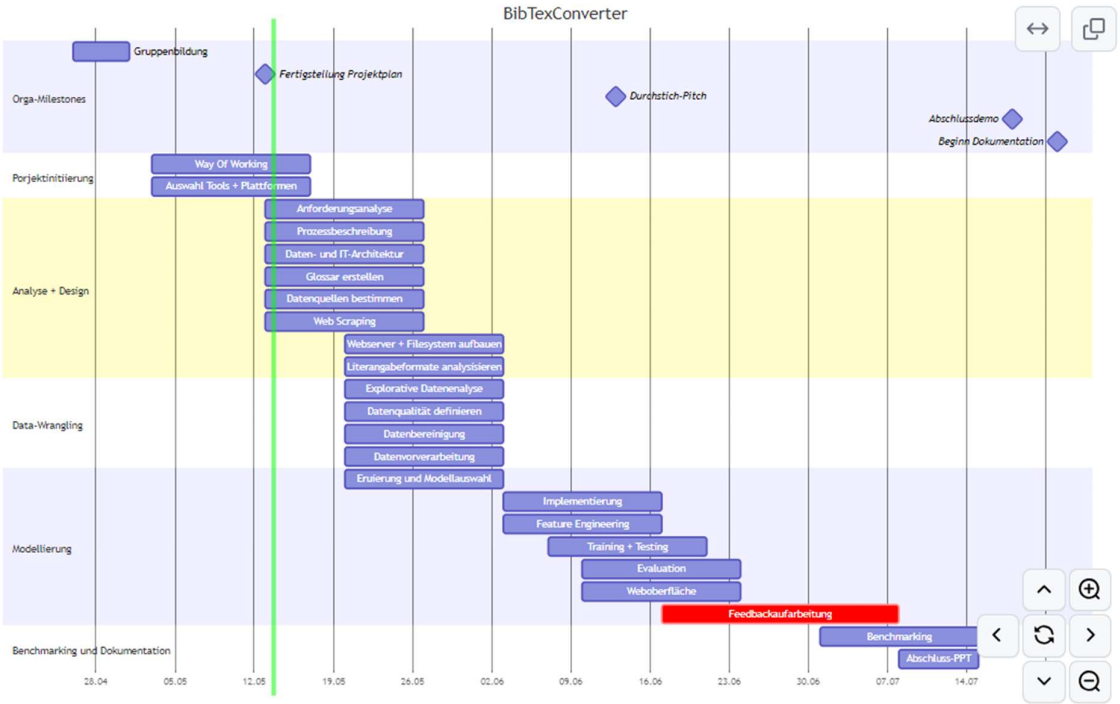
### Aufgabenverteilung

| | *Aufgaben* |
|---|---|
| *Constantin* | t.b.d. |
| *Lars* | t.b.d. |
| *Jürgen* | t.b.d. |
| *David* | t.b.d. |

### Way-of-Working

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

## 5. Milestones



# 6. Datensammlung

## 6.1 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).

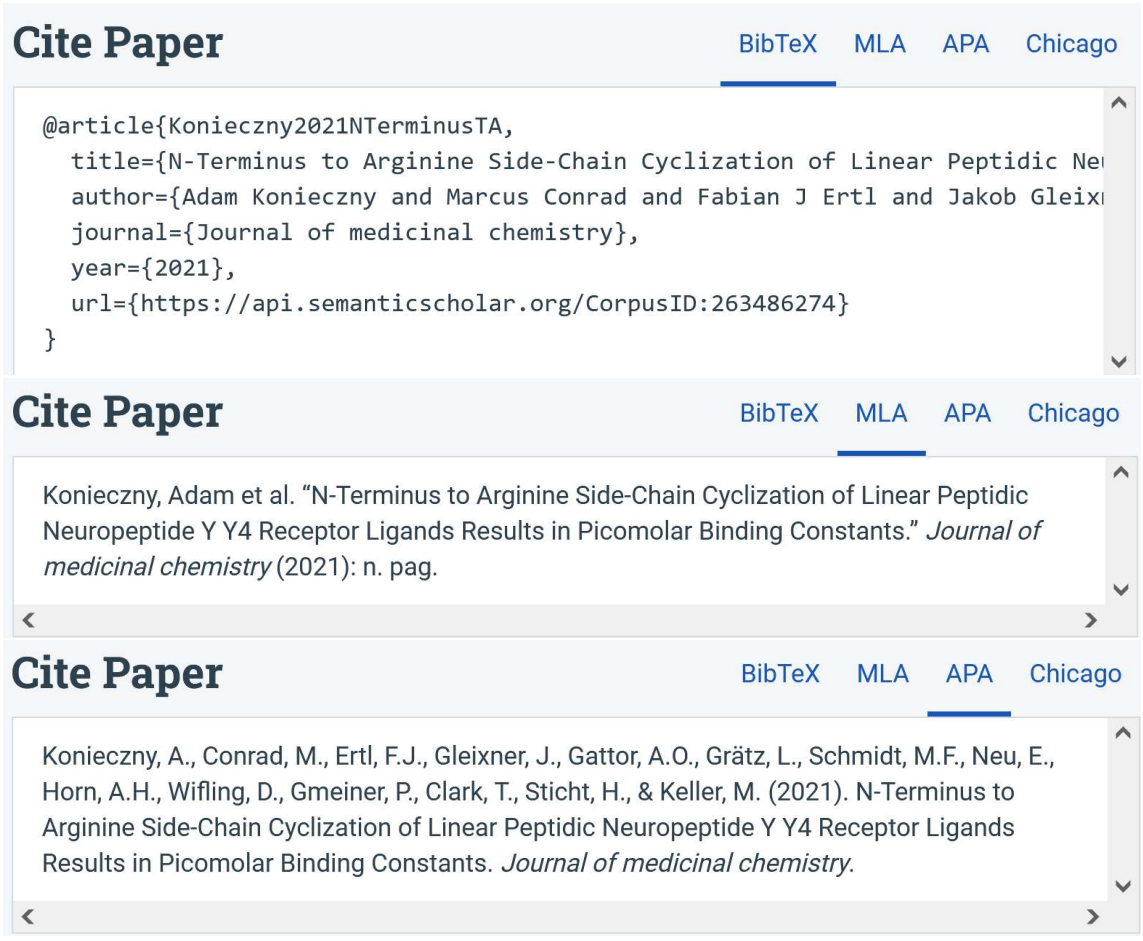
## 6.2 Methode

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:




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

| | BibTeX |
|---|---|
| 0 | @Inproceedings{Vetter1982ScaleFA,\n author = {... |
| 1 | @Article{Hirata2015EffectOI,\n author = {T. Hi... |
| 2 | @Inproceedings{LIShidong2004ReviewOC,\n author... |
| 3 | @Inproceedings{Dizon2019BhutanPN,\n author = {... |
| 4 | @Article{Endo2023IntermolecularIS,\n author = ... |
| 5 | @Article{Rachmawati2021GrowthOM,\n author = {Y... |

- 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.

## 6.3 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

## 7. Evaluation und Test

Grundsätzlich können alle vortrainierten statistischen Modelle aus Spacy genutzt werden. Lediglich die NER-Komponente der NLP-Pipeline muss auf die Problemstellung hin optimiert werden, sodass diese Komponente von Grund auf neu trainiert werden muss. Die NER-Komponente soll dabei mittels Supervised Learning trainiert werden. Das Modell bekommt als Input eine Literaturangabe und die richtige BibTex-Ausgabe übergeben.

Als Resampling-Verfahren soll k-fache Kreuzvalidierung angewendet werden.

Für die Evaluation einer NER kommen Precession, Recall und F1-Score in Frage.

## 8. Aufbereitung und Visualisierung

Die Datensätze (Attribute) und die Fehler sollen mittels Histogrammen visualisiert werden.

Um die Performance des Modells zu beobachten, soll die Lernkurve visualisiert werden.

Die Gegenüberstellung der Ergebnisse zwischen dem selbst erstellten Modell und dem Benchmark-Modell wird für die Klassifikation mithilfe einer Konfusions-Matrix und für Recall, Precession und F-Score mithilfe einer Tabelle erfolgen.

Es wird eine ROC-Kurve visualisiert werden, um Precession und Recall abzuwägen.