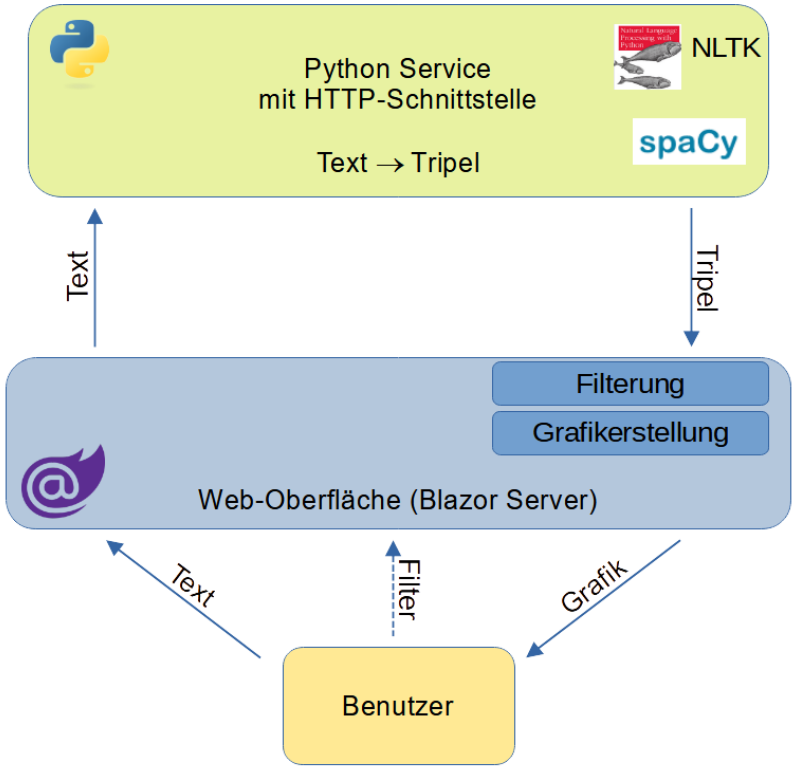
Fachpraktikum Sprachtechnologie

Projektplan zum Thema 4: Generierung von Concept-Maps aus Texten

Erstellt durch Korvin Felix Andreas Walter, Adriana Mikuteit, Sven Nicolai

1. Grobe Lösungsskizze

Es soll eine Anwendung zur automatischen Generierung von *Concept Maps* aus einem vom Benutzer eingegebenen Text entwickelt werden.



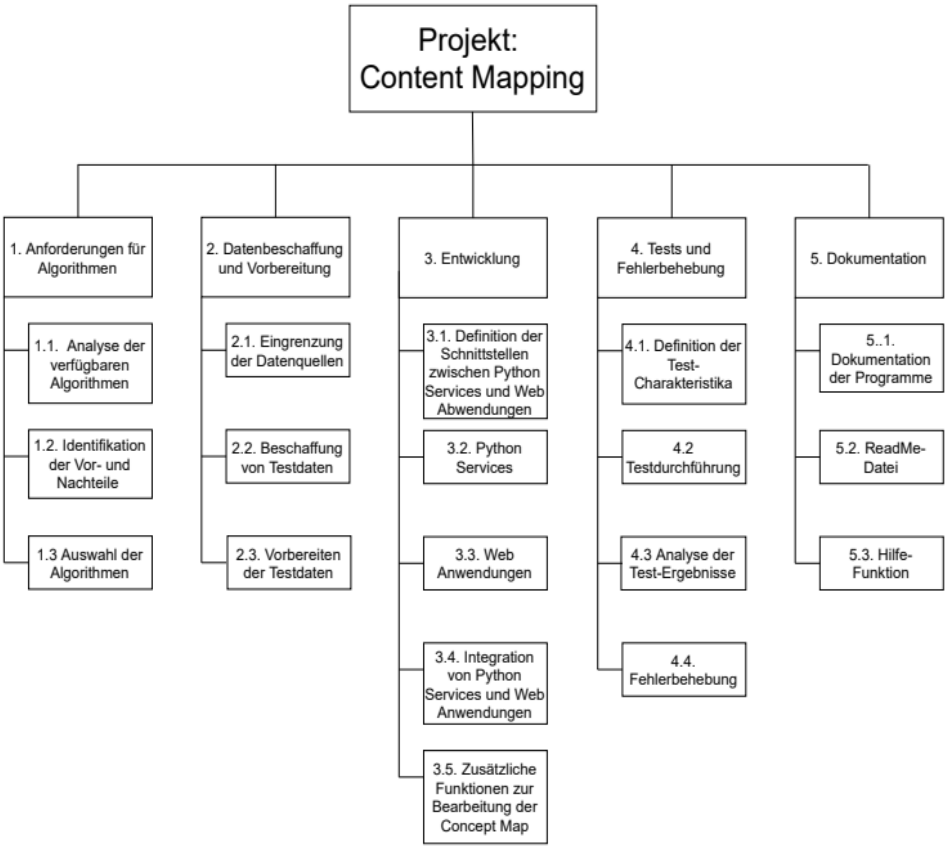
Die Benutzeroberfläche bildet eine Blazor (Server) Webanwendung, welche in C# geschrieben wird. Dies erleichtert die Entwicklung enorm, weil kein JavaScript-Code und keine separaten APIs für die Schnittstelle *Browser → Server* implementiert werden müssen.

Um die *Concept Map* aus einem Text zu generieren, wird zunächst ein Python-Service im Hintergrund aufgerufen:

- Zunächst werden aus dem Text die Tripel extrahiert. Jedes Tripel stellt die Beziehung von Wörtern untereinander dar. Dies soll mit Hilfe von Toolkits wie NTLK und spaCy geschehen.
- Anhand diesen Tripel kann dann die Grafik für die *Concept Map* erstellt werden. Der Benutzer soll jedoch die Möglichkeit haben, die Erstellung, z.B. mittels Filtern zu beeinflussen. Die vorher ermittelten Tripel könnten beispielsweise mittels Blocklist, Allowlist o.ä. gefiltert werden, bevor daraus die Grafik erstellt wird.
- Anschließend könnten die Elemente der Grafik automatisch und manuell angeordnet werden.

Die Bereitstellung (Deployment) soll mittels Docker-Containern ermöglicht werden. Mittels Docker Compose wäre das Deployment aller Container gemeinsam mit einem Konsolenbefehl machbar.

2. **Definition Work Breakdown Structure**



1. Analyse der Anforderungen (Klärung welcher Algorithmus)
2. Datenbeschaffung und -vorbereitung
3. Entwicklung
4. Test- und Fehlerbehebung
5. Dokumentation

3. Team

3.1 Wer kümmert sich um was?

Block	Detail	Hauptverant-wortlich
Analyse der Anforde-rungen	Literaturrecherche	alle
	Aufbereitung der Ergebnisse in Vor- und Nachteilen	Adriana
	Auswertung und Auswahl der Algorithmen	alle
Datenbeschaffung und -vorbereitung	Datensammlung mittels Webcrawler	
	Vorbereitung der Daten	
Definieren der Schnitt-stellen zwischen Py-thon und der Weban-wendung	gemeinsame Festlegung (da Grundlage der Webanwendung/Python Anwendung	alle
Entwicklung der Py-thon-Anwendung	NLTK	Adriana
	PyTorch	Korvin
	spaCy	
	Training der Modelle mit den gesammelten und bearbeiteten Daten	
Entwicklung der We-banwendung		Sven
Integration der Schnitt-stellen	Integration der Python-Anwendung in die Webanwendung unter Nutzung der Schnitt-stellen	Sven
Visualisierung von Concept Maps	Implementierung von Funktionen zur Visua-lisierung und Bearbeitung von Concept Maps	Adriana
Testing und Fehlerbe-hebung	Festlegen von geeigneten Charakteristiken (Katalog)	Korvin
	Planung und Durchführung von Testcases	Korvin
Dokumentation	Webanwendung	Sven
	Python	Adriana/ Korvin
	Statusupdate (Berichte)	Alternierend (alle)
	Präsentation	alle
	Democase	alle

3.2 Way-of-Working

- Hauptverantwortliche benannt, aber gleichzeitig ist jedes Teammitglied in der Lage, den aktuellen Stand der Dinge präzise zu kommunizieren.
- Agile Entwicklungsmethode mit zweiwöchigen Statusreports (1 Seite)
- Folgende Tools werden genutzt: GitHub, Python, Blazor

4. **Grobe Zeitplanung**

Woche 1 (04.05. - 17.05.):
Analyse der Anforderungen und Klärung des geeigneten Algorithmus
Festlegung der erforderlichen Daten für die Anwendung
Erstellung eines detaillierten Projektplans mit Meilensteinen

Woche 2 (11.05. - 17.05.):
Datenbeschaffung und -vorbereitung (z. B. Daten sammeln, bereinigen, formatieren)
Definition der Schnittstellen zwischen Python und der Webanwendung

Woche 3-4 (18.05. - 12.06):
Datenbeschaffung und -vorbereitung (z. B. Daten sammeln, bereinigen, formatieren)
Entwicklung der Python-Anwendung
Entwicklung der Webanwendung

Woche 5-6 (01.06. - 14.06.):
Integration der Python-Anwendung in die Webanwendung unter Verwendung der definierten Schnittstellen (Parallel zur Entwicklung der Anwendungen)

Woche 7-8 (15.06. - 28.06.):
Durchführung von umfangreichen Tests und Fehlerbehebung
Erstellung der Projektdokumentation (z. B. technische Dokumentation, Benutzerhandbuch)

Milestones:
Abschluss der Anforderungsanalyse und Klärung des Algorithmus (17.05.)
Abgeschlossene Datenbeschaffung und -vorbereitung sowie definierte Schnittstellen (31.05.)
Entwicklung der Python-Anwendung abgeschlossen (12.06)
Entwicklung der Webanwendung abgeschlossen (12.06)
Erfolgreiche Integration der Python-Anwendung in die Webanwendung (14.06.)
Prototyp Pitches “Durchstich” (15.06)
Durchgeführte Tests und behobene Fehler (28.06.)
Vollständige Dokumentation des Projekts (11.07.)
Abschlussdemo (12.07)

5. **Beschreibung Datensammlung**

Die Definition der Datensammlung ist in unserem Fall auf zweierlei Bereich möglich, hier listen wir beide Varianten auf. Während der Projektphase grenzen wir es auf einen der beiden Bereiche ein.

Beschreibung Datensammlung

Umfang

- Datensammlung aus dem Internet für verschiedene Stammbäume.
  Die Eingrenzung erfolgt hierbei auf Literarische Werke, da hier im Internet eine große Datenbasis zu finden ist. Folgende Websites werden als Beispiel genannt:
  https://gameofthrones.fandom.com/de/wiki/Game_of_Thrones_Wiki
  https://lotr.fandom.com/de/wiki/Der_Herr_der_Ringe_Wiki?campaign=explore-09-2018
  https://starwars.fandom.com/wiki/Main_Page
  https://harrypotter.fandom.com/wiki/Main_Page

Charakteristik

- Englisch
- Spezifische Domäne: Familienstammbäume aus der Literatur
- Daten müssen den Verwandtschaftsgrad der Personen zueinander beschreiben

Methode

- Auswahl von Werken
- Webcrawler, der die Inhalte der gewählten Websites auf die entsprechenden Merkmale untersucht
- Speicherung der Daten in ein Format, welches im nachfolgenden Prozess durch das zu trainierende Modell eingelesen werden kann (z.B. Json oder RDF)

6. Evaluation & Test

Für den Test und die Evaluation der Concept Maps des fertigen Programms sollen quantitative und qualitative Kriterien verwendet werden, um den Erfolg des Projekts zu verifizieren. Hierdurch sollen auch die zwei Resultate der beiden Python Services (NLTK, PyTorch, spaCy) miteinander verglichen werden, um mögliche Vor- und Nachteile der beiden Implementierungen identifizieren. Hierzu wird parallel zum laufenden Projekt ein Kriterien-Katalog entwickelt und ein Set von Testcases erstellt. Quantitative Kriterien könnten z.B. die korrekte Anzahl von identifizierten Objekten und Relationen zu einem Test-Text oder eine Verteilungsanalyse der Relationen sein. Qualitative Bewertungen können durch die subjektiven Bewertungen der Maps z.B. in Hinsicht auf Übersichtlichkeit erfolgen.
Des Weiteren wird auch die Web-Anwendung mittels qualitativer Kriterien auf die Benutzerfreundlichkeit und mittels der Verwendung von Testcases auf ihre Zuverlässigkeit getestet.

Kurz und prägnant:
Auswertung der Concept Maps (Quantitativ/Qualitativ)
Katalog zur Bewertung der erstellten Concept Map
Testcases werden definiert und auf Basis der Bewertungsmatrix als Erfolgreich/nicht erfolgreich klassifiziert.
Die Concept Maps werden auf ihre Genauigkeit und Relevanz in Bezug auf die Texte, aus denen sie generiert wurden, überprüft.
Die Webanwendung wird auf ihre Benutzerfreundlichkeit und Zuverlässigkeit getestet.

7. Aufbereitung & Visualisierung

Die Dokumentation inklusive Readme-Datei wird bereits parallel zur Entwicklung der Programmteile geschrieben. Sowohl für die Zwischenpräsentation als auch der Abschlusspräsentation werden Demo-Cases vorbereitet, anhand derer die Funktionalitäten des Programms gezeigt werden können. Die quantitativen und qualitativen Ergebnisse der abschließenden Programm-Evaluation werden für die Abschlusspräsentation übersichtlich dargestellt.

Kurz und prägnant:
Showcase vorbereiten (Democase)
Dokumentation wird parallel gepflegt (Readme)
Präsentation der Ergebnisse
Aufbereitung der Testergebnisse

7.1 Aufbereitung

„Wie werden eigentlich die Tripel aus dem Text extrahiert?“

Innerhalb der Toolkits gelieferten Funktionen werden genutzt, um die Schritte Text-zu-Triplet, umzusetzen (z.B. bei NTLK)

1. Tokenisierung
   - Satz wird in einzelne Segmente (Wörter) unterteilt
2. Stemming und/oder Lemmatisierung
   - Umwandlung des Wortes in Basisform
3. Stop-Words
   - Entfernung von „a“, „the“,“on“ etc.
4. POS Tagging
   - Kategorisierung der Wörter nach Nomen, Verben etc.
5. Parse-Tree (Chunking, NER)
   - Zugehörigkeit der einzelnen Wörter im Satz strukturieren

Da wir uns hier auf Familienstammbäume begrenzen, werden die Wörter nach dem Abschnitt POS Tagging für die Concept Map gefiltert bzw. werden nur relevante Wörter in die Concept Map überführt. Wichtig ist, dass die Wörter des Textes bereits auf ihre Basisform umgewandelt wurden und entsprechend getaggt wurden.
Eine Idee wäre es eine Sammlung von entsprechenden Wörtern mit Hilfe der Datensammlung aufzubauen (z.B. Tochter, Sohn etc.). Danach werden wir ein Modell mit diesem annotierten Datensatz trainieren, um so aus den neuen Texten (Input des Users) Muster zu erkennen und die gewünschten Informationen zu extrahieren. Hierbei gibt es unterschiedliche Toolkits/Algorithmen, mit denen ein Modell trainiert werden kann (z.B. Naive Bayes Classifier, NER, Decision Trees etc.), diese werden im Rahmen des Projektes analysiert und eine geeignete Wahl getroffen.

7.2 Visualisierung

Der User kann seinen Text über ein Textfeld innerhalb der GUI eintragen. Über eine Schnittstelle wird der Text in Richtung des Python Services übergeben. Innerhalb des Python Services wird nun der Text in die Triplets zerlegt und zurück in Richtung GUI übertragen, um dort die Visualisierung der Concept Map zu gewährleisten.

Zusätzlich wird mittels der GUI eine Anpassung der Concept Map möglich sein. Dafür wurden sich folgende Funktionalitäten überlegt:

1. Automatische Anpassung der Concept Map:
   Die Anordnung der Concept Map soll automatisch angeordnet werden, damit die Grafik visuell ansprechend für den User ist.

2. Manuelle Anpassung der Concept Map
   Der Nutzer hat die Möglichkeit die Concept Map zusätzlich manuell anzupassen. Hier soll die Möglichkeit etabliert werden, die Objekte zu schieben.

3. Filterung

   Der Nutzer bekommt die Möglichkeit wichtige (Include) und unwichtig (Exclude) Objekte zu filtern. Dabei können z.B. nicht gewünschte Objekte in einer Concept Maps ausgeblendet werden.