Fernuniversität Hagen

Modul 63458 - Fachpraktikum

Natural Language Processing, Information Extraction und Retrieval

WiSe 2022/23

Exposé

Automatische Erstellung einer Wissensrepräsentation aus einem medizinischen Text

eingereicht von

Anne Koch, Clara Jansen, Dietrich Tönnies

Betreuer: Prof. Dr. Hemmje
Dr. Christian Nawroth
Stephanie Heidepriem, M.Sc.

Hagen, 10. November 2022

INHALTSVERZEICHNIS

1 EINLEITUNG

Diese Praktikumsarbeit behandelt die automatische Erstellung einer Wissensrepräsentation aus einem medizinischen Text. Eine prototypische Softwarelösung wird entwickelt, die für einen vorgegebenen Beispieltext einen Wissensgraphen ermittelt.

Gemäß einem Artikel des *Economist* von 2017 [**the_economist_worlds_2017**], der in Verbindung mit den aktuellen technischen Entwicklungen, Künstlicher Intelligenz und Data Science viel zitiert wird, ist nicht mehr Öl die wertvollste Ressource, sondern Daten. Ebenso wie Öl müssen die Daten jedoch erst aufbereitet werden, um tatsächlich von Nutzen zu sein. Auch Informationen, die in Form von für den Menschen lesbaren Texten zur Verfügung stehen, sind nicht direkt für Computer auswertbar. Daher wurden mit den Methoden des *Natural Language Processing* (NLP) Verfahren entwickelt, um automatisch Informationen aus Texten extrahieren zu können. Die allgemeinsprachlichen NLP-Verfahren bedürfen jedoch noch einer Spezialisierung, um auch für Fachtexte wie zum Beispiel aus dem Bereich der Medizin nutzbringend angewendet werden zu können.

Aus diesen Rahmenbedingungen ergibt sich die Motivation und die Problembeschreibung für diese Praktikumsaufgabe, die in den beiden nachfolgenden Abschnitten beschrieben werden.

1.1 MOTIVATION

Die spezielle Herausforderung der Praktikumsaufgabe liegt darin, automatisiert Zusammenhänge zwischen Entitäten medizinischer Texte zu finden und diese Zusammenhänge zu repräsentieren. Dem zugrunde liegt das Ziel des Dissertationsprojektes von Stephanie Heidepriem, zu erforschen, wie ein textbasierter Chatbot zur Unterstützung psychisch kranker Menschen entwickelt werden kann.

Das Dissertationsprojekt ist unter anderem an das Projekt *MENHIR* angelehnt [**menhir**]. Dieses Projekt dient der Erforschung von Konversationstechnologien, die psychisch kranke Menschen unterstützen sollen. In diesem Zusammenhang wird auch ein MENHIR-Chatbot entwickelt, der personalisierte Unterstützung und hilfreiche Bewältigungsstrategien bieten soll.

Das Forschungsprojekt *STop Obesity Platform* beschäftigt sich mit der Extraktion von Wissen unter anderem aus Chatbots, das anschließend aufbereitet und mit weiteren Informationen kombiniert werden soll. Dieses Wissen soll medizinischem Fachpersonal zur Verfügung gestellt werden und außerdem Menschen mit Adipositas dabei helfen, eine gesündere Ernährung einzuhalten [**stopobesity**].

Es ist aber auch denkbar, dass die Problemstellung der Praktikumsaufgabe auch darüber hinaus für weitere Anwendungen interessant ist und mögliche Ergebnisse in verschiedenen Gebieten eingesetzt werden können, in denen Informationen in (medizinischen) Texten zueinander in Zusammenhang gebracht werden sollen.

Es wurde bereits viel an Methoden gearbeitet, die in der Lage sind Entitäten, also wichtige Objekte wie zum Beispiel Personen, Organisationen oder Orte, automatisiert aus Texten zu extrahieren. Dies nennt man Named Entity Recognition. Named Entity Recognition ist ein Bereich des Natural Language Processing und seit circa 30 Jahren werden unterschiedliche Techniken zum Lösen der Aufgaben, die in diesen Bereich fallen, entwickelt [Quelle: Recent Trends NER]. Darunter fallen grammatikbasierte und statistische Methoden wie auch Methoden des Machine Learning.

Außerdem aktualisiert die Ünited States National Library of Medicine"regelmäßig ein auf den medizinischen Bereich spezialisiertes Vokabular, beziehungsweise eine Wissensbasis, MeSH [**mesh**] Die Ünited States National Library of Medicine"hat unter anderem auch schon an einem Named-Entity Recognizer, "MetaMapLite", gearbeitet und stellt diesen zur Verfügung. [**metamaplite**]

1.2 PROBLEMBESCHREIBUNG

Es soll eine Konsolenapplikationen entwickelt werden, die englischsprachige medizinische Texte analysiert und das darin enthaltene Wissen strukturiert in einem RDF-Graphen hinterlegt. Die Konsolenapplikation soll in Python programmiert werden, wobei für die Textanalyse Klassen und Methoden der Open-Source-Bibliothek *spaCy* o.ä. genutzt werden sollen.

Die medizinischen Texte enthalten z.B. Aussagen zu Krankheiten (z.B. Depression) und zählen deren Symptome (z.B. Motivationsverlust) auf. Aufgabe der Konsolenapplikation ist es, Schlüsselbegriffe im Text zu finden und miteinander in Bezug zu setzen, indem z.B. Symptome den ihnen zugrunde liegenden Krankheiten zugeordnet werden. Die von *spaCy* oder anderen NLP-Bibliotheken zur Verfügung gestellte Funktionalität besteht aus einer Pipeline von Analyse-Komponenten, die nacheinander auf den zu analysierenden Text angewendet werden. Diese Komponenten dienen dazu, Texte in einzelne Sätze zu zerlegen und die Sätze grammatikalisch zu analysieren.

PROBLEM 1: VORVERARBEITUNG EINES TEXTES Medizinische und wissenschaftliche Texte bestehen häufig aus Aufzählungen und Zwischenüberschriften. Der Text sollten so aufbereitet werden, dass anschließende Analyse-Algorithmen möglichst erfolgreich arbeiten können. Es soll erprobt werden, wie dies am besten bewerkstelligt werden kann, z.B. durch Umwandlung von Aufzählungen in mehrere Aussagesätze. Auch sollte automatisch erkannt werden, welche Sätze Aussagen (Wissen) formulieren etwa durch Unterscheidung von Indikativ und Konjunktiv oder durch Ignorieren von Fragesätzen.

PROBLEM 2: TRAINING DER NER - KOMPONENTE Neben der grammatikalischen Analyse besteht eine wesentliche Aufgabe von *spaCy* oder anderen NLP-Bibliotheken darin, Schlüsselbegriffe zu finden und nach Möglichkeit zu kategorisieren. Diese Art der Analyse wird als *Named Entity Recognition* (NER) bezeichnet. Bei *spaCy* übernimmt diese Aufgabe der regelbasierte *Entity-Ruler* oder der auf statistischen Modellen basierende *Entity-Recognizer*. Eine weitere Komponente ist der *Entity-Linker*, mit dem Begriffe eindeutig den in einer Wissensbasis gespeicherten Entitäten zugeordnet werden können. Auch der *Entity-Linker* basiert auf statistischen Modellen und wird mit Beispielsätzen trainiert.

Die Standard-NER-Funktionalität von *spaCy* erkennt medizinische Begriffe nur unzureichend. Es ist daher notwendig, eine Datenbank mit medizinischen Begriffen und Kategorien zusammenzustellen, die für das Training der NER-Komponente verwendet werden kann.

Die *United States National Library of Medicine* (NLM) stellt mit dem *Unified Medical Language System* (UMLS) ein mächtiges Werkzeug für die Textanalyse zur Verfügung. Teil von UMLS ist der sogenannte Metathesaurus, der aus einer Vielzahl von Thesauri unterschiedlicher Organisationen zusammengestellt wird. Zu diesen Thesauri gehört u.a. der von der NLM entwickelte Thesaurus *Medical Subject Headings* (MeSH). MetaMap und das weniger umfangreiche MetaMapLite sind eigene *Entity-Recognition*-Werkzeuge der *National Library of Medicine*. Die zugrundeliegende Datenbasis dieser Werkzeuge sollen im Rahmen dieses Praktikums dafür verwendet werden, die *Named Entity Recognition*-Komponenten von *spaCy* zu trainieren. Aus den von der NLM zur Verfügung gestellten Begriffslisten soll eine geeignete Auswahl erfolgen, die für das Training der NER-Komponente verwendet werden kann.

PROBLEM 3: ZUORDNUNG DER ENTITÄTEN Basierend auf den gefundenen Entitäten soll das Python-Programm in der Lage sein, Begriffe wie Krankheiten und Symptome richtig zuzuordnen. Hierzu muss die Struktur von Aussagesätze, Fragen, Aufzählungen und Überschriften analysiert werden und so aufbereitet werden, dass eine automatische Analyse durch NER und *Entity-Linker* möglichst erfolgreich ist. Die Ausgabe der Konsolenapplikation erfolgt im RDF/XML-Format.

1.3 FORSCHUNGSFRAGEN

Dieser Abschnitt beschreibt die Forschungsfragen, die sich aus dem in Abschnitt 1.2 angeführten Problemen ableiten.

FF 1 Wie kann ein medizinischer Text vorverarbeitet werden, so dass relevante Aussagen (Überschriften, Aufzählungen, Leerzeilen und Dialoge) automatisch erkannt werden?

FF 2 Wie kann ein medizinisches Fachvokabular über depressive Erkrankungen automatisiert in eine maschinenlesbare Form überführt werden?

FF 3 Wie kann eine Wissensrepräsentation über einen gegebenen Text maschinenlesbar erstellt werden?

1.4 FORSCHUNGSMETHODE

In dieser Arbeit kommt die in der Forschung zu Informationssystemen etablierte Methode von Nunamaker, Chen und Purdin [**nunamaker_systems_1990-1**] zum Einsatz, die einen strukturierten Rahmen zur Durchführung von Forschungsarbeiten bereitstellt.

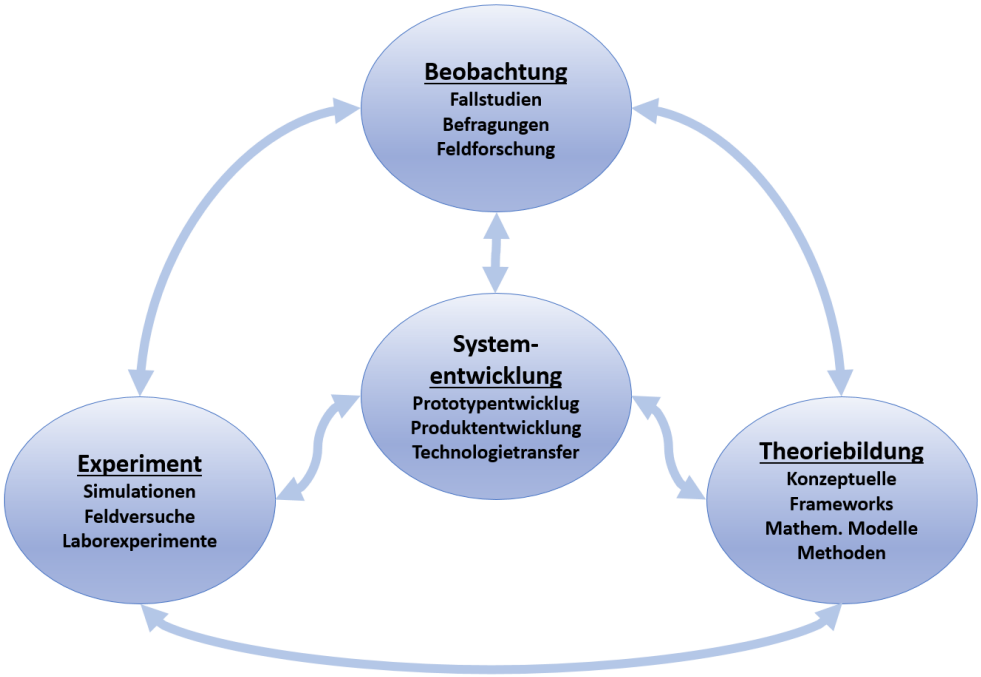

Abbildung 1.1: Forschungsmethode nach [**nunamaker_systems_1990-1**]

Die Forschungsmethode umfasst die folgenden in Abbildung 1.1 dargestellten Phasen:

BEOBACHTUNG: Insbesondere wenn der Untersuchungsgegenstand relativ unbekannt ist, werden Fallstudien, Feldversuche oder Umfragen durchgeführt, um ein „Gefühl" für den Aufwand zu erhalten. Auf dieser Grundlage können dann konkrete Hypothesen erstellt werden, die durch Experimente geprüft werden.

In dieser Arbeit findet aufgrund der Art des Untersuchungsgegenstandes in der Beobachtungsphase hauptsächlich die Literaturrecherche und Ermittlung des Standes von Wissenschaft und Technik statt.

THEORIEBILDUNG: In dieser Phase werden neue Ideen, Konzepte Methoden oder Modelle entwickelt. Diese Theorien beschreiben das System-

verhalten allgemein, haben jedoch wenig praktische Bedeutung für die Zieldomäne. Sie können aber genutzt werden, um Forschungshypothesen aufzustellen, die Planung von Experimenten zu unterstützen und systematische Beobachtungen durchzuführen.

SYSTEMENTWICKLUNG: Diese Phase besteht aus fünf Teilen:

- Konzeptentwurf
- Erstellung der Systemarchitektur
- Erstellung von Prototypen
- Produktentwicklung
- Technologietransfer

EXPERIMENT: In dieser Phase werden die gefundenen Theorien und entwickelten Systeme evaluiert. Die Ergebnisse der Experimente können genutzt werden, um die Theorien zu weiterzuentwickeln und die Systeme zu verbessern.

Obwohl die Phasen in der Methode keine vorgegebene Reihenfolge haben, sondern sich gegenseitig beeinflussen, wird in der vorliegenden Arbeit die oben angeführte Abfolge verwendet.

1.5 FORSCHUNGSZIELE

Dieser Abschnitt beschreibt die Forschungsziele, die sich aus den in Abschnitt 1.3 angeführten Problemen ableiten.

FZ 1 Wie kann ein medizinischer Text vorverarbeitet werden, so dass relevante Aussagen (Überschriften, Aufzählungen, Leerzeilen und Dialoge) automatisch erkannt werden?

- FZ 1.1 Aufbau und Struktur medizinischer Texte (Observation)
- FZ 1.2 Theoriebildung zur Vorverarbeitung medizinischer Texte
- FZ 1.3 Implementierung zur Vorverarbeitung medizinischer Texte
- FZ 1.4 Evaluation zur Vorverarbeitung medizinischer Texte

FZ 2 Wie kann ein medizinisches Fachvokabular über depressive Erkrankungen automatisiert in eine maschinenlesbare Form überführt werden?

- FZ 2.1 Überblick über depressive Erkrankungen (Observation)
- FZ 2.2 Automatisierung durch NLP (Observation)
- FZ 2.3 Theoriebildung zur Überführung medizinischen Fachvokabulars in maschinenlesbare Form zur weiteren Verarbeitung durch NLP
- FZ 2.4 Implementierung zur Überführung medizinischen Fachvokabulars in maschinenlesbare Form zur weiteren Verarbeitung durch NLP

FZ 2.5 Evaluation zur Überführung medizinischen Fachvokabulars in maschinenlesbare Form zur weiteren Verarbeitung durch NLP

FZ 3 Wie kann eine Wissensrepräsentation über einen gegebenen Text maschinenlesbar erstellt werden?

FZ 3.1 Wissensrepräsentation mittels RDF (Observation)

FZ 3.2 Theoriebildung zur Wissensrepräsentation

FZ 3.3 Implementierung zur Wissensrepräsentation

FZ 3.4 Evaluation zur Wissensrepräsentation

1.6 LÖSUNGSANSATZ

Die Wissensrepräsentation medizinischer Texte soll Entitäten unterschiedlicher Kategorie miteinander in Beziehung setzen. Naheliegend ist es z.B., automatisch in einem Text Symptome und Krankheiten zu identifizieren, diese miteinander in Beziehung zu setzen und eine Wissensrepräsentation der Art

loss of appetite - is a symptom of - depression

zu erzeugen. Dazu muss spaCy befähigt werden, nicht nur Fachbegriffe zu erkennen, sondern diese auch richtig zu kategorisieren, als z.B. Krankheit oder Symptom. Folgender Lösungsansatz ist angedacht:

MetaMapLite verwendet eine auf UMLS basierende Datenbasis mit medizinischen Fachbegriffen. Der Umfang der Datenbasis von MetaMapLite ist etwas reduziert und umfasst z.B. nur englische Fachbegriffe. Auf der MetaMapLite-Website kann das zip-Archiv

public_mm_data_lite_usabase_2022aa.zip

heruntergeladen werden. In diesem Archiv befindet sich die Datei *postings* in dem Unterordner

\public_mm_lite\data\ivf \2022AA\USAbase\indices\cuisourceinfo.

Die Datei *postings* enthält ca. 11 Millionen englischsprachige Einträge und ist mit einer Größe von etwa 739 MB etwas handlicher als die Datenbestände von UMLS. Um zu prüfen, ob diese Datenbasis geeignet ist, ist ein Auszug von Einträgen, die für den zu analysierenden Beispieltext über Depressionen relevant sind, in der folgenden Tabelle aufgelistet.

CUI bezeichnet den Concept Unique Identifier, mit dem Synonyme gefunden werden können. Der CUI besteht nur aus dem mit dem Buchstaben 'C' beginnenden Teil und ist hier ergänzt durch einen sogenannten Term-Type. Der Term-Type 'PT' kennzeichnet z.B. bevorzugte Einträge. SUI bezeichnet den String Unique Identifier, mit dem gleichlautende (und damit redundante) Bezeichnungen gefunden werden können.

Die Tabelle listet nur Einträge auf, in denen der Begriff, z.B. 'depression', einer in Klammern hinter dem Begriff stehenden Kategorie zugeordnet ist,

hier z.B. ‘disease’. Über den CUI können in der Datei *postings* Synonyme gefunden werden, wobei nur der mit ‘C’ beginnende Teil relevant ist. Diese Synonyme enthalten häufig keine Kategorien in Klammern. Die Kategorien können aber über den CUI den Synonymen zugeordnet werden. Die sehr zahlreichen Synonyme bzw. alternativen Ausdrücke sind in der Tabelle nicht aufgeführt.

CUI	SUI	Item	Source
FNC0232933	S3225525	Abnormal menstrual cycle (finding)	SNOMEDCT_US
PTC0424569	S3221759	Circumstances interfere with sleep (disorder)	SNOMEDCT_US
YC0009806	S3235964	Constipation (finding)	SNOMEDCT_US
LAC3845528	S14560529	Depressed mood (e.g., feeling sad, tearful)	LNC
SYC0344315	S3252744	Depressed mood (finding)	SNOMEDCT_US
GTC0011570	S1431189	depression (disease)	AOD
SYC0681028	S1431190	depression (economic)	AOD
ETC0021603	S3263624	Disorders of initiating and maintaining sleep (disorder)	SNOMEDCT_US
FNC2939186	S3264511	Disturbance in mood (finding)	SNOMEDCT_US
SYC0349217	S20166983	Episode of depression (finding)	SNOMEDCT_US
FNC1288289	S3312713	Fearful mood (finding)	SNOMEDCT_US
FNC0150041	S3313279	Feeling hopeless (finding)	SNOMEDCT_US
FNC0022107	S3313282	Feeling irritable (finding)	SNOMEDCT_US
FNC0424000	S3313310	Feeling suicidal (finding)	SNOMEDCT_US
ETC0917801	S3373158	Insomnia (disorder)	SNOMEDCT_US
PTC0015672	S3386372	Lack of energy (finding)	SNOMEDCT_US
FNC2981158	S3386389	Lack of libido (finding)	SNOMEDCT_US
FNC1971624	S3397609	Loss of appetite (finding)	SNOMEDCT_US
FNC0178417	S3397620	Loss of capacity for enjoyment (finding)	SNOMEDCT_US
FNC0456814	S3397668	Loss of motivation (finding)	SNOMEDCT_US
FNC0424219	S3397688	Loss of self-esteem (finding)	SNOMEDCT_US
FNC0679136	S3398077	Low self-esteem (finding)	SNOMEDCT_US
PTC5444612	S20749480	mood (physical finding)	MTH
FNC0424566	S3439673	Not getting enough sleep (disorder)	SNOMEDCT_US
FNC2945580	S3485453	Poor self-esteem (finding)	SNOMEDCT_US
FNC0235160	S3513783	Restless sleep (finding)	SNOMEDCT_US
FNC0424570	S3580589	Symptoms interfere with sleep (disorder)	SNOMEDCT_US
PTC0424570	S3580589	Symptoms interfere with sleep (disorder)	SNOMEDCT_US
PTC0233481	S3620195	Worried (finding)	SNOMEDCT_US

In der Tabelle finden sich drei Kategorien, ‘disease’, ‘disorder’ und ‘finding’ (Die *postings*-Datei enthält noch weitere wie z.B. ‘situation’ oder ‘pro-

cedure'). Hier bietet sich an, die Begriffe und Kategorien dieser Datenbasis für als das Trainieren des Entity-Recognizers oder Entity-Rulers von spaCy zu verwenden. Die Zuordnung 'finding' bedeutet eigentlich Befund, kann hier aber auch als Symptom verstanden werden. 'disorder', also Störung, kennzeichnet in den meisten Fällen Krankheiten. Der Begriff 'insomnia' findet sich nicht im zu analysierenden Text, dort ist stattdessen von 'disturbed sleep' die Rede ist. Hier muss versucht werden, über Synonyme eine richtige Kategorisierung zu erreichen. Die meisten Einträge stammen von der Datenbank SNOMED CT (Systematized Nomenclature of Medicine Clinical Terms), die seit 2003 Teil des UMLS Metathesaurus ist.

Der Inhalt der Datei *postings* muss aufbereitet (und ggf. auch gekürzt) werden, so dass die Daten für das Training des Entity-Recognizers verwendet werden können. Es muss untersucht werden, wie basierend auf der Wörterliste geeignete Trainingsdaten erzeugt werden können, etwa durch automatisch erzeugte Beispielsätze. Alternativ kann der Entity-Ruler zum Einsatz kommen, dem die Wörterliste einfach übergeben werden kann und dessen Funktion vorhersagbarer ist als die des Entity-Recognizers.

Der zu analysierende Text muss durch ein Python-Programm aufbereitet werden, mit dem Ziel, dass die NLP-Komponenten der Prozesspipeline des NLP-Frameworks, möglichst erfolgreich arbeiten. Ein Vorgehen könnte beispielsweise sein, Aufzählungen in mehrere vollständige Sätze zu zerlegen, so dass möglichst sinnvolle vollständige Sätze entstehen, die eine Krankheit und ein Symptom enthalten, so dass die automatische Textanalyse nicht überfordert wird, Krankheit und Symptome zusammenzubringen. Hierbei werden die von *spaCy* zur Verfügung gestellten Pipeline-Komponenten wie der Lemmatizer und Parser genutzt.

Es ist zu prüfen, inwieweit der *Dependency Parser* von *spaCy* genutzt werden kann, um Abhängigkeiten und Subjekt-Prädikat-Objekt-Beziehungen in Sätzen zu erkennen, die für die Erstellung des RDF-Graphen genutzt werden können.

Idealerweise entsteht jedoch bereits durch die NER ein Text mit Aussagesätzen, die jeweils eine Krankheit und ein oder mehrere Symptome enthalten. Für die Wissensrepräsentation soll der Entity-Linker eingesetzt werden. Normalerweise wird der Entity-Linker verwendet, um (mehrdeutige) Entitäten, die vom Entity-Recognizer oder Entity-Ruler gefunden werden, einer eindeutigen Entität zuzuordnen (z.B. einem Wikipedia-Eintrag). Der Entity-Linker wird trainiert, so dass die Zuordnung Kontext-basiert erfolgt.

Für die Erstellung der Wissensrepräsentation wird der Entity-Linker auf unorthodoxe Weise verwendet und mit dieser Komponente die Wissensrepräsentation aufgebaut. Dafür ist die Wissensbasis des Entity-Linkers von zentraler Bedeutung. In dieser werden durch das Python-Programm als zusammengehörend erkannte Krankheiten und Symptome gespeichert. Dabei können Symptome mehreren Krankheiten zugeordnet werden. Der Entity-Linker wird mit Sätzen aus den zu analysierenden Texten trainiert, die es ihm erlauben, auch bei anderen Texten Kontext-basiert einem Symptom die richtige Krankheit zuzuordnen. Analysiert man viele Texte hintereinander,

kann der Entity-Linker so mit der Zeit eine umfangreiche Wissensbasis aufbauen. Analysiert man einen Text dann mit dem Entity-Recognizer und danach mit dem Entity-Linker, dann entsteht automatisch die Wissensrepräsentation, wobei etwa die Entität 'lack of energy' vom Entity-Recognizer gefunden wird und als 'Symptom' kategorisiert wird und anschließend vom Entity-Linker der Krankheit 'depression' zugeordnet wird, sofern sich dies aus dem Kontext ergibt.

Arbeitspakete:

- Manuelle Erstellung einer Wissensrepräsentation basierend auf dem zu analysierenden Text.
- Identifizierung eines medizinischen Vokabulars, das zum Training der NER-Komponente verwendet werden kann.
- Untersuchung, auf welche Weise die Trainingsdaten für möglichst gute Ergebnisse der NER-Komponente aufbereitet werden müssen (z.B. durch automatisch erzeugte Beispielsätze). Alternativ Verwendung des Entity-Rulers.
- Entwicklung einer Programm-Komponente, die die zu analysierenden Texte vorverarbeitet, so dass die NLP-Pipeline möglichst effizient arbeitet.
- Entwicklung einer Programm-Komponente, die basierend auf den vom NER gefundenen Entitäten zusammengehörende Entitäten (etwa Symptom und Krankheit) identifiziert und zusammen mit Beispielsätzen aus dem zu analysierenden Text die Wissensbasis aufbaut.
- Entwicklung einer Programm-Komponente, die die Wissensbasis als RDF/XML-Datei ausgibt.

2 STAND DER WISSENSCHAFT

2.1 FZ 1.1. AUFBAU UND STRUKTUR MEDIZINISCHER TEXTE

Medizinische Texte existieren in verschiedenen Formen mit unterschiedlichen Zielsetzungen. Beispielsweise gibt es Lehrbuch- und Enzyklopädietexte, die überblicksweise über medizinische Sachverhalte informieren, wissenschaftliche Studien, in denen neue wissenschaftliche Erkenntnisse vorgestellt werden und außerdem Arztberichte, in denen über individuelle Patienten und deren Krankheitsverlauf informiert wird. Die für diese Arbeit genutzten Analysetexte stellen enzyklopädische Texte dar, in denen insbesondere Symptome und Merkmale psychologischer Erkrankungen beschrieben werden. Charakterisiert sind diese Texte dadurch, dass sie durch Zwischenüberschriften gegliedert sind und neben Fließtext auch Aufzählungen enthalten.

2.2 FZ 2.1. ÜBERBLICK ÜBER DEPRESSIVE ERKRANKUNGEN

Als Quellen für diesen Abschnitt dienten [**mpg_depression**], [**who_depression**] und [**psychrembel_depression**].

2.2.1 *Allgemein*

Depression ist eine psychische Krankheit, die geschätzt 3,8% der Weltbevölkerung und 5% der Erwachsenen betrifft. Sie kann dazu führen, dass betroffene Menschen Schwierigkeiten haben im Arbeits- und Familienleben zurecht zu kommen, und sie führt im schlimmsten Fall zur Selbsttötung.

2.2.2 *Symptome*

Übliche Symptome eine Depression sind eine traurige Grundstimmung, Konzentrationsprobleme, Hoffnungslosigkeit, Müdigkeit und Reizbarkeit. Auch Schlafstörungen, eine Libidostörung, verminderter oder gesteigerter Appetit und ein vermindertes Selbswertgefühl oder Selbstbewusstsein sind Symptome. Im schlimmsten Fall treten Selbsttötungsgedanken auf.

2.2.3 *Formen der Depression*

Die Schwere einer typischen Depression wird durch leichte, mittelgradige und schwere Episoden unterschieden. Dabei sind die Übergänge fließend. Besondere Formen der Depression sind die chronische Depression, bei der trotz therapeutischer Maßnahmen nur wenig Besserung eintritt. Desweiteren

gibt es die manisch-depressive Depression, bei der sich depressive und manische Phasen abwechseln. Außerdem gibt es auch kurze, akute depressive Verstimmungen, die nur zwischen einem Tag und zwei Wochen dauern.

2.2.4 *Ursachen*

Depressionen entstehen durch ein kompexes Zusammenspiel von sozialen, psychologischen und biologischen Faktoren. Dabei wird angenommen, dass für eine typische Depression die Genetik 50Konkret können Kindheitserfahrungen, Verluste und Arbeitslosigkeit eine Depression begünstigen.

2.2.5 *Behandlung*

Je nach Schweregrad der Krankheit werden zur Behandlung von Depressionen psychologische Behandlungen angewandt, und/oder den betroffenen Personen Antidepressiva verschrieben.

2.3 FZ 2.2. AUTOMATISIERUNG DURCH NLP

2.3.1 *Natural Language Processing (NLP)*

Natural Language Processing ist ein Teil der Künstlichen Intelligenz. Er befasst sich mit der Aufgabe Maschinen den Umgang mit der menschlichen Sprache beizubringen, also das Verstehen der Sprache und dem richtigen Antworten darauf. Mithilfe von NLP-Methoden werden zum Beispiel Sätze analysiert, die Bedeutung von Texten erfasst, Chatbots erstellt und sogar ganze Geschichten geschrieben. Die grundlegendsten Schritte, die beim Erarbeiten eines NLP-Modells (oder allgemeiner eines Machine-Learning-Modells) ausgeführt werden, sind folgende:

1. Aufbereitung von Daten
2. Wählen eines geeigneten Algorithmus
3. Trainieren des Modells mit Trainingsdaten
4. Testen des Modells mit Testdaten
5. Modell anwenden / Vorhersagen treffen

2.3.2 *Named Entity Recognition (NER)*

Named Entity Recognition bezeichnet einen Teilbereich des Natural Language Processing, bei dem es darum geht wichtige Entitäten, wie zum Beispiel Personen, Orte oder Institutionen, in einem Text zu erkennen. Methoden zur Bewältigung dieser Aufgabe werden seit circa 30 Jahren entwickelt. Darunter fallen grammatikbasierte und statistische Methoden, sowie auch Methoden des Machine Learning.

2.3.3 *Entity Linking*

Als Entity Linking wird im Bereich des NLP die Aufgabe beschrieben, die den Entitäten (z.B. Personen, Orte) in einem Text das korrekte Äquivalent in einer Wissensbasis zuordnen soll. In [**shen_entity_2021**] wird Entity Linking folgendermaßen definiert (übersetzt):

Definition 2.3.1 (Entity Linking) *Gegeben sei ein Dokument D, welches die erkannten Entitäten $M = \{m_1, m_2, \ldots, m_{|M|}\}$ enthält, sowie eine Ziel-Wissensbasis KB, welches die Entitäten $E = \{e_1, e_2, \ldots, e_{|M|}\}$ enthält. Das Ziel ist es jede Entität m_i in M seinem korrekten Äquivalent e_i in E zuzuordnen.*

2.3.4 *spaCy*

spaCy ist eine Python-Bibliothek, die die erforderlichen Daten und Algorithmen enthält, die zum verarbeiten von Texten mit natürlicher Sprache benötigt werden. SpaCy enthält vortrainierte Modelle für über 70 verschiedene Sprachen, unter anderem Spanisch, Englisch, Griechisch und Deutsch. Zudem ermöglicht spaCy das Einbinden von Modellen, die mithilfe anderer Python-Bibliotheken (zum Beispiel PyTorch oder TensorFlow) trainiert wurden. Anders als einige andere NLP-Bibliotheken konzentriert sich spaCy darauf, Software für den Produktionseinsatz zur Verfügung zu stellen. Erstmals wurde spaCy 2015 von Matthew Honnibal veröffentlicht. [**vasiliev2020natural**] [**github_spacy**] [**spacy**]

2.3.5 *Weitere Python-Bibliotheken für Machine Learning (ML) und NER*

Eine der wichtigsten Python-Bibliotheken für NLP ist NLTK (Natural Language Toolkit) [**bird2006nltk**]. Weitere Python-Bibliotheken, die für NLP-Aufgaben genutzt werden können, sind unter Anderem Gensim [**vrehuuvrek2011gensim**], Pattern [**de2012pattern**][**github_pattern**], scikit-learn und PyTorch.

2.4 FZ 3.1. WISSENSREPRÄSENTATION MITTELS RDF

Das *Resource Description Framework* (RDF) [**w3c_all_2022**] ist ein Framework zur Darstellung von Informationen im Semantischen Web, das von der *RDF Working Group* des *World Wide Web Consortium* (W3C) erstellt wurde. Das RDF-Modell besteht aus einem Datenmodell, mit dem Aussagen über Ressourcen in Form eines Graphen dargestellt werden. Die Informationen werden als Tripel von Subjekt, Prädikat und Objekt gespeichert und ermöglichen auf diese Weise eine maschinenlesbare Bereitstellung semantischer Informationen.

Die Subjekte und Objekte sind dabei die Knoten des Graphen und die Prädikate die Kanten zwischen diesen Knoten.

Abbildung 2.1 zeigt einen einfachen RDF-Graphen, in dem das Subjekt „depression“ mit dem Prädikat „hasSymptom“ mit zwei Symptomen ver-

bunden und über das Prädikat „isA“ mit dem Objekt „mentalDisorder“ verbunden ist.

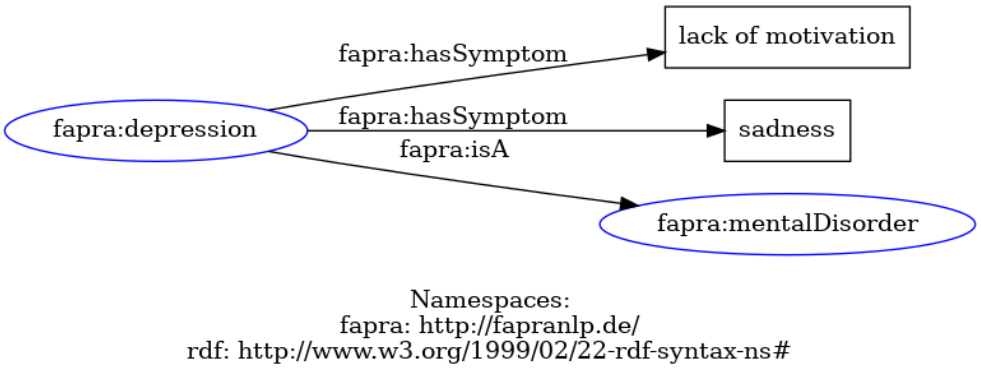

Abbildung 2.1: einfacher RDF-Graph (visualisiert mit `https://www.ldf.fi/service/rdf-grapher`

2.4.1 *Resource Description Framework Schema*

Zusätzlich zu RDF stellt das W3C auch eine Empfehlung für ein Datenmodellierungsvokabular für RDF-Daten bereit: das RDF-Schema. Dies stellt eine Erweiterung des grundlegenden RDF-Vokabulars dar. ... [TODO]

2.4.2 *Dublin Core Metadata Initiative*

DCMI ... [TODO]

2.4.3 *Serialisierung von RDF-Graphen*

Für die Serialisierung von RDF-Graphen stehen mehrere Formate zur Verfügung. So wird der MeSH-Datensatz im *N-Triples*-Format bereitgestellt und in diesem Praktikum erfolgt die Ausgabe der Wissensrepräsentation im RDF/XML-Format.

Ein Beispiel für die Serialisierung des in Abbildung 2.1 dargestellten Graphen im RDF/XML-Format ist in Listing 2.1 angeführt.

Listing 2.1: RDF/XML

```
<?xml version="1.0" encoding="utf-8"?>
<rdf:RDF
  xmlns:fapra="http://fapranlp.de/"
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
>
  <rdf:Description rdf:about="http://fapranlp.de/depression">
    <fapra:hasSymptom>lack of motivation</fapra:hasSymptom>
    <fapra:hasSymptom>sadness</fapra:hasSymptom>
    <fapra:isA rdf:resource="http://fapranlp.de/mentalDisorder"/>
```

```
  </rdf:Description>
</rdf:RDF>
```

Für Python steht mit RDFLib [**rdflib_team_rdflib_2022**] eine Bibliothek zur Arbeit mit RDF-Graphen bereit.

2.4.4 *SPARQL Protocol and RDF Query Language*

SPARQL ... [TODO]

3 PLANUNG

In diesem Kapitel wird die vorläufige Gliederung und Zeitplanung des Fachpraktikums vorgestellt.

3.1 GLIEDERUNG

1. Einleitung
   - 1.1 Motivation
   - 1.2 Problembeschreibung
   - 1.3 Forschungsfragen
   - 1.4 Forschungsmethode
   - 1.5 Forschungsziele
2. Stand der Wissenschaft
   - 2.1 FZ 1.1 Aufbau und Struktur medizinischer Texte
   - 2.2 FZ 2.1 Überblick über depressive Erkrankungen
   - 2.3 FZ 2.2 Automatisierung durch NLP
   - 2.4 FZ 3.1 Wissensrepräsentation mittels RDF
3. Theoriebildung
   - 3.1 FZ 1.2 Theoriebildung zur Vorverarbeitung medizinischer Texte
   - 3.2 FZ 2.3 Theoriebildung zur Überführung medizinischen Fachvokabulars in maschinenlesbare Form zur weiteren Verarbeitung durch NLP
   - 3.3 FZ 3.2 Theoriebildung zur Wissensrepräsentation
4. Implementierung
   - 4.1 FZ 1.3 Implementierung zur Vorverarbeitung medizinischer Texte
   - 4.2 FZ 2.4 Implementierung zur Überführung medizinischen Fachvokabulars in maschinenlesbare Form zur weiteren Verarbeitung durch NLP
   - 4.3 FZ 3.3 Implementierung zur Wissensrepräsentation
5. Evaluation
   - 5.1 FZ 1.4 Evaluation zur Vorverarbeitung medizinischer Texte

5.2 FZ 2.5 Evaluation zur Überführung medizinischen Fachvokabulars in maschinenlesbare Form zur weiteren Verarbeitung durch NLP

5.3 FZ 3.4 Evaluation zur Wissensrepräsentation

5.4 Zusammenfassung

6. Zusammenfassung und Diskussion

6.1 Ergebnisse

6.2 Offene Fragen

- Anhang
- Literatur

3.2 ARBEITS- UND ZEITPLAN

	Phase	Beginn	Ende
1	Gesamtprojekt	15.11.22	28.03.23
2	PE3 Prototypische Softwareentwicklung	15.11.22	16.12.22
3	Meilenstein 1	15.11.22	16.12.22
4	Modellierung und Theoriebildung	15.11.22	16.12.22
5	Einleitung	15.11.22	16.12.22
6	Stand der Technik	15.11.22	16.12.22
7	FF1.1, 1.2, 2.1, 2.2, 2.3, 3.1, 3.2	15.11.22	16.12.22
8	Meilenstein 2	17.12.22	17.02.23
9	Systementwicklung	17.12.22	17.02.23
10	Implementation	17.12.22	17.02.23
11	FF1.3, 2.4, 3.3	17.12.22	17.02.23
12	Meilenstein 3	18.02.23	20.03.23
13	Experiment	18.02.23	20.03.23
14	Evaluation	18.02.23	20.03.23
15	FF1.4, 2.5, 3.4	18.02.23	20.03.23
16	PE4, Präsentation der Ergebnisse	21.03.23	28.03.23
17	Präsentation	21.03.23	25.03.23
18	Kolloquium	25.03.23	28.03.23

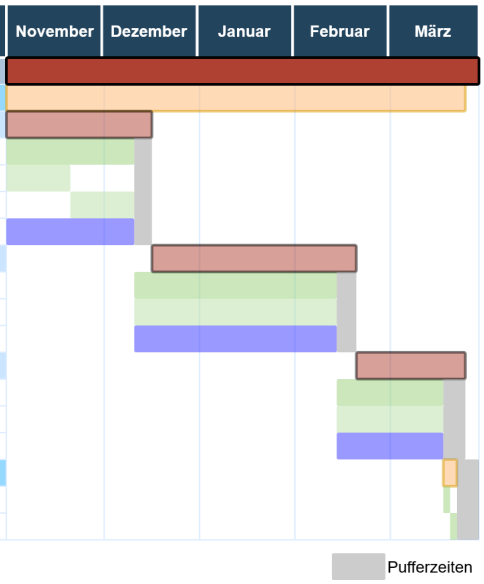

Abbildung 3.1: Arbeits- und Zeitplan