Universität Leipzig
Fakultät für Mathematik und Informatik
Institut für Informatik

Bachelorarbeit

Das Netzwerk des Widerstands

Jonas Richter

14. März 2022

Betreut von Dr. Andreas Niekler, Dr. Christian Kahmann

Inhaltsverzeichnis

Abkürzungsverzeichnis

GDW Gedenkstätte Deutscher Widerstand

GND Gemeinsame Normdatei

IE Information Extraction

NDW Netzwerk des Widerstands

NE Named Entity

NEL Named Entity Linking

NER Named Entity Recognition

NLP Natural Language Processing

POS Part of Speech

STTS Stuttgart Tübingen Tagset

Vorwort

Nachdem die Nationalsozialisten 1933 in Deutschland an die Macht gekommen waren, entwickelten sie schnell einen brutalen und sehr effizienten Verfolgungsapparat, um den Widerstand gegen sie zu brechen. Wer sich aktiv gegen den Nationalsozialismus wandte, musste mit massivster Repression rechnen. Und dennoch sind uns heute Menschen bekannt, die mit verschiedensten Mitteln und Motivationen versuchten, dieses verbrecherische Regime zu schwächen. Vom Verstecken von Verfolgten, über die Verbreitung von oppositionellen Schriften, wie die Weiße Rose, bis hin zur versuchten Tötung der NSDAP-Führung, wie Georg Elser. Das Aktionsspektrum der Widerständigen war breit gefächert. Jede Widerstandsaktion erforderte aber den größten Mut, denn wie die Schicksale von Georg Elser und den Mitgliedern der Weißen Rose zeigen, kostete der Widerstand gegen die Nationalsozialisten häufig das Leben. Und auch wenn nicht, drohte oft Folter, Zwangsarbeit und Haft im Konzentrationslager. Heute sind die Zustände in Deutschland andere und trotz und vielleicht gerade deshalb darf der Mut und die Überzeugung dieser Menschen nicht in Vergessenheit geraten. In dieser Arbeit soll das *Netzwerk des Widerstands (NDW)* vorgestellt werden, das in der Lage ist, den Aufbau des Widerstandsnetzwerks aus Texten zu extrahieren.

Zur besseren Lesbarkeit wurden in dieser Arbeit teilweise Schreibweisen für bestimmte Objekte verwendet. Projekte und Anwendungen sind *kursiv* geschrieben, Programmauszüge wie Klassen, Aufrufe oder Dateien sind dagegen `unformatiert`.

1 Einleitung

1.1 Motivation

Die Analyse von sozialen Netzwerken hat sich in den letzten Jahren zu einer verbreiteten Methode in den Sozialwissenschaften entwickelt und ist inzwischen auch in den Populärwissenschaften angekommen[1]. Doch beziehen sich solche Analysen häufig auf Netzwerke in sozialen Medien und eher selten auf soziale Strukturen in der analogen Welt. Der Grund dafür ist vermutlich, dass die relevanten Daten in diesem Bereich viel schwerer erhoben werden können als im Digitalen, wo eine einfache Auswertung von Follower- oder Freundschaftslisten ausreicht. Wie aufwändig und unvollständig die Modellierung von aktuell bestehenden analogen sozialen Netzwerken sein kann, zeigte Stanley Milgram schon 1969 mit seinem Small-World-Experiment [TM69]. Deutlich erfolgversprechender scheint da die Modellierung von vergangenen analogen sozialen Netzwerken. Mit dieser Thematik setzt sich die noch relativ junge Historical Network Research Community[2] auseinander, die 2009 aus einer Workshopreihe zum gleichen Thema hervorging und seit 2017 ein eigenes Journal veröffentlicht.

Ein großer Vorteil der Netzwerkforschung ist, dass sich der Forschungsgegenstand sehr anschaulich und übersichtlich durch Graphstrukturen visualisieren lässt. So kann komplexes Wissen schnell greifbar gemacht werden, was solche Netzwerke auch für historische Ausstellungen nutzbar machen kann.

Diese Arbeit beschäftigt sich, wie oben erwähnt, also mit dem sozialen NDW gegen den Nationalsozialismus. Als digitaler Wissensgraph soll es schnell Informationen über Widerstandskämpfer:innen, Widerstandsgruppen, Orte und Daten des Widerstands vermitteln. Diese Informationen sollen zuvor mithilfe von natürlicher Sprachverarbeitung automatisch aus Texten über Widerstandskämpfer:innen extrahiert werden. Dabei sollen vor allem die Beziehungen zwischen einzelnen Aktivist:innen, Gruppen, Orten und Daten

[1] vlg. z.B.: YouTube-Video von Ultralativ „Ich habe ganz YouTube Deutschland ausgewertet und analysiert“ https://youtu.be/jXb-zSPjhPI (abgerufen 24.01.2022)

[2] Webseite der Historical Network Research Community https://historicalnetworkresearch.org/ (14.02.2022)

im Fokus stehen.

Das Augenmerk dieser Arbeit liegt also insbesondere im Aufbau einer Pipeline zur Extraktion von Informationen und Strukturierung derselben in einem Wissensgraphen. Die Domäne des Netzwerks des Widerstands gegen den Nationalsozialismus dient hierbei vor allem als Arbeitsbeispiel und soll mehr den Nutzen dieser Pipeline hervorheben als tatsächlich neue Erkenntnisse in diesem Bereich zu erarbeiten.

1.2 Stand der Forschung

Es gibt eine Reihe verwandter Arbeiten zu dieser Arbeit. Dabei lässt sich feststellen, dass einige der Arbeiten eher in technologischer Hinsicht verwandt sind und andere eher in thematischer Sicht.

Beispielhaft für den ersten Fall ist das FashionBrain Projekt [Che+17], das sich mit der Informationsextraktion in der Domäne Mode beschäftigt und explizit die Position der europäischen Modeindustrie stärken soll. In diesem Projekt werden Methoden wie Named Entity Recognition (NER), Relation Extraction und Named Entity Linking (NEL) behandelt, die auch der vorliegenden Arbeit eine zentrale Rolle spielen. Im Zuge des Projekts ist außerdem das Framework zur natürlichen Sprachverarbeitung *flair* entstanden [Akb+19], das sich als eines der wichtigsten Tools für diese Arbeit herausgestellt hat. Insofern ist das Projekt FashionBrain und insbesondere das Framework *flair* also nicht nur eng verwandt mit dieser Arbeit, sondern stellt auch eine wichtige Grundlage dar.

Eine wichtige theoretische Grundlage liefert die Dissertation von Christoph Benedikt Alt, in der die wichtigsten Methoden der natürlichen Sprachverarbeitung anschaulich dargestellt werden [Alt21]. Vor allem der Bereich der Information Extraction (IE) und insbesondere die Methode Relation Extraction werden dort ausführlich besprochen. An dieser Stelle sei auch auf die deutlich erkennbaren Parallelen zwischen der Architektur dieser Arbeit, die in Abschnitt 2.1 beschrieben wird, und dem Aufbau[3] und Abb. 2.2 eines typischen IE-Systems nach Alt hingewiesen.

Besonders eng verwandt zu diesem Projekt ist das *Sonar-Projekt*[4], das sich zum Ziel gesetzt hat, eine Forschungsumgebung zur historischen Netzwerkanalyse zu entwickeln. Wie aus einem Gespräch mit dem Ansprechpartner des Projekts Gerhard Müller hervorging, hat sich das Projekt mit dem gesamten Prozess der historischen Netzwerkforschung von der Datensammlung bis zur Netzwerkanalyse beschäftigt. Unterschiedliche Forschungs-

[3] vlg. hierzu Abb. 1.1

[4] Projekt-Webseite: https://sonar.fh-potsdam.de/ (20.01.2022)

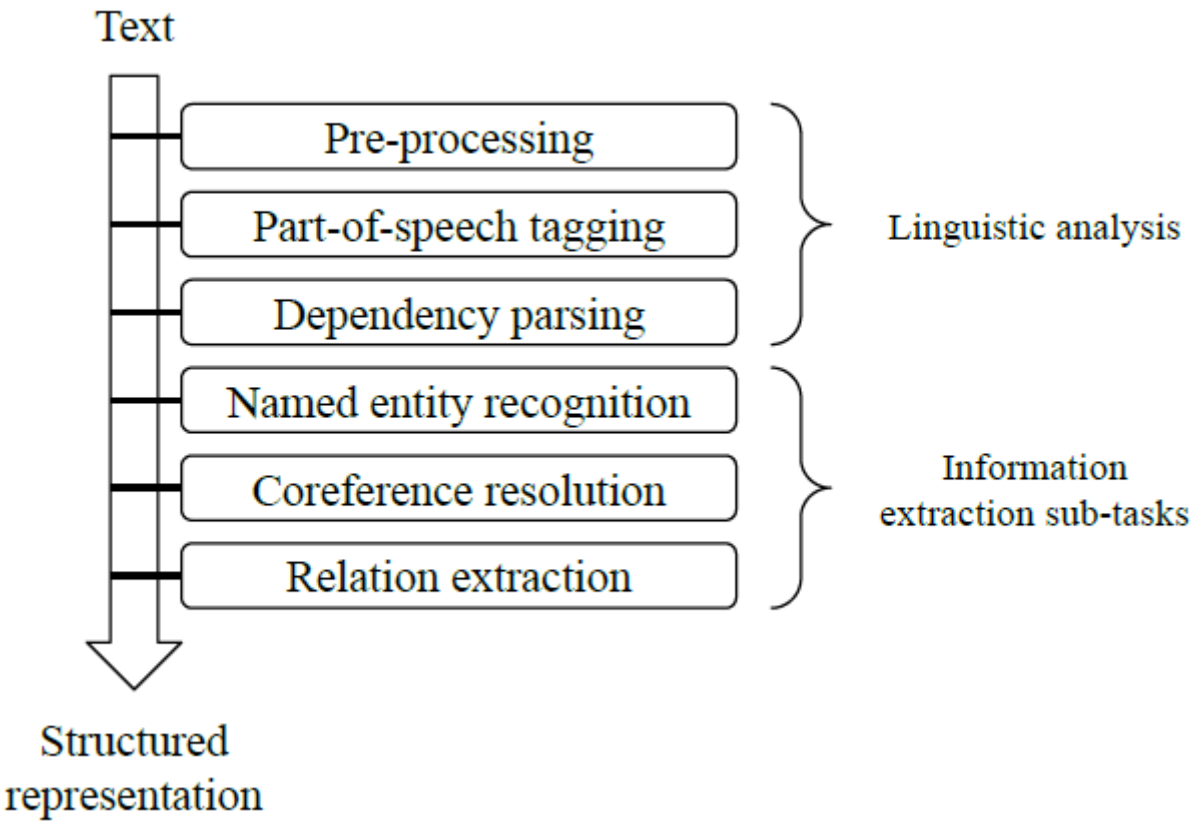

Abbildung 1.1: Darstellung einer typischen IE Pipeline entnommen aus [Alt21, S.19]

gruppen haben dabei verschiedene Aspekte dieses Prozesses untersucht und versucht, bestehende Lücken zu schließen. Auch IE und insbesondere NER und NEL wurden dort in einem Proof-of-Concept behandelt [Men+21]. Interessant ist dabei auch die Auswahl der Entitäten. Genau wie in der vorliegenden Arbeit stehen Personen, Organisationen und Orte im Fokus des NER-Prozesses [Lab+19]. Zusätzlich nutzt *SoNAR* die in der Gemeinsame Normdatei (GND) verwendeten Entitätentypen Konferenzen, Sachbegriffe und Werke. Die vorgeschlagene Nutzeroberfläche[5] von *SoNAR* liefert einige Inspirationen für die Visualisierung des Wissensgraphen. Die größten Unterschiede zwischen dieser Arbeit und dem *SoNAR*-Projekt bestehen daher vermutlich in der Auswahl des Zeitfensters sowie im Ansatz der Relation Extraction.

Hier werden ausschließliche die Jahrzehnte vor, während und nach der nationalsozialistischen Herrschaft betrachtet, mit einem besonderen Fokus auf die Jahre 1933–1945. Da *SoNAR* auf der GND aufbaut, kann dort ein wesentlich größerer Zeitraum umfasst werden. Mindestens die Jahrhunderte zwischen 1500 und 2000 werden abgedeckt. Währenddessen arbeitet die erwähnte NER-Studie mit mehr als 2 Millionen gescannten Zeitungsseiten aus den Jahren 1856–1940 [Men+21, vgl. S. 239].

Wie oben bereits erwähnt, unterscheidet sich der in *SoNAR* gewählte Ansatz zur Relation Extraction bedeutend von dieser Arbeit. Dort werden mögliche Relationen ausschließlich aus den Verlinkungen der GND extrahiert, was einer Auswertung bereits codierter Metadaten entspricht. Das heißt konkret, dass mögliche Relationen, die nie händisch herausgearbeitet wurden und damit nicht in den Metadaten zu finden sind, auch nicht berücksichtigt werden können. Dieser Fakt spricht zwar nicht gegen das *SoNAR*-Projekt, da es sich bei vielen Werken in der GND um Belletristik und nicht um Sachliteratur handelt. Dennoch ist der in dieser Arbeit gewählte Ansatz deutlich stärker auf die innerhalb der Texte versteckten Relationen fokussiert, wie noch in Abschnitt 2.2.1 beschrieben wird. Das hat den Vorteil, dass auch Informationen gefunden werden, die noch nicht herausgearbeitet und in den Metadaten codiert sind.

Weiterhin ist hier die Historical Network Research Community zu nennen, die sich mit der Erforschung und Analyse von sozialen Netzwerken im historischen Kontext beschäftigt. Dort sind viele Publikationen von verschiedenen Autoren zu diesem Thema zu finden. Auch zum Thema Widerstand gegen den Nationalsozialismus sind in diesem Zusammenhang mehrere Arbeiten erschienen. Beispielhaft sei hier die Promotionsarbeit von Linda von Keyserlingk-Rehbein erwähnt [Key18]. Diese beschäftigt sich mit dem Netz-

[5] Demovideo der *SoNAR*-UI https://sonar.fh-potsdam.de/assets/videos/sonar_prototype-demo_komp.mp4 (Zugriff am 20.01.22)

werk der Widerstandsgruppe vom 20. Juli 1944 aus der Perspektive der Verfolger und ist damit thematisch eng mit der vorliegenden Arbeit verwandt. Von Keyserlingk-Rehbein stellt in ihrer Arbeit mehrere Graphen des Netzwerks vor. Dabei hat sie die Informationen eigenhändig aus vielen verschiedenen historischen Quellen, wie zum Beispiel den Kaltenbrunner-Berichten, extrahiert [Key18, S. 73]. Deshalb besteht die Vermutung, dass das entstandene Netzwerk deutlich kleiner sein wird als das, welches in der hier vorliegenden Arbeit entstanden ist. Dieser quantitative Nachteil kann aber wahrscheinlich durch eine deutlich höhere Qualität der extrahierten Informationen ausgeglichen werden. Um die Qualität der Informationsextraktion zu bewerten, werden in Abschnitt 3.1 die Netzwerke von Keyserlingk-Rehbein mit dem hier entstandenen Wissensgraphen verglichen.

1.3 Forschungsinteresse

Diese Arbeit ist besonders für Historiker:innen interessant, was sich in der oben erwähnten Promotion zeigt. Dort wird die starke Eingrenzung auf einige wenige Quellen folgendermaßen begründet:

> „Eine komplette Analyse des komplexen Netzwerks des 20. Juli 1944 auf Grundlage aller vorhandener Quellen ist von keiner einzelnen Person zu leisten.“ (*[Key18, S. 73]*)

Diese Aussage trifft nicht nur auf das Netzwerk des 20. Juli 1944, sondern nach Aussage von Linda von Keyserlingk-Rehbein auf viele historische soziale Netzwerke zu. Mithilfe eines Systems, das Informationen aus Texten automatisch in einen Wissensgraphen überführen kann, könnte hier also die Anwendung von Netzwerkanalysen als historische Methode gestärkt werden.

Auch ein Gespräch mit den Historikern Uwe Fuhrmann und Henning Fischer, die zur Geschichte der Arbeiterbewegung und der Lagergemeinschaft der Überlebenden des Konzentrationslagers Ravensbrück forschen, hat das Forschungsinteresse dieser Arbeit deutlich gemacht. Sie sehen den Wissensgraphen vor allem als potenzielles Forschungswerkzeug, mit dem leicht und schnell ein Überblick zu dem Thema geschaffen und neue Forschungsperspektiven eröffnet werden können. Die Nutzung in Ausstellungen sei außerdem als „didaktische Spielerei“ vorstellbar. Zusätzlich scheint ein recht großes Interesse an der automatischen Informationsextraktion zu bestehen, die diese Arbeit mit einer großen Zahl von Textquellen erheblich erleichtern könnte.

Weiterhin zeigt sich auch an der öffentlichen Förderung des *SoNAR*-Projekts, dass durchaus ein öffentliches Interesse an der Historischen Netzwerkforschung besteht. Und

besonders der Fakt, dass es mit *Qurator*[6] bereits ein vom BMBF gefördertes Nachfolgeprojekt gibt, verdeutlicht, wie groß das Interesse an diesem Forschungsbereich ist.

Insgesamt lässt sich also ein Interesse an dieser Arbeit nicht nur von Forscher:innen erkennen und es zeichnen sich bereits erste Ideen für mögliche Anwendungen des Wissensgraphen ab.

1.4 Vorangegangene Arbeiten

Diese Arbeit ist nicht die erste, die sich mit der Thematik des Netzwerks des Widerstands beschäftigt. Die grundlegende Formulierung dieser Idee stammt aus einer Hausarbeit von 2018, in der über das Gedenken an die Anarchosyndikalistin und Widerstandskämpferin Anna Götze geschrieben wurde [Ric18]. Weitere konzeptionelle Überlegungen sind in einer Projektarbeit ebenfalls im Jahr 2018 entstanden [RL18]. Im Zuge des Text-Mining-Praktikums ist dann ein erster Prototyp des *NDW* entstanden [ABR21]. Dieser Prototyp soll hier kurz beschrieben werden und dabei vor allem auf die Probleme eingegangen werden, die in der vorliegenden Arbeit behoben werden sollen.

Prototyp

Der Prototyp besteht aus vier Submodulen: crawling, mining, graph und app, die jeweils die Aufgaben Crawling und Strukturierung der gegebenen Onlineressource, Verarbeitung der gefundenen Texte, Übertragung in eine entsprechende Graphstruktur und Darstellung als Webanwendung erledigen. Zur Sprachverarbeitung wird hier die Python-Bibliothek *Stanza* genutzt und alle Schritte ihrer Sprachverarbeitungspipeline durchlaufen. Von besonderer Relevanz sind hierbei aber die Named-Entity-Recognition und die Dependenzanalyse. Während die erste Methode entsprechende Entitäten findet, wird mithilfe der Dependenzgrammatik die Relationsextraktion durchgeführt. Anschließend werden die Ergebnisse mithilfe von *NetworkX* – einer Python-Bibliothek, die explizit auf die Arbeit mit komplexen Netzwerken ausgelegt ist – in eine Graphstruktur überführt und mit einer *Flask*-Webanwendung visualisiert.

Der Prototyp hat aber noch einige Mängel. So wurden unter anderem keine Maßnahmen für das NEL getroffen, was dazu führt, dass im entstehenden Netzwerk verschiedene Knoten für verschiedene Schreibweisen ein und derselben Person auftauchen. Beispielsweise tauchen in diesem Netzwerk die Knoten «Adolf Hitler», «Hitlers» und «Hitler» auf, die jeweils Kanten zu verschiedenen Knoten haben.

[6] https://qurator.ai/ (abgerufen am 25.01.2022)

Außerdem wurden in diesem Graphen auch Substantive als Knoten modelliert, bei denen es sich weder um Orte noch um Organisationen oder Personen handelte, sondern stattdessen zusätzlichen Kontext zu Relationen oder einzelnen Entitäten lieferten. So wurden aus dem ersten Satz der Kurzbiografie von Wolfgang Abendroth:

> „Wolfgang Abendroth wird 1906 in Elberfeld geboren und wächst in einer sozialdemokratischen Lehrerfamilie auf.“

die in der Abbildung 1.2 markierten Knoten und Relationen extrahiert. Zum einen konnten so mehr Informationen insbesondere für einzelne Entitäten codiert werden, zum anderen blähten diese zusätzlichen Informationen den Graphen insgesamt übermäßig auf. Zudem ist fraglich, ob dieser zusätzliche Kontext für die historische Netzwerkforschung überhaupt interessant ist, vor allem wenn man Keyserlingk-Rehbein berücksichtigt:

> „Bei Netzwerkanalysen stehen in der Regel weniger die Akteursmerkmale als vielmehr die Beziehungen zwischen den Akteuren im Mittelpunkt der Betrachtung.“ (*[Key18, S.53]*)

Wie in der Abbildung 1.2 auch zu sehen ist, wurden die Zeitangaben nicht direkt der Relation zugeordnet, sondern als einzelne Knoten modelliert. Das hat eine weitere Vergrößerung der Knotenzahl und damit eine Verschlechterung der Übersichtlichkeit zur Folge.

Ein weiteres Problem stellen gleichnamige Knoten dar. So gibt es im Datensatz zwei verschiedene Personen mit dem Namen Kurt Schumacher, die während der Verarbeitung zu einem Knoten zusammengefasst werden.

Weiterhin ist auch festzustellen, dass der Prototyp kaum vorgeplant wurde, dadurch ist der Quellcode teilweise sehr unübersichtlich und nur schwer zu überarbeiten.

1.5 Anforderungen

Im Zuge dieser Arbeit soll am Beispiel des Netzwerks gegen den Nationalsozialismus eine Pipeline entstehen, die unstrukturierte Informationen aus Texten durch Methoden der natürlichen Sprachverarbeitung extrahiert und dieses Wissen in eine Graphstruktur überträgt. Im Fokus stehen dabei vor allem einzelne Widerstandskämpfer:innen, Gruppen, Orte und ihre Beziehungen untereinander. Wenn immer möglich, soll außerdem die Zeitangabe einer Relation extrahiert werden, um eine gerade für Historiker:innen sehr relevante zeitliche Einordnung zu ermöglichen. Weiterhin soll der Wissensgraph zwar die wichtigsten Informationen zum Widerstandsnetzwerk kompakt zusammenfassen, gleichzeitig aber Informationen, die keine Erkenntnisse zum Netzwerk an sich liefern, mit dem Ziel die mögliche Visualisierung so übersichtlich wie möglich zu halten, aussparen.

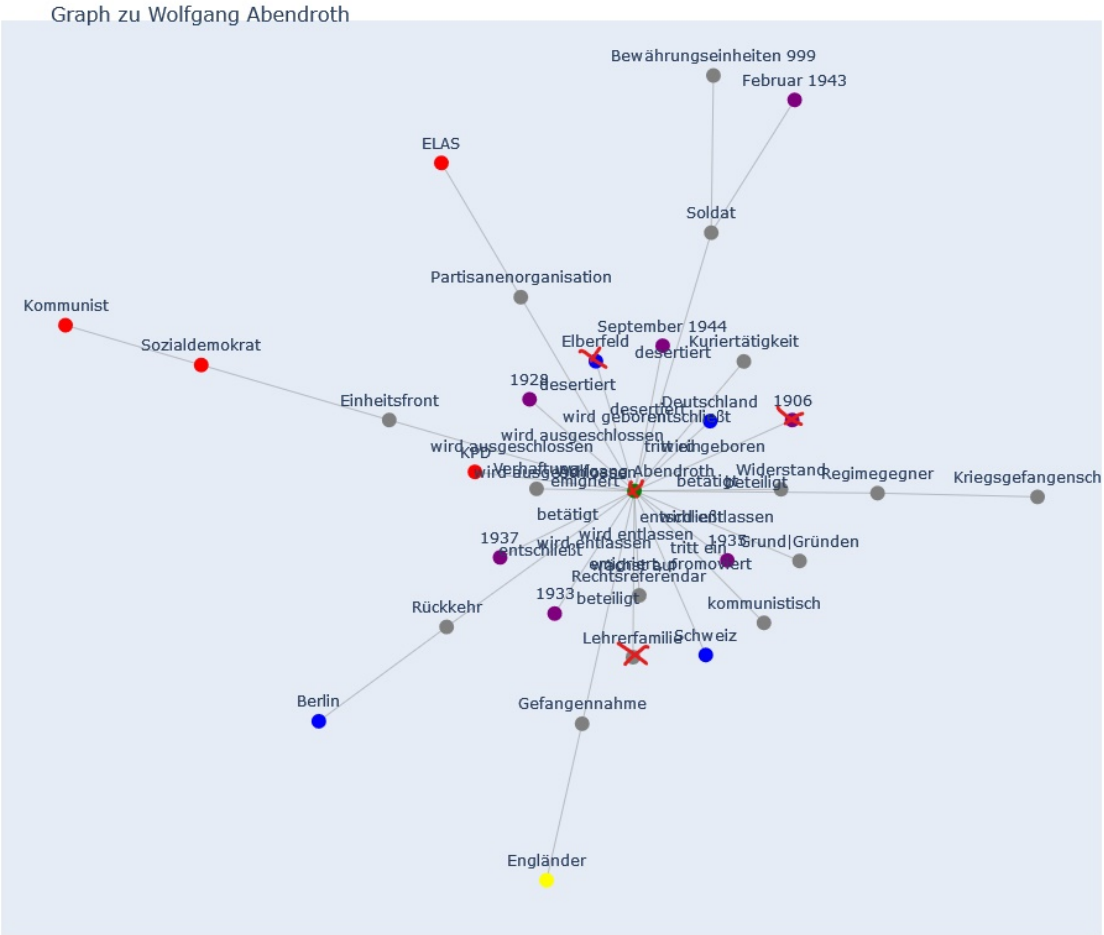

Abbildung 1.2: Resultierender Teilgraph von Wolfgang Abendroth aus dem Prototyp-System. Mit roten Kreuzen sind die angesprochenen Entitäten markiert. (Person: grün, Ort: blau, Organisation: rot, Datum: lila, Sonstige: gelb, Nichtentitäten: grau)

2 Methoden und Umsetzung

Im folgenden Kapitel sollen die theoretischen Hintergründe der Arbeit betrachtet werden. Insbesondere sollen die verfolgten Ansätze aus dem Bereich der natürlichen Sprachverarbeitung eingehender betrachtet werden, aber auch die Planung der Pipeline soll hier besprochen werden. Dazu werden im nachfolgenden Kapitel die Arbeitsansätze, in der Reihenfolge, in der sie auch die IE-Pipeline durchlaufen, vorgestellt. Es sei außerdem an dieser Stelle betont, dass das Hauptaugenmerk in dieser Arbeit auf dem Aufbau dieser Pipeline liegt. Jegliche Analysen und Interpretationen des Netzwerkes sollten stets mit dem entsprechenden Vorwissen oder Kontext stattfinden.

2.1 Planung

2.1.1 Datenstrukturierung

Grundsätzlich sollen alle extrahierten Informationen in einem *NetworkX*-Graphen gespeichert werden. Die Entscheidung für *NetworkX* basiert dabei vor allem auf den guten Erfahrungen mit der Ressource im Prototyp. Entitäten sollen dabei als Knoten und Relationen als Kanten behandelt werden. Die Bibliothek *NetworkX* bietet außerdem die Möglichkeit, jeweils zusätzliche Informationen zu diesen Bausteinen als Kanten bzw. Knotenattribute zu speichern. Zuvor werden die extrahierten Informationen aber bereits, wie nachfolgend beschrieben, mit selbstdefinierten Objekten vorstrukturiert.

UML-Diagramm der Entitäten

Ein wichtiger Vorteil der Graphbibliothek *NetworkX*, der im Prototypen kaum ausgenutzt wurde, ist die Möglichkeit, auch komplexe hashbare Datentypen als Knoten zu speichern [HSS08]. Das heißt also, dass durch das Nutzen von selbstdefinierten Datentypen deutlich mehr Informationen effizienter in dem Graphen gespeichert werden können als mit einem einfachen String. Daher ist es naheliegend, vor der Überführung in eine Graphstruktur das extrahierte Wissen durch die Anwendung von Python-Klassen bereits zu strukturieren.

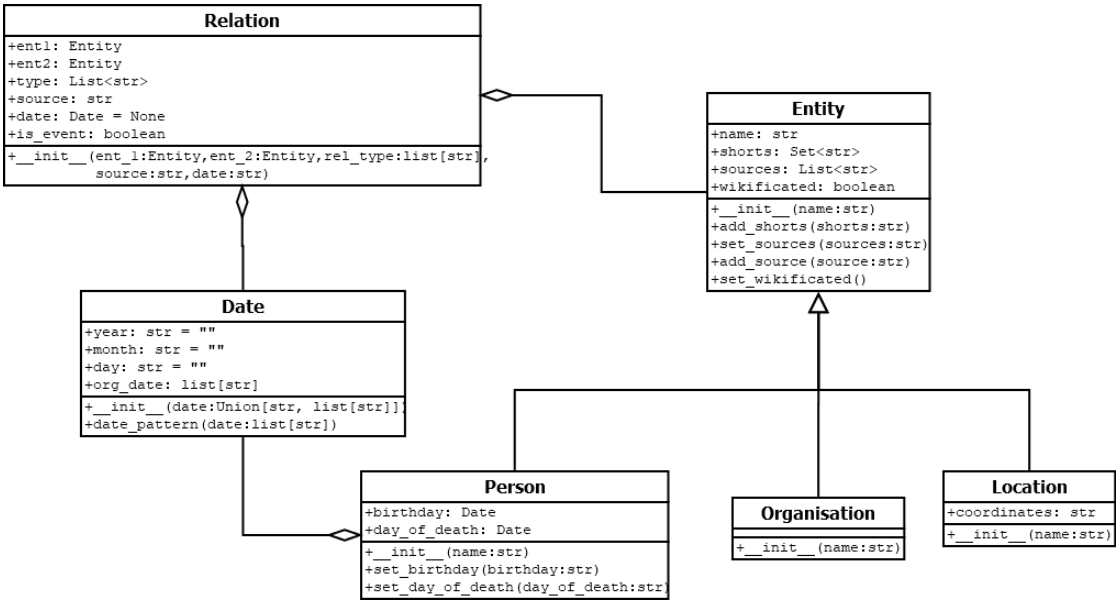

Abbildung 2.1: UML-Diagramm der strukturierten Daten

Das in Abbildung 2.1 gezeigte UML-Diagramm zeigt, wie diese Vorstrukturierung hier vorgenommen wurde. Neben der Superklasse Entity, die in diesem Projekt immer nur als eine der drei Kindklassen aufgerufen wird, wird auch jede Relation als Objekt gespeichert. Das macht vor allem das Lesen- und Schreiben der Zwischenausgabedatei `rel_ent_collection` einfacher und kann aber in dieser Form nicht als Kante in *NetworkX* genutzt werden. Trotzdem ermöglicht die Relationenklasse ein effizientes Überführen der entsprechenden Klassenattribute in *networx*-Kantenattribute. Zurzeit haben einige Attribute der Entitäten - insbesondere die der Kindklassen - noch keine Anwendung gefunden und sollen mehr einen Ausblick geben, welche Wissensabfragen und Visualisierungen in späteren Versionen des *NDW* möglich sein können.

An dieser Stelle sei außerdem erwähnt, dass ursprünglich die Datumsangaben als `date`-Datentyp gespeichert werden sollten. Dort können aber nur vollständige Datumsangaben verarbeitet werden. Unvollständige Datumsangaben wie zum Beispiel „1906“ würden so automatisch als 01.01.1906 interpretiert werden.

2.1.2 Architektur

Während der Arbeit an diesem Projekt lag stets ein Schwerpunkt auf einer hohen Modularität und der Trennung von nicht-verwandten Aufgaben. Ähnliche Aufgaben werden in

gemeinsamen Modulen verarbeitet. Beispielhaft sei hier das `i_o.py`-Modul erwähnt, das fast alle benötigten Schreib- und Leseoperationen durchführt. Ziel dieser angestrebten Modularität ist, dass in Zukunft ohne Probleme Anpassungen und Verbesserungen in bestimmten Bereichen vorgenommen werden können, ohne dass das ganze Projekt neu aufgesetzt werden muss.

Bevor hier der Ablauf der Pipeline beschrieben wird, sei noch auf die Struktur der Eingabedatei eingegangen. Es handelt sich hierbei um eine JSON-Datei, die eine Liste von Widerstandskämpfer:innen enthält, die wie unten strukturiert sind:

```
{
"Name":"Wolfgang Abendroth",
"Lebensdaten":"02. Mai 1906 - 15. September 1985",
"Beschreibung":"Wolfgang Abendroth wird 1906 in Elberfeld
   geboren und wächst in einer sozialdemokratischen...",
"Literatur":["Wolfgang Abendroth: Der gemeinsame Kampf mit
   den Griechen. ..." ]
}
```

Die Verarbeitungspipeline des *NDW* ist in Abbildung 2.2 nachzuvollziehen und lässt sich in zwei Teile spalten. Beim ersten Teil handelt es sich um eine klassische Pipeline zur Information Extraction, wie sie auch in [Alt21, S. 19] vorgestellt wird. Das Preprocessing ist hier etwas ausgelagert und säubert zuerst sämtliche Eingabetexte, bevor dann die eigentliche Pipeline beginnt.

Der Verarbeitungsprozess beginnt im Modul `information_extraction`. Dort wird nun die oben genannte Liste von Widerstandskämpfer:innen nach und nach abgearbeitet. Das Modul trägt dabei zu Recht diesen allgemeinen Namen, weil es als Ausgangspunkt der IE fungiert. Zuerst werden für die Beschreibung einer Person die Methoden des Part of Speech (POS)-Taggings und der NER durchgeführt, wie in Abschnitt 2.2.1 beschrieben. Da das Resultat dieser Methoden mit *flair* eine Liste von annotierten Sätzen ist und der Ansatz der Relation Extraction ebenfalls auf einzelne Sätze ausgelegt ist, läuft die weitere Pipeline für jeden Satz einer Beschreibung einzeln durch. Dazu wird das Modul `relation_extraction` aufgerufen, das die Klasse `RelationExtraction` enthält. Zu den Klassenattributen gehören neben den POS-Tags und NER-Annotationen auch eine Liste von „wikifizierten“ Entitäten, die hier mithilfe des Moduls `entity_linking` erzeugt wird. Auch die Coreference Resolution wird erst hier durchgeführt. Dieser etwas verschachtelte Weg macht Sinn, weil die „wikifizierten“ Entitäten und die Coreference Resolution nur für

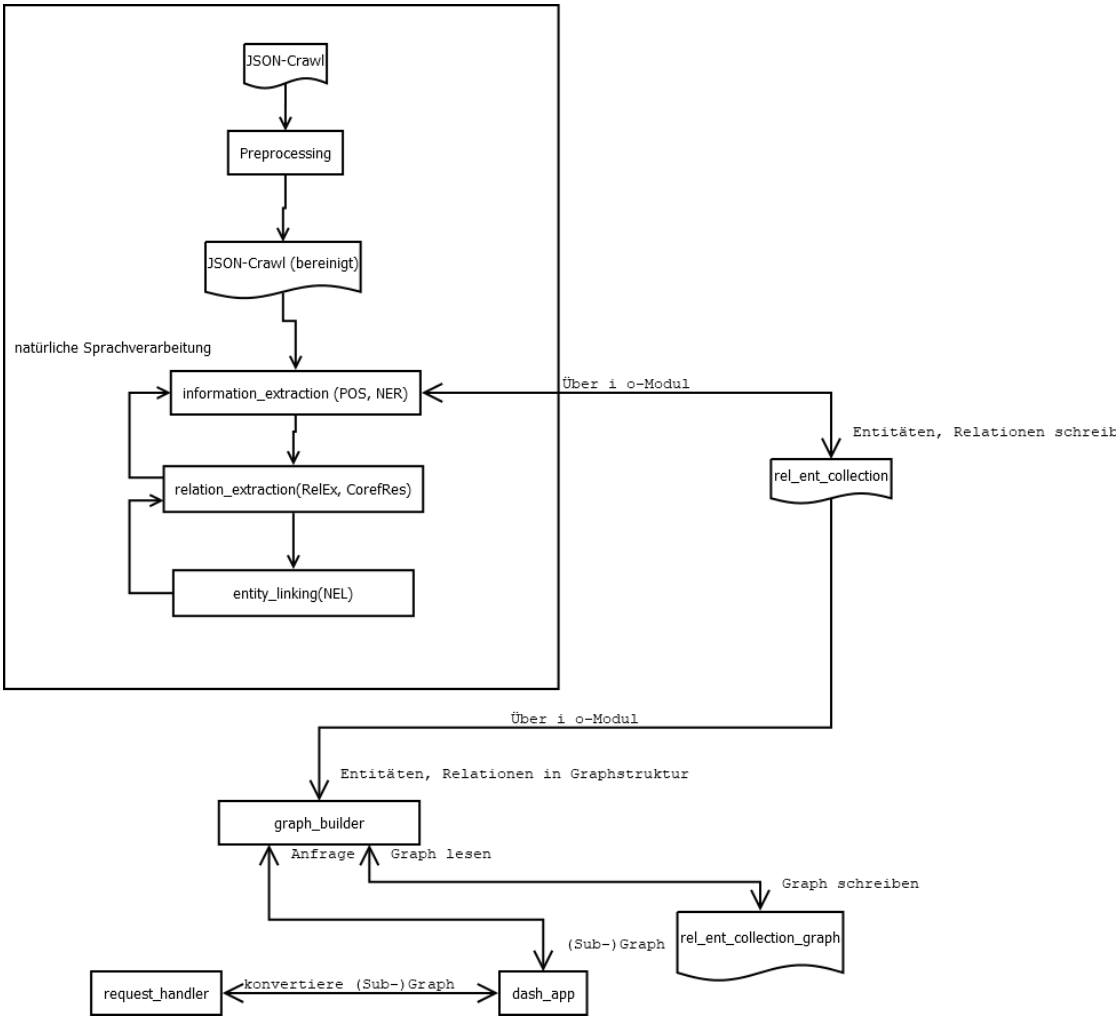

Abbildung 2.2: Der Ablaufplan der NDW-Pipline. In den Kästen sind die jeweiligen Module angegeben und in Klammern dahinter die dort ausgeführten IE Aufgaben.

die Relation Extraction benötigt werden, weil erst hier eine konkrete Analyse der Sätze erfolgt. Nach jeder verarbeiteten Person wird die `rel_ent_collection`, in der gefundene Relationen und Entitäten gesammelt werden, gespeichert.

Nachdem alle Personen auf diese Weise verarbeitet wurden, ist die IE-Pipeline abgeschlossen und der zweite Teil, der für die Graphgenerierung und die Visualisierung zuständig ist, kann beginnen. Hierbei ist zu betonen, dass diese beiden Teile logisch voneinander getrennt sind und nur über die oben erwähnte `rel_ent_collection`, die Ausgabedatei des ersten und Eingabedatei des zweiten Teil ist, miteinander verbunden sind. Das Modul `graph_builder` ist zum einen für die erste Generierung des Wissensgraphen zuständig, erstellt aber auch in der Visualisierung angefragte Subgraphen und fungiert somit auch als Backend für die in Abschnitt 2.2.2 beschriebene Webanwendung. Das Modul `dash_app` beschreibt das Frontend dieser Anwendung und fragt bei `graph_builder` den (Sub-)Graphen mit den von den Nutzern:innen gewählten Parametern an, während der `request_handler` als Midend die *NetworkX*-Graphen des `graph_builder` in Plotly-Graphen überführt und so eine Darstellung mit *Dash* ermöglicht.

2.1.3 Verwendete Daten

Für diese Arbeit wurde - wie schon im Text-Mining-Praktikum - ausschließlich auf einem Crawl der Biografiensammlung der Gedenkstätte Deutscher Widerstand (GDW)[7] gearbeitet. Diese Biografiensammlung ist aus verschiedenen Gründen nicht ideal. Zum einen fällt bei der Arbeit mit der Biografiensammlunge schnell auf, dass die GDW einen recht starken Fokus auf den militärischen Widerstand und besonders das Attentat vom 20. Juli 1944 legt. Zum anderen werden in der Sammlung der Gedenkstätte fast ausschließlich Widerstandskämpfer:innen aus Deutschland behandelt, während Widerstandskämpfer:innen aus den annektierten und besetzten Gebieten nur vereinzelt behandelt werden. Zu Beginn der Arbeit stand die Überlegung im Raum, den verwendeten Wissensschatz noch zu erweitern. Mögliche Kandidaten hierfür wären zum Beispiel ein Crawl der Wikipedia-Kategorie Person Widerstand gegen den Nationalsozialismus[8] oder die Sammlung von Stolperstein-Biografien vom Stolpersteine-Guide[9]. Allerdings hätte eine solche Erweiterung wahrscheinlich auch neue Probleme im Bereich des Preprocessing mit sich gebracht.

[7] Personenverzeichnis der GDW: https://www.gdw-berlin.de/vertiefung/biografien/personenverzeichnis/(Crawl vom Februar 2021)

[8] Wikipedia-Kategorie: Person Widerstand gegen den Nationalsozialismus: https://de.wikipedia.org/wiki/Kategorie:Person_(Widerstand_gegen_den_Nationalsozialismus)

[9] Stolperstein-Guide: https://stolpersteine-guide.de/map/

Möglich wäre außerdem, dass sich die Informationen der verschiedenen Quellen gegenseitig widersprechen oder ergänzen. Diese Problematik sollte bei der Erweiterung des Datensatzes berücksichtigt werden, ähnlich zum Vorgehen von von Keyserlingk-Rehbein [Key18, vgl. 57ff.]. Doch da sich diese Arbeit vorrangig auf die generelle technische Umsetzung konzentrieren soll, wurde dieses Thema hier zunächst nicht weiter vertieft.

2.2 Methoden und Umsetzung

2.2.1 Methoden der natürlichen Sprachverarbeitung

Nachfolgend sollen die Natural Language Processing (NLP)-Methoden näher betrachtet werden, die in dieser Arbeit eine Rolle gespielt haben. Ein großer Teil dieser Methoden bewegt sich dabei im Gebiet der Information Extraction. Es gilt:

> „Information extraction aims to automatically extract the desired information from unstructured text data so it can be analyzed or subsequently used in higher-level NLP tasks, such as [...] knowledge base population [...]. Information extraction systems typically process the text input in multiple steps, similar to a pipeline[...].“ (*[Alt21, S. 19]*)

Die einzelnen Schritte dieser Pipeline sind in Abschnitt 2.1 nachzuvollziehen und sollen im folgenden Teil beschrieben werden.

Coreference Resolution

Zur Klärung des Begriffs Coreference Resolution hilft schon eine Übersetzung ins Deutsche, also Auflösung von Koreferenz. Laut dem Linguistischen Wörterbuch von Theodor Lewandowski[10] handelt es sich bei einer Koreferenz um:

> „Die Eigenschaft verschiedener nominaler Ausdrücke bzw. Pronomen, sich innerhalb eines Textes auf dasselbe Referenzobjekt zu beziehen (Referenzidentität)“ (*[Lew94]*)

Diese Koreferenzen aufzulösen heißt also, die verschiedenen Ausdrücke und Pronomen wieder auf das ursprüngliche Referenzobjekt zurückzuführen. Dieser Schritt ist besonders in Bezug auf die Relation Extraction für die Pipeline relevant. Andernfalls könnte beispielsweise eine gefundene Relation zwischen zwei Pronomen niemals korrekt auf die referenzierten Entitäten verweisen. Außerdem werden die Texte in dem hier vorgestellten

[10] Eine aktualisierte Online-Version ist auf der Webseite des Bereichs Germanistik der Otto-v.-Guericke-Universität Magdeburg veröffentlicht. https://www.ger.ovgu.de/Fachgebiete/Germanistische+Linguistik (Zugriff 11.11.2021)

System satzweise verarbeitet. Daraus folgt, dass Sätze, die eine in einem vorigen Satz erscheinende Entität referenzieren, im Zweifel nicht richtig verarbeitet werden können.

Zur Umsetzung der Coreference Resolution wurde ein Versuch mit dem *CorZu*-System von Don Tuggener [Tug16], das auf dem *ParZu*-Modul von Rico Sennrich aufbaut [SVS13], durchgeführt. Letzteres fungiert als Dependency-Parser und benötigt weitere Tools zum POS-Tagging [SVS13] und zur morphologischen Analyse [SK14]. *CorZu* nutzt dann die Ergebnisse des Dependency-Parsings und des POS-Taggings, um die Koreferenzen innerhalb eines Textes aufzulösen, indem alle Ausdrücke, für die eine Referenzidentität vorliegt, mit einer gleichen Zahl versehen werden.

Es hat sich während der Arbeit mit *CorZu* relativ schnell herausgestellt, dass dieses Modul für die Pipeline nicht gut geeignet ist. Ein erstes Problem ist, dass *CorZu* in Python 2 geschrieben wurde, für die Nutzung von *flair* allerdings Python 3.6 oder höher benötigt wird. Die Coreference Resolution müsste also ausgelagert werden und in einem separaten Preprocessing-Schritt durchgeführt werden.

Ein weiteres Problem stellt das Ausgabeformat von *CorZu* dar. Standardmäßig wird ein Text in einem an CoNLL orientierten Format ausgegeben[11], mit der optionalen Möglichkeit einer Ausgabe in HTML. In beiden Formaten werden gefundene Koreferenzen auch nicht ersetzt, sondern sie werden lediglich mit der oben erwähnten Zahl markiert. Um hier eine Kompatibilität mit *flair* herzustellen, muss *CorZu* modifiziert werden, sodass auch eine Ausgabe des Eingabestrings mit ersetzten Koreferenzen möglich ist. Diese Aufgabe gestaltete sich als relativ aufwändig, unter anderem auch, weil *CorZu* nur dürftig dokumentiert ist und aus mehreren Python-Skripten besteht, die wiederum mit einem shell-Skript aneinandergereiht sind.

Erste Tests zur Coreference Resolution mit unserem Datensatz zeigten schnell eine weitere große Schwäche von *CorZu*. Das Tool arbeitet nur sehr unzuverlässig und besonders in Texten mit einer etwas komplexeren Satzstruktur werden derart viele und gravierende Fehler erzeugt, dass eine automatische Coreference Resolution mit *CorZu* zu großen Aufwand bedeutet. Beim Versuch, auftretende Fehler mit einem halbautomatischen Ansatz zu beheben, stellte sich dann schnell heraus, dass die Fehlerquellen sehr vielfältig sind und in einer so großen Zahl auftreten, dass eine manuelle Coreference Resolution vermutlich nur unwesentlich Mehraufwand bedeutet hätte.

Im Beispiel 2.2.1 sollen einige Fehlerquellen präsentiert werden, die bei der Coreference Resolution mit *CorZu* auftreten. Es wurden einige Spalten des erweiterten CoNLL-Formats ausgespart, um Platz zu sparen. Zu sehen sind die nur Spalten token ID, lexeme

[11] Das CoNLL-Format wird zum Beispiel hier vorgestellt [TD03]

und coreference.

Im Beispiel sind zwei häufige und nur schwer zu behebende Fehler zu sehen, die bei der Anwendung von *CorZu* auftreten:

1. Der Name der Widerstandskämpferin Cora Berliner wird nicht vollständig erkannt.
2. Die Koreferenz im zweiten Satz wird nicht korrekt aufgelöst.[12]

Darüber hinaus treten in anderen Texten noch weitere Fehler auf, die eine händische Korrektur weiter verkomplizieren.

```
1     Cora     (0)
2     Berliner     -
3     studiert     -
4     Mathematik     (1)
5     und     -
6     Staats-     -
7     und     -
8     Rechtswissenschaft     -
9     in     -
10     Berlin     (3)
11     und     -
12     Heidelberg     (2)
13     und     -
14     promoviert     -
15     1916     -
16     .     -

1     Bis     -
2     1919     -
3     ist     -
4     sie     (1)
5     Angestellte     -
6     der     -
7     Stadtverwaltung     -
8     Schöneberg     -
```

[12]Statt der referenzierten Cora Berliner wird Mathematik als Koreferenz erkannt.

```
9    ,    -
10    daneben    -
11    Dezernentin    -
12    ,    -
13    später    -
14    Geschäftsführerin    -
15    und    -
16    Vorstandsvorsitzende    -
17    beim    -
18    Verband    -
19    jüdischer    -
20    Jugendvereine    -
21    in    -
22    Heidelberg    (2)
23    .    -
```

Aufgrund des großen Aufwands, der bereits in die Coreference Resolution ohne zufriedenstellende Ergebnisse geflossen ist, wurde schließlich von einer weiteren Verwendung von *CorZu* abgesehen. Stattdessen wurde eine relativ einfache Heuristik entwickelt, die in Sätzen nach Vorkommen der Personalpronomen ‚er' und ‚sie', sowie nach Kurzformen oder alternativen Schreibweisen der Person, die gerade bearbeitet wird, sucht und diese dann zur Liste der gefundenen Entitäten hinzufügt. Dabei besteht die Annahme, dass sich diese Koreferenzen - insbesondere die Personalpronomen - mit hoher Wahrscheinlichkeit genau auf die Person beziehen, die in dem Text behandelt wird.

Eine Auswertung dieser Heuristik ergibt, dass in mehr als 2.600 Sätzen eine Entitätsergänzung stattfindet, wobei in einer Überprüfung von 260 zufällig ausgewählten Sätzen 10 Falschzuordnungen gefunden wurden. Dabei ist zu erwähnen, dass die meisten dieser Fehler auf die Mehrdeutigkeit des Pronomens „sie" zurückzuführen sind und keine schwerwiegenden Fehlinformationen bewirken. Zum Beispiel taucht in der Biografie von Maria Terwiel der folgende Satz auf:

> „ Sie lernen Harro Schulze-Boysen und John Graudenz kennen und beteiligen sich an den Aktionen der Widerstandsgruppe um Schulze-Boysen . "

Hier bezieht sich das „Sie" auf Maria Terwiel und ihren Verlobten Helmut Himpel. Dennoch ist das Hinzufügen der Entität «Maria Terwiel » nicht falsch, sondern nur unvollständig und da in diesem Fall auch ein ähnlicher Satz in der Biografie von Helmut Himpel zu finden ist, durchaus akzeptabel. Es zeigt sich also, dass dieser Ansatz zumindest für

Datensätze, in denen sich die Pronomen so leicht auflösen lassen wie hier, eine recht hohe Erfolgsquote bei sehr geringem Aufwand hat und damit eine legitime Alternative zu *CorZu* darstellt.

Named Entity Recognition

Bei der Named Entity Recognition handelt es sich um einen der zentralen Aspekte im Bereich der IE. Ziel ist es, hierbei in einem Text verschiedene Entitäten zu erkennen und einer entsprechenden Kategorie zuzuordnen. Welche Kategorien dabei berücksichtigt werden, lässt sich nicht allgemein sagen, weil schon die Definition einer Named Entity (NE) nicht eindeutig ist. Während in einigen Werken beispielhaft:

> „Personen-, Firmen-, Produktnamen, komplexe Datums-, Zeit-, und Maßausdrücke“ (*[Car10]*)

als mögliche Named Entities genannt werden, betrachtet die CoNLL-Challenge von 2003 [TD03] ausschließlich Namen von Personen, Orten und Organisationen als Named Entites.

Da es sich bei der NER um so eine wichtige Aufgabe handelt, verwundert es nicht, dass es eine große Zahl an NER-Tools gibt[13], die diese Aufgabe mit unterschiedlichen Ansätzen und Methoden angehen. Trotz dieser großen Zahl an NER-Tools ist die Auswahl der nützlichen Trainingsdatensätze – insbesondere für die deutsche Sprache – stark eingeschränkt. Wenn ein NER-Tool überhaupt mit deutschen Texten arbeiten kann, handelt es sich bei den zugrundeliegenden Datensätzen meist um ConLL03 [TD03] oder GermEval2014 [Ben+14]. Daraus resultiert, dass die möglichen Kategorien, denen gefundene Entitäten zugeordnet werden können, auf PERson, LOCation, ORGanisation und OTHer/MISCellaneous begrenzt sind. Entitäten wie Mengen- oder Datumsangaben werden in beiden Corpora nicht berücksichtigt.

Ein Tool, das hier besonders hervorsticht, ist *flair*, das bereits in Abschnitt 1.2 angesprochen wurde. Dabei handelt es sich nicht nur um ein NER-Tool, vielmehr sollen damit perspektivisch eine Vielzahl an NLP-Aufgaben gelöst werden. So wurde in *flair* 0.9 zum Beispiel auch eine Möglichkeit zur Relation Extraction und zum NEL vorgestellt, aber noch ohne entsprechende Modelle für die deutsche Sprache. Gleichzeitig fungiert *flair* auch als PyTorch NLP-Framework und erleichtert so auch das Training von eigenen NLP-Modellen. Diese Möglichkeit wurde in dieser Arbeit aus Zeitgründen vernachläs-

[13] Die Webseite Papers With Code https://paperswithcode.com/task/named-entity-recognition-ner liefert alleine knapp 500 Veröffentlichungen zu diesem Thema (Stand 16.11.2021).

sigt, bietet aber interessante Perspektiven für einen möglichen Ausbau, wie in Abschnitt 3.2 beschrieben.

Der NER-Prozess konnte so ohne großen Aufwand gelöst werden. Hierfür stellt *flair* alle nötigen Schritte bereit, um von einem blanken Text zu einer annotierten Datenstruktur zu kommen. Texte, die aus mehreren Sätzen bestehen, werden hier zuerst mit einem sogenannten Sentence Splitter in eine Liste von Sätzen konvertiert. Standardmäßig wird in *flair* der Segtok Sentence Splitter verwendet. Anschließend können für jeden Satz in der resultierenden Liste die Named Entities erkannt und entsprechend getaggt werden. *Flair* stellt hierfür eine große Auswahl an verschiedenen vortrainierten NER-Modellen zur Verfügung. Diese unterscheiden sich vor allem in der trainierten Sprache. Teilweise gibt es auch für eine Sprache mehrere Modelle, die sich dann vor allem in der Größe der Trainingsdaten und damit im F1-Wert unterscheiden. So werden für die deutsche Sprache drei verschiedene Modelle bereitgestellt: 'de-ner-fast', 'de-ner' und 'de-ner-large'. Für diese Arbeit wurde das NER-Modell 'de-ner-large' genutzt, weil es mit einem F1-Wert von 92.31 das zuverlässigste Modell für die NER liefert, das derzeit für die deutsche Sprache zu finden ist.

Part-of-Speech-Tagging

Unter dem POS-Tagging wird die Klassifizierung jedes Tokens im Text gemäß seiner Wortart verstanden [EE10, vgl. 271]. Hierbei handelt es sich um einen der wenigen Prozesse dieser Arbeit, die nicht direkt in den Bereich der IE fallen. Dennoch liefert das POS-Tagging eine wichtige Grundlage für die später beschriebene Relation Extraction. Ähnlich wie bei der NER Trainingsdatensätze existieren, die das Spektrum der möglichen Annotationen vorgeben, gibt es verschiedene Tagsets für das POS-Tagging. Für die deutsche Sprache ist das bekannteste das Stuttgart Tübingen Tagset (STTS) [Sch+99], das zwar zwischen vielen verschiedenen Verbformen unterscheiden kann, aber beim Taggen von Substantiven keine Unterscheidung zwischen z.B. Genus, Kasus oder Numerus vornimmt.

Für das POS-Tagging stellt *flair* ein Modell bereit. Und auch hier handelt es sich mit einer Genauigkeit von 98.5 um den aktuellen State of the Art, für die deutsche Sprache. Zum Markieren der Wortarten wird auch in *flair* das häufig genutzte STTS genutzt.

Zudem stellt *flair* sogar eine Möglichkeit zur Verfügung, mehrere unterschiedliche Tagging-Prozesse in einem Schritt zu bearbeiten. So können die NER und das POS-Tagging schnell und unkompliziert in einem Arbeitsschritt gelöst werden.

Event Extraction

Laut [Hog+11] handelt es sich bei der Event Extraction um den Vorgang, Wissen in Bezug auf Ereignisse zu extrahieren. In dieser Arbeit soll die Event Extraction vor allem den Zeitpunkt eines Ereignisses bestimmen, da Wissen wie beteiligte Entitäten und die Relationen zwischen ihnen an anderer Stelle extrahiert wird. Mit dieser Methode eröffnet sich die Möglichkeit, extrahierte Relationen in zwei Arten aufzuteilen: zum einen die Ereignisse mit konkreter Zeitangabe und zum anderen die Fakten, die also unabhängig von einem Zeitpunkt sind. Da diese Arbeit vor allem auf geschichtlichen Texten basiert, ist zu erwarten, dass sich ein großer Teil der extrahierten Informationen der Gruppe der Ereignisse zuordnen lässt.

Für die Umsetzung der Event Extraction wurde der recht intuitive Ansatz des Prototyps übernommen. Die Annahme ist dort: die Möglichkeit, Datumsangaben in deutschen Texten zu verwenden, lässt sich auf einige leicht erkennbare Muster reduzieren. Daher wurden drei Muster für reguläre Ausdrücke formuliert, die einen großen Teil der Datumsangaben abdecken, die in dem Datensatz auftauchen. Zeitangaben, die sich auf zuvor genannte Daten beziehen (z.B. „5 Jahre später“ oder „kurz darauf“,...), bleiben mit diesem Ansatz aber verdeckt.

Named Entity Linking

Wie bereits in Abschnitt 1.4 angesprochen, gibt es im Prototyp noch einige Probleme mit Entitäten, die sich auf dieselbe Entität beziehen, aber nicht als diese erkannt werden. Zur Erinnerung: dort gibt es unter anderem je einen Knoten «Adolf Hitler», «Hitlers» und «Hitler», die offensichtlich alle dieselbe Person Adolf Hitler meinen. Dieses Problem lässt sich mit dem NEL angehen. Nach [Hac+13] handelt es sich hierbei um den Prozess, Vorkommen von Named Entities mit entsprechenden Einträgen in einer Wissensbank zu verknüpfen.

Für das NEL wurde hier der Ansatz der sogenannten Wikification verfolgt, was nichts anderes heißt, als dass es sich bei der zugrundeliegenden Wissensbank um einen Wikipedia-Dump handelt. Dazu wird das Tool *REL* verwendet, das eine solche Wikification sehr leicht macht [Hul+20]. Die Ersteller:innen liefern hier auch eine Web-API, die auf einem englischsprachigen Wikipedia-Dump basiert. *REL* bietet dabei zwei mögliche Herangehensweisen: Zum einen das dort so bezeichnete Entity Linking, das einen längeren Text erwartet und selbst zuerst eine NER durchführt. Und zum anderen die sogenannte Entity Disambiguation, die als Eingabe nur eine NE erwartet. Interessant an diesem Tool ist,

dass es ebenfalls auf dem *flair*-Framework aufbaut, was noch einmal die Relevanz von *flair* für den Bereich der natürlichen Sprachverarbeitung hervorhebt.

Wie aus dem folgenden Vorgehen klar werden sollte, zeigt sich hier der Nutzen der in Abschnitt 2.1 beschriebenen Entitätsklassen. Durch sie wird das Zusammenführen verschiedener Entitäten in einem Objekt ermöglicht.

Jede gefundene Entität in einem Satz durchläuft vor der weiteren Verarbeitung den nachfolgenden NEL Prozess:

1. Übergabe der Entität an die *REL*-API, als Eingabe wird die Entität und eine Liste `[(0,len(entity))]`[14] abgeschickt.

2. *REL* liefert als Ergebnis für jeden gefundenen Eigennamen eine Liste der Struktur:

```
[Index_Begin_of_Entity, Index_End_of_Entity,
   Entity_Text, Wikification_Result,
   Confidence_Score_Wikification,
   Confidence_Score_NER, NER_Tag]
```

   Für das NEL ist nur der Wert des `Wikification_Result` von Interesse, eine Überprüfung des Confidence Scores findet nicht statt, da der eher wenig über die Korrektheit von `Wikification_Result` aussagt[15].

3. Im nächsten Schritt findet dann das eigentliche NEL statt. Hierzu wird der Wert von `Wikification_Result` in dem dictionary `entity_collection` gesucht, das folgende Struktur hat:

```
entity_collection = {Wikification_Result:
   Entityclass}
```

4. Falls sich `Wikification_Result` bereits in der Sammlung befindet, wird das zugehörige Objekt zurückgegeben.

5. Andernfalls wird zuerst ein neues Objekt für die jeweilige Entität angelegt und in `entity_collection` gespeichert.

[14] Ohne diese Liste würde *REL* selber nach Entitäten in dem gegeben Text suchen, das würde z.B. bei langen Eigennamen wie Claus Schenk Graf von Stauffenberg Probleme bereiten.

[15] Z.B. hat die falsche Wikification von „Frings“ zu „Torsten_Frings“ einen Confidence-Score von 0.95, während die richtige Zuordnung von „Wolfgang Abendroth“ auf „Wolfgang_Abendroth“ einen Confidence-Score von 0.39 hat.

6. Wenn das Ergebnis von *REL* leer ist, wird trotzdem eine Suche in `entity_collection` durchgeführt, in der Annahme, dass vorige Anfragen erfolgreicher waren.

Bei Bedarf kann außerdem auch für „unwikifizierte“ Entitäten ein Objekt angelegt und in `entity_collection` gespeichert werden. Dieses Verhalten ist zum Beispiel interessant, wenn die Pipeline auf weniger gut erforschte Domänen angewandt wird.

Da es sich bei den gefundenen Entitäten häufig um relativ bekannte deutschsprachige Eigennamen handelt, ist die Erfolgsrate von *REL*, trotz des englischsprachigen Wikipedia-Dumps, akzeptabel. Zu Problemen kommt es häufiger bei kleineren deutschen Ortschaften, die einen Namensvetter in den USA haben, oder wenn nur der Nachname einer Person in einem Satz auftaucht. Als Beispiel sei hier ein Satz aus der Kurzbiografie des Kardinals Josef Frings gegeben:

> „Frings tritt in seinen Predigten immer wieder für Verfolgte und gegen staatliche Unterdrückung ein.“

Hier wird von *flair* „Frings“ vollkommen richtig als Personenname erkannt und zur Wikification an *REL* weitergegeben. Weil es aber mehrere Einträge für das Wort Frings gibt[16], hat *REL* keine Möglichkeit, ohne weiteren Kontext den korrekten Frings zu erkennen, weshalb hier der Fußballspieler Torsten Frings anstelle des eigentlich gemeinten Kardinals Josef Frings, als Resultat zurückgegeben wird. In diesem Beispiel hat dieses Problem keine Auswirkung auf den Graphen, da keine Relationen in diesem Satz zu finden sind. Bei anderen Fällen stellt das aber ein Problem dar, dessen Lösung *REL* nicht leistet und alternative Ansätze zum NEL erfordert.

Relation Extraction

Mit dem Abschluss des NEL bleibt noch die Relation Extraction:

> „At this point only the structure between entities is missing. The objective of relation extraction is to establish this structure by detecting relationships between the mentioned entities or concepts.“ (*[Alt21, S. 20]*)

Dazu gibt es für diese Aufgabe eine Vielzahl verschiedener Ansätze, die von Alt in die vier Kategorien „traditional“, „distant“, „semi“ und „no supervision“ eingeordnet werden. Grundlegende Annahme für die Relation Extraction ist häufig, dass genau dann eine Relation vorliegt, wenn in einem Satz zwei oder mehr Entitäten zu finden sind.

Die Anzahl der miteinander in Relation stehenden Entitäten wird dabei Stelligkeit genannt. Häufig werden bei der Relation Extraction vor allem zweistellige Relationen

[16] siehe https://en.wikipedia.org/wiki/Frings

betrachtet. Um aber den Informationsgehalt hier möglichst hoch zu halten, wurden in dieser Arbeit auch mehrstellige Relationen betrachtet. Da es sich aber um ein Netzwerk mit Fokus auf Personen handelt, wurde für mehrstellige Relationen die zusätzliche Annahme getroffen, dass ausschließlich alle Entitäten vom Typ Person mit allen anderen Entitäten in Relation stehen.

An dieser Stelle nutzt jetzt die oben beschriebene Coreference Resolution. Durch diese Methode können auch Entitäten entdeckt werden, die durch Koreferenzen in Sätzen codiert sind, in denen sonst nur eine Entität auftauchen würde.

In dieser Arbeit wurde ein Ansatz gewählt, der auf lexikalischen Merkmalen basiert, was nach [Alt21] in die Kategorie der traditional supervised Relation Extraction fällt. Alt zählt hierzu unter anderem das Ausnutzen von sogenannten Triggerwörtern und von POS-Tags zum Erkennen von Relationen.

Hier handelt es sich um eine Zusammensetzung von beidem. POS-Tags werden genutzt, um alle finiten Vollverben, alle nicht flektierten Vollverben und alle abgetrennten Verbzusätze zu extrahieren. Im STTS sind das die Tags: VVFIN, VVPP und PTKVZ. Die Entscheidung ist auf diese Tags gefallen, weil sie den höchsten Informationsgehalt für eine gesuchte Relation liefern.

Um aber zu verhindern, dass aus Sätzen wie z.B.:

„ 1925 wird [Hans Adloch] Vizepräsident des [Stuttgarter Katholikentages].“

Relationen wie:

Hans Adloch - wird - Stuttgarter Katholikentages

extrahiert werden, werden auxiliare und nicht flektierte Verben hier nicht berücksichtigt. Wie im obigen Beispiel schnell klar wird, kann eine solche Relation das Verhältnis zwischen Adloch und dem Katholikentag nicht erfassen. Eine deutlich aussagekräftigere Relation wäre hier:

Hans Adloch - Vizepräsident - Stuttgarter Katholikentages

Das Verhältnis zwischen der Person Hans Adloch und der Organisation Stuttgarter Katholikentag, wird sofort deutlich. Es handelt sich um den Vizepräsidenten derselben.

Hier kommen also die Triggerwörter hinzu, denn Informationen wie Kasus und Numerus werden nicht im STTS berücksichtigt.

Um ein gutes Gleichgewicht zwischen aussagekräftigen Relationen und der Generalisierung von ähnlichen Relationen zu erreichen, wurden diese Regeln abhängig von den gefundenen Entitätstypen gruppiert. So gibt es beispielsweise die Regelgruppe PER_LOC

(siehe Listing 2.2.1), die Anwendung findet, wenn an der Relation mindestens eine Person und ein Ort beteiligt sind. In dieser Gruppe sind wiederum semantisch ähnliche Schlüsselwörter zu einer Relation zusammengefasst. Der Großteil dieser Regeln wurde dabei aufgestellt, nachdem ein erstes Mal die Relationen aus den POS-Tags extrahiert wurden. Und zwar auf Basis aller Sätze, in denen zwei oder mehr Entitäten aber keine Relation gefunden wurden[17]. Sehr häufig handelt es sich bei den Schlüsselwörtern um Berufs- und Amtsbezeichnungen, offenbar werden diese besonders oft mit auxiliaren Verben verwendet.

```
PER_LOC = {
    'geboren_in': ['zur Welt', 'geborene'],
    'studiert': ['Studium'],
    'Rückkehr': ['Rückkehr', 'zurückkehren'],
    'befreit': ['Befreiung'],
    'arbeitet_in': ['Landgerichtsdirektor', 'Landgerichtsrat
       ', 'Kommandant der Invalidensiedlung', 'Direktor', 'tä
       tig', 'aktivsten Köpfe der Bekennende Kirche', '
       Pfarrer', 'Rechtsanwalt', 'Redakteur', 'Tuberkulose-Fü
       rsorgerin', 'Superintendent', 'Finanzfachmann', '
       Stadtkämmerer', 'Polit-Kommissar', 'Vizepräsident'],
    'kämpft': ['militärische Auseinandersetzung'],
    'Attentat': ['Anschlag'],
    'Kontakte_nach': ['Kontakt'],
    'politisches Amt': ['Vorsitzender', 'Landrat', '
       Ministerpräsident', 'politischer Sekretär', 'Oberbü
       rgermeister', 'Regierungsreferendar','Oberpräsident',
       'Statdverordneter', 'Vorsitzender', 'Kanzler', '
       minister', 'Landtagsmitglied', 'Reichstagsabgeordneter
       '],
    'Migration': ['emigrieren', 'verlassen', 'ausreisen', '
       entkommen'],
    'lebt_in': ['Mitglied', 'überleben'],
    'stationiert_in': ['Inspekteur', 'Rittmeister', 'Militä
       rbefehlshaber', 'Divisionskommandeur', 'Chef des
```

[17] etwa 150 Sätze

```
    Stabes'],
'Regimekritiker': ['Regimekritiker'],
'Zwangsarbeit_in': ['Zwangsarbeit'],
'Aufenthalt_in': ['kongress in'],
'Flucht': ['freikommen']
    }
```

Der entscheidende Vorteil an dieser Gruppierung der Regeln liegt darin, dass z.B. Berufsbezeichnungen mehrfach in verschiednenen Regelgruppen vorkommen können und abhängig von den beteiligten Entitätsklassen entsprechend sinnvolle Relationen gesetzt werden. So taucht zum Beispiel in mehreren Sätzen das Triggerwort „Chef des Stabes“ auf, was in einigen Fällen aber eine Relation von zwei Personen und in anderen Fällen eine Relation zwischen einer Person und einem Ort ist.

Auf diese Weise können insgesamt 1862 Relationen extrahiert werden, die als Kanten im Wissensgraph auftauchen. Dabei werden 684 verschiedene Relationstypen extrahiert, wobei am häufigsten die Typen „verurteilt“ (128), „ermordet“ (127), „Bekanntschaft“ (107), „festgenommen“ (83) und „verhaftet“ (79) auftreten. Eine ausführlichere Auswertung findet sich im Anhang 4.1.

2.2.2 Das Wissen in den Graphen bringen

An die IE-Pipeline schließt sich nun die Speicherung zur weiteren Verarbeitung im Wissengraphen an. Wie in Abschnitt 2.1 bereits erläutert, ist das Resultat der IE-Pipeline eine Sammlung aller gefundenen Relationen und der beteiligten Entitäten, die nun mithilfe der in [HSS08] beschriebenen Graphbibliothek *NetworkX* in eine Graphstruktur übertragen wird. Wie bereits in Abschnitt 2.1.1 angesprochen, liegt ein bedeutender Vorteil von *NetworkX* in der Möglichkeit, auch selbstdefinierte Datenstrukturen als Knoten zu speichern. Darüber hinaus zeichnet sich die Bibliothek durch ihre sehr gute Dokumentation, ihre einfache Benutzung und ihre vielfältigen Möglichkeiten zur Graphmanipulation und -analyse aus.

Zur Überführung in eine Graphstruktur werden zunächst für jede gefundene Relation die beteiligten Entitätsobjekte als Knoten gespeichert. Jeder Knoten speichert als Knotenattribut außerdem den jeweiligen NE-Tag sowie eine für diesen Tag festgelegte Farbe[18]. Anschließend wird dann die entsprechende Relation als Kante zwischen den Knoten hinzugefügt. Auch hier werden zusätzliche Kantenattribute gespeichert. Neben dem Namen

[18] PER-grün, LOC-blau, ORG-rot

der Relation[19] ist das zum einen das etwaige extrahierte Datum der Relation und der Satz in dem die Relation gefunden wurde.

Graphstatistiken

Der so generierte Wissensgraph kann nun auf seine strukturellen Eigenschaften hin analysiert werden. Graphentheoretische Fragen wie Zusammenhang, Durchmesser, Knoten mit maximalem Grad, Knoten- und Kantenanzahl können bereits eine erste Idee zum Aufbau des Widerstandsnetzwerks liefern. Auch ein Ranking der Knoten nach Grad liefert schon einige Informationen. Im Anhang 4.1 finden sich die Graphstatistiken für das Ergebnis der vorliegenden Arbeit.

2.2.3 Graphexploration durch Wissensabfragen

Um die Ergebnisse der Pipeline auch veranschaulichen zu können, lohnt es sich auch, erste Ideen für eine Visualisierung zu formulieren und umzusetzen. Hier wurde sich stark an den Überlegungen und der Umsetzung der Visualisierung des Prototyps orientiert. An erster Stelle steht also die Darstellung des Wissensgraphen als ungerichteter Graph in einer Benutzeroberfläche. In dieser Oberfläche sollen verschiedene Wissensabfragen als Eingabe ermöglicht werden, die mit der Ausgabe eines entsprechenden Teilgraphen beantwortet werden sollen. Damit diese Anwendung möglichst niedrigschwellig angeboten werden kann, soll sie als Web-App bereitgestellt und so ohne großen Aufwand in sämtlichen Browsern erreicht werden.

Visualisierung in der Web-App

Wie bereits erwähnt wurde bei der Umsetzung der Web-Anwendung auf den Ergebnissen des Prototyps aufgebaut. Besonders das Midend, das den *NetworkX*-Graphen in einen *plotly*-Graphen [Inc15] überführt, konnte hier mit minimalem Aufwand übernommen werden. Die einzige Änderung, die hier von Relevanz war, war das Anpassen der Informationen, die aus dem Graphen geholt werden müssen. Für das Frontend wurde das Framework *Dash* [Hos19] genutzt, das von den *plotly*-Entwicklern zur Erstellung von Datenanalyse-Anwendungen bereitgestellt wird. *Dash* zeichnet sich auch durch seine gut nachvollziehbare Dokumentation und die einfache Verwendung aus.

[19] also die extrahierte Relation

Mögliche Wissensabfragen

Mit der umgesetzten Web-App sind einige sehr unterschiedliche Wissensabfragen möglich.

- **Knotensuche:** Eingegeben wird die Bezeichnung einer bestimmten Entität (z.B. KPD). Ausgegeben wird der Wissensgraph für den gesuchten Knoten, oder ein leerer Graph, falls der Knoten nicht existiert.
- **Kämpfer:innen-Suche:** Wie bei der Knotensuche wird der Subgraph für eine:n bestimmte:n Widerstandskämpfer:in ausgegeben. Die Eingabe erfolgt dagegen durch die Auswahl aus einer Liste aller Widerstandskämpfer:innen.
- **Pfadsuche:** Eingegeben wird eine kommaseparierte Liste von Knoten. Die Ausgabe ist der kürzeste Pfad, der alle angegebenen Knoten verbindet.
- **Relationensuche:** Im Unterschied zu allen vorigen Abfragen stehen hier nicht die Knoten, sondern die Kanten, also die Beziehungen zwischen verschiedenen Entitäten, im Mittelpunkt. Erwartet wird ein Begriff, der auch in der Menge der Relationen vorkommt. Ausgegeben werden alle Kanten, die diesen Begriff beinhalten.

Diese vier Grundabfragen können außerdem angepasst werden. Zum einen können sie zum Teil miteinander kombiniert werden. Zum Beispiel kann mit einer Kombination aus Knoten- und Relationensuche nach allen Knoten gesucht werden, die in der Relation „geboren“ mit dem Knoten Berlin stehen. Des Weiteren ist es möglich, bestimmte Abfragen mittels Suchtiefe und Entitätenfilter weiter zu konkretisieren. Die Suchtiefe ist standardmäßig auf 1 gesetzt, d.h. für einen gesuchten Knoten werden der Knoten selbst und alle seine direkten Nachbarn ausgegeben. Wenn nun die Frage besteht, mit welchen Entitäten ein Knoten indirekt in Beziehung steht, kann die Suchtiefe auf 2 gestellt werden, wodurch auch alle Nachbarn der Nachbarknoten mit ausgegeben werden.

2.3 Anwendungsfälle für das Netzwerk des Widerstands

Für das *SoNAR*-Projekt werden in der Projektbeschreibung mehrere mögliche Anwendungsfälle beschrieben. Weil es sich dabei um eine Forschungstechnologie handelt, sind auch die Ideen zur Anwendung vor allem im Forschungsbereich zu verorten, laut [MSO22] kann *SoNAR* Forscher:innen von der Planung über die Vorbereitung bis zur Durchführung in ihrer Arbeit unterstützen. Konkret werden als mögliche Anwendungsfälle von *SoNAR* unter anderem die Entwicklung einer Forschungsfrage, die Datenexploration, die

Recherche von Informationen, die Ermittlung von Datenquellen, und die Analyse der Daten genannt [MSO22]. Diese Anwendungsfälle von *SoNAR* lassen sich in Teilen auch auf das *NDW* übertragen. Insbesondere in den ersten Forschungsschritten kann auch das *NDW* Forscher:innen unterstützen.

Darüber hinaus liefert auch die Webseite des Digitalen Deutschen Frauenarchivs Inspiration für einen Anwendungsfall. Dort wird unter anderem über wichtige Akteurinnen der Frauenbewegung informiert[20]. Auch ein Wissensgraph ist dort integriert und lädt zur weiteren Erkundung ein. Das Digitale Deutsche Frauenarchiv zeigt so also, dass solche Netzwerke nicht nur in der Forschung relevant sind, sondern auch die Lehre und das Vermitteln von Wissen unterstützen und sogar digitalisieren können.

Eine entsprechende Einbindung des *NDW*, zum Beispiel im Personenverzeichnis der GDW, könnte auf ähnliche Weise wirken und die Arbeit mit diesem ansprechender gestalten. Es sollte aber betont werden, dass die Nutzung des *NDW* ohne weiteres Kontextwissen z.B. durch ergänzende Texte oder bereits erlerntes Wissen sich eher nicht anbietet, da ein solches Netzwerk sehr komplexe Zusammenhänge stark verkürzt .

[20] Akteurinnen der Frauenbewegung https://www.digitales-deutsches-frauenarchiv.de/akteurinnen?term= (abgerufen am 25.01.2022)

3 Fazit

3.1 Auswertung

Eine Auswertung der ohne weitere Tools umgesetzten Methoden findet sich in der jeweiligen Methodenbeschreibung. Zur Auswertung der in dieser Arbeit beschriebenen Pipeline wurden zwei Ansätze gewählt. Zum einen wird der Graph mit den Ergebnissen aus [Key18] verglichen. Zum anderen werden stichprobenartig die Teilgraphen von 11 Widerstandskämpfer:innen mit den entsprechenden Kurzbiografien verglichen.

3.1.1 Vergleich

Der Vergleich des hier generierten Wissensgraphen mit den Ergebnissen von von Keyserlingk-Rehbein erfolgte anhand der Quellenbelege für die Auswertung der Kontaktmerkmale [Key18, 585.ff]. Dort sind für alle Personen des sogenannten Netzwerks des 20. Juli 1944 alle nachgewiesenen Kontakte mit jeweiliger Quellenangabe hinterlegt. Diese Angaben können also mit dem entstandenen Wissensgraphen verglichen werden. Dazu wird zuerst in dem *NetworkX*-Graphen nach einem Knoten mit dem jeweils zuerst genannten Namen gesucht und anschließend überprüft, ob sich ein Knoten mit dem zweiten Namen unter den Nachbarn des ersten Knoten befindet. Diese Überprüfung führt zu den folgenden Ergebnissen:

Von Keyserlingk-Rehbein hat insgesamt 776 Relationen zwischen 135 verschiedenen Personen ermittelt. Von diesen 776 Relationen konnten nur 18 auch im Wissensgraph gefunden werden.

Für das Fehlen der restlichen 758 Relationen gibt es verschiedene Erklärungesansätze. So wurden von den oben genannten 135 Personen 24 ebenfalls nicht im Wissensgraphen gefunden. Doch auch wenn diese aus der Rechnung ausgeschlossen werden, können 646 Relationen nicht bestätigt werden.

Der eigentliche Grund für die niedrige Erfolgsquote liegt vermutlich in den doch sehr unterschiedlichen Forschungsansätzen und Quellen der beiden Arbeiten. Während von Keyserlingk-Rehbein eine Viezahl von Primärquellen verwendet und akribisch ausgewer-

tet hat, basiert der Wissensgraph auf Forschungsquellen, die als Kurzbiografien stark komprimiertes Wissen beinhalten. Damit ist diese Überprüfung weniger ein Vergleich zwischen dem Wissensgraphen und der Arbeit von von Keyserlingk-Rehbein, sondern im weiteren Sinne auch ein Vergleich der Biographiensammlung der GDW mit dem Wissen der Gestapo über das Netzwerk des 20. Juli. Das heißt also, dass fehlendes Wissen im Graphen nicht unbedingt auf Fehler in der Pipeline hinweist, sondern die Informationen auch schon in den Kurzbiografien nicht erwähnt worden sein können.

Gleichzeitig ist aber auch das Wissen der Verfolger keine besonders zuverlässige Quelle. So hebt von Keyserlingk-Rehbein hervor:

> „dass die Verhörten darum bemüht waren, ihre Freunde und Bekannten zu schützen und dass sie daher versuchten, ihre tatsächlichen Kontakte zu Mitverschwörern zu verschweigen, sie zu bagatellisieren oder auch falsch darzustellen.“ (*[Key18, S. 74]*)

3.1.2 Stichprobe

Um die eben aufgestellte These zu stützen, sollen stichprobenartig 11 der 111[21] übereinstimmenden Personen genauer überprüft werden. Dazu wurden 11 Namen von Widerstandskämpfer:innen randomisiert ausgegeben und deren Teilgraphen mit ihren Ursprungstexten verglichen. Dabei wurden die Relationen in eine der vier Kategorien eingeteilt:

- richtige Verarbeitung (jede Relation, die korrekt verarbeitet wurde)
- falsche Entität (jede Relation, die eine falsch zugeordnete Entität enthält)
- fehlende Entität (fehlende Relation, wegen fehlender Entität)
- falsche Relation (jede Relation, mit korrekten Entitäten, aber mit Fehler in der Relation Extraction)

Mithilfe dieser Einteilung können dann Rückschlüsse auf die wahrscheinlichen Fehlerquellen gezogen werden. Eine falsche Entität kann zum einen mit einer falschen Coreference Resolution oder mit einem Fehler im NEL zusammenhängen. Während bei einer fehlenden Entität ein Fehler im NEL oder in der NER wahrscheinlich ist. Dagegen ist eine falsche Relation ausschließlich auf einen Fehler in der Relation Extraction zurückzuführen. Eine Übersicht der Auswertung ist in Tabelle 3.1 zu finden. Die zugrunde liegenden Texte und Teilgraphen sind im Anhang 4.2 aufgeführt.

[21] also ca. 10%

	Kra	Mol	Kem	Doh	Mie	Höß	Kla	Kle	Leu	Sch	Kai	gesamt
Entität fehlt	3	4	7	4	6	5	9	5	8	7	4	62
Entität falsch	0	1	1	1	0	0	2	0	0	1	1	7
Relation falsch	1	1	0	2	0	0	0	2	0	0	0	6
korrekt	3	4	4	4	3	4	1	2	3	0	4	32

Tabelle 3.1: Auswertung der 11 Widerstandskämpfer

Es fällt schnell auf, dass das größte Problem fehlende Entitäten sind. Die These aus Abschnitt 3.1.1 ist damit aber nicht widerlegt, da es sich bei vielen der fehlenden Entitäten eher um Organisationen, wie Parteien oder Einrichtungen des NS-Staats handelt. So wurden die meisten der 11 Widerstandskämpfer vom Volksgerichtshof verurteilt, der von *flair* gar nicht als Organisation erkannt wird. Es ist außerdem denkbar, dass weitere Fehler im *NEL*-Prozess entstehen, da nicht-wikifizierbare Entitäten in dieser Arbeit nicht berücksichtigt werden. Dennoch ist nicht von der Hand zu weisen, dass auch Personen, die teilweise sogar in der Biografiensammlung vertreten sind, zu den fehlenden Entitäten gehören. Solche Fehler sind vermutlich in der Pipeline selbst zu suchen.

3.2 Ausblick

Die Arbeit an dieser Pipeline ist noch nicht abgeschlossen und bietet viele Ausbaumöglichkeiten. Wie aus Abschnitt 3.1 hervorgeht, gibt es noch einige Fehlerquellen, die behoben werden müssen, bevor dieser Wissensgraph präsentiert oder ausgestellt wird.

3.2.1 Verbesserungspotenzial

Im Fokus von nachfolgenden Überarbeitungen sollte vor allem das NEL stehen. Die dort entstehenden Fehler können leicht zu falschen Darstellungen und Interpretationen im Ergebnisgraphen führen. Hier sollte eine deutliche Verbesserung der Ergebnisse erreicht werden, indem für *REL* ein deutschsprachiger aktueller Wikipedia-Dump genutzt wird. Dabei sollte aber auch die in Abschnitt 2.2.1 angesprochene Problematik berücksichtigt werden, dass *REL* bei Mehrfachtreffern stets das erste gefundene Ergebnis liefert. Eine mögliche Herangehensweise wäre hierfür ein Ansatz, der stärker auf den jeweiligen Kontext einer Entität eingeht und zum Beispiel bei der Personensuche auch das Geburtsdatum berücksichtigt. Das könnte auch das in Abschnitt 1.4 angesprochene Problem der Namensdopplungen lösen, das in dieser Arbeit nicht gelöst wurde.

Auch die Methode der Coreference Resolution hat noch Verbesserungspotenzial bzw. wäre im Allgemeinen ein zuverlässiges Tool für diese Methode für die deutsche Sprache wünschenswert. Wie aber schon allein die Notwendigkeit einer Dissertation zu diesem Thema [Tug16] zeigt, ist dieses Problem nicht trivial und wird vermutlich noch einige Jahre Forschungsarbeit kosten.

Zusätzlich kann aber auch die Relation Extraction noch ausgebaut werden. So gibt es bereits einige neuronale Ansätze für diese Aufgabe, aber bis jetzt fehlen noch die entsprechenden deutschsprachigen Modelle.

An dieser Stelle seien noch einmal die neuen Funktionalitäten von *flair* erwähnt, mit denen vor allem die Probleme des NEL und der Relation Extraction überarbeitet werden können. Dazu müssen aber noch passende Datensätze erarbeitet oder gefunden und die entsprechenden Modelle trainiert und validiert werden. Auch das Training eines domänenspezifischen NER-Modells mit *flair*, kann die Ergebnisse verbessern. Denn wie in Abschnitt 3.1.2 angesprochen werden einige der Institutionen des NS-Staats wie der Volksgerichtshof nicht als Organisationen erkannt.

3.2.2 Vergrößerung des Datensatzes

Eine der nächsten Aufgaben sollte auch die Auseinandersetzung mit der Erweiterbarkeit der Datenquellen sein. Denn wie in Abschnitt 3.3 deutlich gemacht wird, handelt es sich beim *NDW* keineswegs um eine vollständige Darstellung des Widerstandsnetzwerks. Um das ansatzweise realistisch darzustellen, reichen auch die in Abschnitt 2.1.3 vorgeschlagenen zusätzlichen Datenquellen nicht aus. Es sollte zusätzlich darüber nachgedacht werden auch (digitalisierte) Originalquellen in den Wissensgraphen miteinzubeziehen. Zentrale Fragen, die für die Vergrößerung des Datensatzes eine Rolle spielen, sind unter anderem:

- Wie verhält sich der Preprocessor zu neuen Quellen? Zum Teil sind die Schritte hier sehr spezifisch auf Besonderheiten in der Sammlung der GDW angepasst (z.B. Bildbeschreibungen löschen, Anpassung der Wehrkreisbezeichnungen). Ähnliche Besonderheiten sind mit jeder neuen Quelle zu erwarten.

- Wie verhält es sich, wenn die Arbeit mit Originalquellen fortgesetzt werden soll? Hier kommen ganz neue Problematiken ins Spiel, da nicht zu erwarten ist, dass die Originalquellen bereits digitalisiert geschweige denn transkribiert sind. Wie auch in [Men+21] festgestellt wird, ist schon der Prozess der Optical Character Recognition für alte Zeitungen nicht trivial und verringert den Erfolg der nachfolgenden Verarbeitungsschritte deutlich. Für handschriftliche Quellen, wie sie gerade in Strafakten

aus der NS-Zeit häufiger auftauchen, wird dieses Problem noch größer, da der Prozess der Handschrifterkennung insbesondere für die Sütterlin- und andere deutsche Kurrentschriften kaum ausgereift ist.

- Wie wird mit kontradiktorischen Quellen umgegangen? Vor allem mit Blick auf die Originalquellen muss davon ausgegangen werden, dass es deutliche Diskrepanzen zwischen verschiedenen Texten geben kann (z.B. in verschiedenen Verhörprotokollen). So weist von Keyserlingk-Rehbein beispielsweise darauf hin, dass in ihrer Arbeit Quellen mit durchaus gegensätzlichen oder widersprüchlichen Angaben auftauchen, aber teilweise Quellen auch unvollständige Informationen enthielten, die wiederum in anderen Quellen ergänzt wurden [Key18, vgl. S.57].

Eine weitere Möglichkeit, den Wissensgraphen zu erweitern, ohne dabei aber den Datensatz zu vergrößern, bietet das *SoNAR*-Projekt. Dort wird nicht nur die Möglichkeit gegeben, selber strukturierte Daten zur Wissensbasis hinzuzufügen, sondern es wird das dort extrahierte Wissen auch für sogenannte nachfolgende Systeme bereitgestellt. Das heißt das *NDW* und *SoNAR* können so gegenseitig voneinander profitieren und ihre jeweiligen Wissensbasen erweitern.

3.2.3 Weitere Ausbaumöglichkeiten

Zusätzlich können auch in der Visualisierung noch weitere Ergänzungen sinnvoll sein. Nennenswert wäre zum Beispiel ein Zeitstrahl, der alle gefundenen Datumsangaben einordnen kann. Auch die Einbindung der Graphstatistiken in die Visualisierung wäre eine Möglichkeit, die den Nutzer:innen wichtige Informationen liefern kann.

3.3 Zusammenfassung

In dieser Arbeit konnte eine IE-Pipeline aufgebaut werden, die wichtige Probleme des in 1.4 beschriebenen Prototyps bewältigt. Durch die Anwendung von NEL konnte insbesondere die Knotendopplung gelöst werden. Weiterhin liefert die Pipeline einen wesentlich schlankeren Wissensgraphen, der aber nicht weniger Informationen liefert als sein Vorgänger.

Dennoch muss betont werden, dass das Ergebnis dieser Arbeit keinesfalls den Anspruch haben kann, ein vollständiges Netzwerk des Widerstands gegen den Nationalsozialismus darzustellen. Vielmehr wird hier das Netzwerk des Widerstands auf Grundlage der Kurz-

biografien der GDW dargestellt. Weder diese kurzen Texte noch der darauf basierende Wissensgraph bilden die reale Größe oder Komplexität des Widerstands ab.

Die Leistung der vorliegenden Arbeit liegt in der aufgebauten Pipeline, die automatisch Informationen aus deutschsprachigen Texten extrahiert, strukturiert und in einen Wissensgraphen überführt. Dabei ist diese Pipeline durch ihren modularen Aufbau leicht verbesserbar und außerdem kaum an eine konkrete Domäne gebunden. Die einzige spezifisch an die Texte angepasste Methode sind die Triggerwörter in der Relation Extraction. Das heißt, dass perspektivisch auch andere historische oder literarische Beispiele mit dieser Pipeline verarbeitet werden können.

Literatur

[ABR21] Richard Aude, Sandra Bernstein und Jonas Richter. »Das Netzwerk des Widerstands Wissensrepräsentation in Graphen«. Arbeit auf Anfrage verfügbar. Feb. 2021.

[Akb+19] Alan Akbik u. a. »FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP«. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*. Minneapolis, Minnesota: Association for Computational Linguistics, Juni 2019, S. 54–59. DOI: 10.18653/v1/N19-4010. URL: https://www.aclweb.org/anthology/N19-4010 (besucht am 16. 06. 2021).

[Alt21] Christoph Benedikt Alt. »Neural sequential transfer learning for relation extraction«. en. In: (2021). Accepted: 2021-01-20T09:02:45Z. DOI: 10.14279/depositonce-11154. URL: https://depositonce.tu-berlin.de/handle/11303/12278 (besucht am 13. 01. 2022).

[Ben+14] Darina Benikova u. a. »GermEval 2014 Named Entity Recognition Shared Task«. In: 2014.

[Car10] Kai-Uwe Carstensen. »Anwendungen«. de. In: *Computerlinguistik und Sprachtechnologie: Eine Einführung*. Hrsg. von Kai-Uwe Carstensen u. a. Heidelberg: Spektrum Akademischer Verlag, 2010, S. 553–658. ISBN: 978-3-8274-2224-8. DOI: 10.1007/978-3-8274-2224-8_5. URL: https://doi.org/10.1007/978-3-8274-2224-8_5 (besucht am 18. 11. 2021).

[Che+17] Alessandro Checco u. a. »FashionBrain Project: A Vision for Understanding Europe's Fashion Data Universe«. en. In: *arXiv:1710.09788 [cs]* (Okt. 2017). arXiv: 1710.09788. URL: http://arxiv.org/abs/1710.09788 (besucht am 01. 12. 2021).

[EE10] Christian Ebert und Cornelia Ebert. »Methoden«. de. In: *Computerlinguistik und Sprachtechnologie: Eine Einführung*. Hrsg. von Kai-Uwe Carstensen u. a. Heidelberg: Spektrum Akademischer Verlag, 2010, S. 169–479. ISBN: 978-3-

8274-2224-8. DOI: 10.1007/978-3-8274-2224-8_3. URL: https://doi.org/10.1007/978-3-8274-2224-8_3 (besucht am 06. 01. 2022).

[Hac+13] Ben Hachey u. a. »Evaluating Entity Linking with Wikipedia«. en. In: *Artificial Intelligence*. Artificial Intelligence, Wikipedia and Semi-Structured Resources 194 (Jan. 2013), S. 130–150. ISSN: 0004-3702. DOI: 10.1016/j.artint.2012.04.005. URL: https://www.sciencedirect.com/science/article/pii/S0004370212000446 (besucht am 19. 11. 2021).

[Hog+11] F. Hogenboom u. a. »An Overview of Event Extraction from Text«. In: *DeRiVE@ISWC*. 2011.

[Hos19] Shammamah Hossain. »Visualization of Bioinformatics Data with Dash Bio«. In: *Proceedings of the 18th Python in Science Conference* (2019). Conference Name: Proceedings of the 18th Python in Science Conference, S. 126–133. DOI: 10.25080/Majora-7ddc1dd1-012. URL: http://conference.scipy.org/proceedings/scipy2019/shammamah_hossain.html (besucht am 02. 03. 2022).

[HSS08] Aric A Hagberg, Daniel A Schult und Pieter J Swart. »Exploring Network Structure, Dynamics, and Function using NetworkX«. en. In: (2008), S. 5.

[Hul+20] Johannes M. van Hulst u. a. »REL: An Entity Linker Standing on the Shoulders of Giants«. en. In: *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval* (Juli 2020). arXiv: 2006.01969, S. 2197–2200. DOI: 10.1145/3397271.3401416. URL: http://arxiv.org/abs/2006.01969 (besucht am 26. 05. 2021).

[Inc15] Plotly Technologies Inc. *Collaborative data science*. 2015. URL: https://plot.ly.

[Key18] Linda von Keyserlingk-Rehbein. *Nur eine "ganz kleine Clique?": die NS-Ermittlungen über das Netzwerk vom 20. Juli 1944*. Deutsche Erstausgabe, 1. Auflage. Gedenkstätte Deutscher Widerstand, Schriften der Gedenkstätte Deutscher Widerstand / A, Band 12. Lukas Verlag, 2018. ISBN: 978-3-86732-303-1.

[Lab+19] Kai Labusch u. a. »BERT for Named Entity Recognition in Contemporary and Historical German«. In: Okt. 2019.

[Lew94] Theodor Lewandowski. *Linguistisches Wörterbuch*. 6. Aufl., unveränd. Nachdr. d. 5., überarb. Aufl. UTB ; 1518. Quelle & Meyer, 1994. ISBN: 978-3-494-02173-7.

[Men+21] Sina Menzel u. a. »Named Entity Linking mit Wikidata und GND – Das Potenzial handkuratierter und strukturierter Datenquellen für die semantische Anreicherung von Volltexten«. de. In: *Qualität in der Inhaltserschließung*. De Gruyter Saur, Okt. 2021, S. 229–258. ISBN: 978-3-11-069159-7. DOI: 10.1515/9783110691597-012. URL: https://www.degruyter.com/document/doi/10.1515/9783110691597-012/html (besucht am 18. 01. 2022).

[MSO22] Gerhard Müller, Larissa Schmid und Felix Ostrowski. *SoNAR / Social Network Analysis and related Research Implementierungs- und Betriebskonzept*. de. Jan. 2022. URL: https://raw.githubusercontent.com/sonar-idh/reports/main/AP1-SBB-1-Implementierungskonzept-short.pdf (besucht am 17. 02. 2022).

[Ric18] Jonas Richter. »Das Gedenken an Anna Götze«. Arbeit auf Anfrage verfügbar. März 2018.

[RL18] Jonas Richter und Christoph Landgraf. »The first steps towards a digital memorial«. Arbeit auf Anfrage verfügbar. März 2018.

[Sch+99] Anne Schiller u. a. »Guidelines für das Tagging deutscher Textcorpora mit STTS«. In: *University of Stuttgart and Seminar für Sprachwissenschaft*. 1999.

[SK14] Rico Sennrich und Beat Kunz. »Zmorge: A German Morphological Lexicon Extracted from Wiktionary«. In: *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland: European Language Resources Association (ELRA), Mai 2014, S. 1063–1067. URL: http://www.lrec-conf.org/proceedings/lrec2014/pdf/116_Paper.pdf.

[SVS13] Rico Sennrich, Martin Volk und Gerold Schneider. »Exploiting Synergies Between Open Resources for German Dependency Parsing, POS-tagging, and Morphological Analysis«. In: *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*. Hissar, Bulgaria: INCOMA Ltd. Shoumen, BULGARIA, Sep. 2013, S. 601–609. URL: https://aclanthology.org/R13-1079 (besucht am 05. 07. 2021).

[TD03] Erik F. Tjong Kim Sang und Fien De Meulder. »Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition«. In: *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*. 2003, S. 142–147. URL: https://aclanthology.org/W03-0419.

[TM69] Jeffrey Travers und Stanley Milgram. »An Experimental Study of the Small World Problem«. en. In: *Sociometry* 32.4 (Dez. 1969), S. 425. ISSN: 00380431. DOI: 10.2307/2786545. URL: https://www.jstor.org/stable/2786545?origin=crossref (besucht am 05. 01. 2022).

[Tug16] Don Tuggener. »Incremental Coreference Resolution for German«. eng. Publication Title: Tuggener, Don. Incremental Coreference Resolution for German. 2016, University of Zurich, Philosophische Fakultät. Dissertation. University of Zurich, 2016. DOI: 10.5167/uzh-124915. URL: https://www.zora.uzh.ch/id/eprint/124915/ (besucht am 05. 07. 2021).

Abbildungsverzeichnis

4 Anhang

4.1 Graphstatistiken des *NDW*

Bei dem entstandenen Wissensgraph handelt es sich um einen nicht-zusammenhängenden Graphen mit 57 Komponenten. Die größte Komponente enthält 1205 Knoten, die restlichen 56 haben eine Größe zwischen 2 und 6. Insgesamt sind 1353 Knoten und 1862 Kanten enthalten. Der Knoten mit dem maximalen Grad ist «Berlin » mit einem Grad von 120. 615 Knoten haben einen Grad von 1. Zum Vergleich: Der entstandene Wissensgraph des Prototyp hat 6491 Knoten und 22338 Kanten. Hier war der Knoten mit maximalen Grad ebenfalls «Berlin » mit einem Grad von 331.

Auffällig ist hier die Diskrepanz der Knoten-/Kantenverhältnisse. Vermutlich ist diese auf den in Abschnitt 2.2.1 beschriebenen Ansatz, der nur aus Sätzen mit zwei oder mehr gefundenen Entitäten Relationen extrahiert, zurückzuführen. Zudem wurden, wie in Abschnitt 1.4 erwähnt, im Prototyp auch andere Substantive als Knoten behandelt und entsprechende Kanten gezogen.

Nachfolgend findet sich ein Ranking der 10 Knoten mit höchsten Grad, sowie Rankings abhängig vom jeweiligen NE-Tag.

Diese Rankings verdeutlichen anschaulich die Effizienz des nationalsozialistischen Repressionsapparats. So ist sehr Auffällig, dass mit Brandenburg an der Havel und Plötzensee zwei wichtige Gefängnisse der Nazis sehr hoch gerankt sind. Auch der Fakt, dass die Gestapo mit Abstand die höchstgerankte Organisation ist, fügt sich in dieses Bild ein. Ein Ranking der Kanten ergänzt diese Beobachtungen.

Trotzdem muss nochmals betont werden, dass sämtliche Ergebnisse stark von der verwendeten Quelle abhängen.

Rang	Grad	Knoten
1	120	Berlin
2	87	Plötzensee_Prison
3	74	Gestapo
4	68	Adolf_Hitler
5	66	Germany
6	42	Social_Democratic_Party_of_Germany
7	39	German_Communist_Party
8	33	Wehrmacht
9	23	Brandenburg_an_der_Havel
10	17	Weimar_Republic

Tabelle 4.1: Top 10 Knoten mit höchsten Grad

Rang	Grad	Knoten
1	120	Berlin
2	87	Plötzensee_Prison
5	66	Germany
9	23	Brandenburg_an_der_Havel
10	17	Weimar_Republic
11	17	Hamburg
12	17	Munich
14	15	United_States
15	15	Sachsenhausen_Concentration_camp
17	12	Switzerland

Tabelle 4.2: Top 10 Knoten mit höchsten Grad mit Tag „LOC“

Rang	Grad	Knoten
3	74	Gestapo
6	42	Social_Democratic_Party_of_Germany
7	39	German_Communist_Party
8	33	Wehrmacht
13	16	Nazi_Party
27	8	Young_Communist_League_of_Germany
28	8	Schutzstaffel
38	7	South_Australia[22]
68	6	Humboldt_University_of_Berlin
108	5	Free_Workers'_Union_of_Germany

Tabelle 4.3: Top 10 Knoten mit höchsten Grad mit Tag „ORG“

Rang	Grad	Knoten
4	68	Adolf_Hitler
16	15	Carl Friedrich Goerdeler
19	11	Bernhard Bästlein
23	10	Herbert Baum
24	10	Wilhelm Leuschner
26	9	Alice Licht
31	8	Theodor Leipart
32	8	Herbert Tschäpe
34	7	Ludwig Beck
39	7	John Graudenz

Tabelle 4.4: Top 10 Knoten mit höchsten Grad mit Tag „PER“

Rang	Grad	Kante
1	128	verurteilt
2	127	ermordet
3	107	Bekanntschaft
4	83	festgenommen
5	79	verhaftet
6	73	an
7	72	auf
8	72	arbeitet
9	60	tritt
10	58	gehört

Tabelle 4.5: Top 10 Kanten mit höchsten Grad

4.2 Stichproben


```
{
    "Name": "Alfred Kranzfelder",
    "Lebensdaten": "10. Februar 1908 - 10. August 1944",
    "Beschreibung": "Alfred Kranzfelder tritt 1927 in die
       Reichsmarine ein und dient in den folgenden Jahren zun
       ächst in Seeverbänden. Im Februar 1940 wird er aus
       Gesundheitsgründen nach Berlin versetzt und arbeitet
       in der Operationsabteilung der 1.Seekriegsleitung beim
        Oberkommando der Marine. Hier begegnet er dem
       Marineoberstabsrichter Berthold Schenk Graf von
       Stauffenberg, einem Bruder des späteren Attentäters
       Claus Schenk Graf von Stauffenberg. Kranzfelder
       verschafft sich als Verbindungsoffizier zum Auswä
       rtigen Amt einen Einblick in die Kriegslage und
       gewinnt in Gesprächen mit den Brüdern Stauffenberg
       bald die Überzeugung, dass der NS-Staat beseitigt
       werden müsse. Obwohl seine Möglichkeiten zur Unterstü
       tzung des Umsturzes nur begrenzt sind, stellt
       Kranzfelder sich für die Planungen zum Staatsstreich
```

```
        zur Verfügung. Unmittelbar nach dem gescheiterten
        Attentat vom 20.Juli 1944 nimmt die Gestapo Berthold
        von Stauffenberg fest. Den Verfolgern wird die
        Beteiligung von Kranzfelder kurz darauf bekannt. Er
        wird am 24.Juli 1944 verhaftet, zwei Wochen später vom
         Volksgerichtshof zum Tode verurteilt und am 10.August
         1944 in Berlin-Plötzensee ermordet.",
}
```

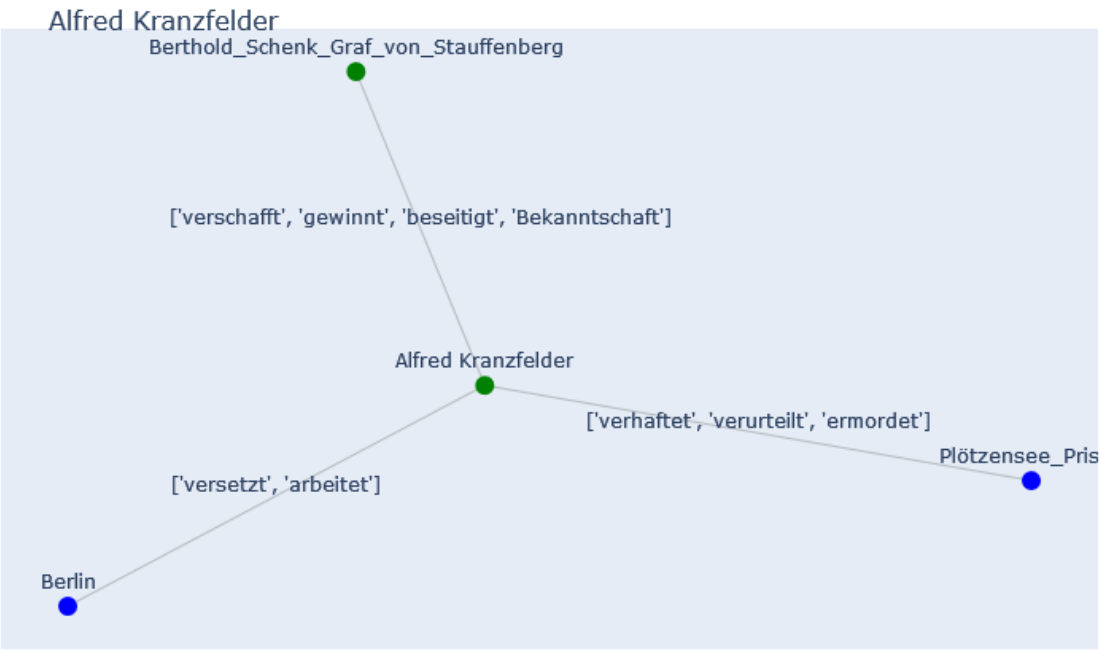

Abbildung 4.1: Resultierender Teilgraph von Alfred Kranzfelder

```
{
    "Name": "Helmuth James Graf von Moltke",
    "Lebensdaten": "11. März 1907 - 23. Januar 1945",
    "Beschreibung": "Helmuth James Graf von Moltke studiert
       seit 1925 in Berlin Rechts- und Staatswissenschaften.
       Engagiert leitet er in Schlesien ein freiwilliges
       Arbeitslager für Studenten, Bauern und
```

```
Industriearbeiter. Moltke, der den demokratischen Krä
ften seiner Zeit nahe steht, verfolgt Hitlers Aufstieg
 mit offener Kritik. Daher verzichtet er 1933 auf ein
Richteramt und lässt sich 1935 als Anwalt in Berlin
nieder. Zwischen 1935 und 1938 absolviert er zudem
eine Ausbildung als britischer Rechtsanwalt (Barrister
) und plant die Übernahme eines Anwaltsbüros in London
, die durch den Kriegsbeginn im September 1939
verhindert wird. Im selben Monat wird Moltke als
Kriegsverwaltungsrat in das Amt Ausland/Abwehr des
Oberkommandos der Wehrmacht in Berlin verpflichtet.
Als Sachverständiger für Kriegs- und Völkerrecht
versucht er, sich gegen Unrecht und Willkür
einzusetzen. Besonders engagiert er sich für die
humane Behandlung von Kriegsgefangenen und die
Einhaltung des Völkerrechts. Bereits 1939 verfasst
Moltke erste Denkschriften zur politischen
Neuorientierung Deutschlands. Anfang 1940 stößt Peter
Graf Yorck von Wartenburg zu einer Gruppe von
Regimegegnern um Moltke. Moltke und Yorck werden zu
den führenden Köpfen des daraus entstehenden Kreisauer
 Kreises und nehmen an den meisten der Beratungen in
Berlin und in Kreisau teil. Moltke versucht, durch
systematische Ausweitung seine Kontakte zu
protestantischen und katholischen Kirchenführern und
zu den Führern der politischen sozialdemokratischen
Opposition zu erweitern. Nachdem Moltke Mitglieder des
 Solf-Kreises vor einer Gestapo-Überwachung warnt und
dies entdeckt wird, wird er am 19.Januar 1944
verhaftet. Seine Beteiligung an den Staatsstreichplä
nen wird erst nach dem gescheiterten Umsturzversuch
vom 20.Juli 1944 bekannt. Am 11.Januar 1945 verurteilt
 der Volksgerichtshof ihn zum Tode. Helmuth James Graf
 von Moltke wird am 23.Januar 1945 in Berlin-Plö
tzensee ermordet.",
```

```
}
```

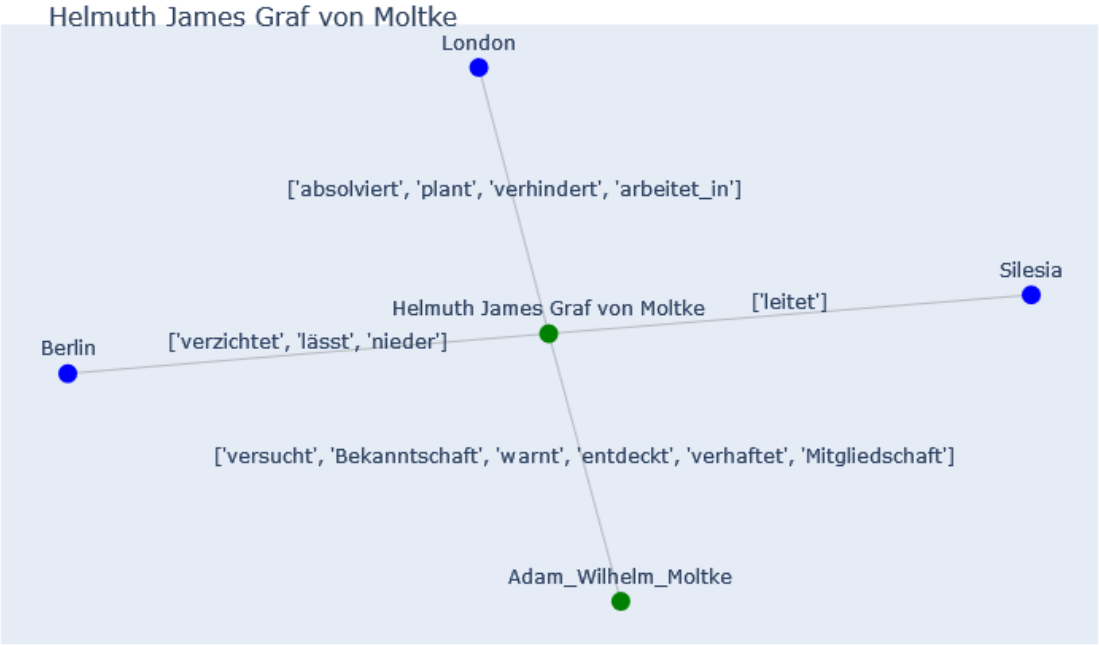

Abbildung 4.2: Resultierender Teilgraph von Helmuth James Graf von Moltke


```
{
    "Name": "Franz Kempner",
    "Lebensdaten": "19. Oktober 1879 - 05. März 1945",
    "Beschreibung": " Der Jurist Franz Kempner arbeitet ab
       1906 in der Kolonialverwaltung in Deutsch-Ostafrika
       und wird dort im Ersten Weltkrieg mehrfach verwundet.
       Nach seiner Rückkehr nach Deutschland ist er zunächst
       im Wiederaufbau-Ministerium tätig, bevor er 1922 in
       die Reichskanzlei eintritt, wo er von 1925 bis 1926
       das Amt des Staatssekretärs innehat. In den folgenden
       Jahren zieht er sich weitgehend ins Privatleben zurück
       . Franz Kempner, der während der Weimarer Republik
       Mitglied der Deutschen Volkspartei (DVP) ist,
       verbindet seit der gemeinsamen Tätigkeit in der
```

Reichskanzlei eine enge Freundschaft mit Erwin Planck. Wie dieser lehnt Kempner den Nationalsozialismus ab und hat Verbindung zu zahlreichen Persönlichkeiten des Widerstands. Er nimmt an einem Gesprächskreis oppositionell eingestellter ehemaliger Diplomaten, Beamter und Wissenschaftler im Hause von Johanna Solf teil, bei dem offen Kritik am nationalsozialistischen Regime geäußert wird. Zudem hat er Kontakt zum Kreis um Ulrich von Hassell, Johannes Popitz und Carl Goerdeler, den er 1943 kennenlernt und der ihn für das Amt des Staatssekretärs der Reichskanzlei nach einem gelungenen Umsturz zu gewinnen versucht. Nach dem 20. Juli 1944 wird er festgenommen und gesteht in Verhören der Gestapo seine Vorbehalte gegenüber der nationalsozialistischen Rassen- und Kirchenpolitik. Franz Kempner wird am 12.Januar 1945 vom Volksgerichtshof zum Tode verurteilt und am 5.März in Plötzensee ermordet.",
}

{
"Name": "Heinrich Graf zu Dohna-Schlobitten",
"Lebensdaten": "15. Oktober 1882 - 14. September 1944",
"Beschreibung": "Heinrich Graf zu Dohna verwaltet in der ersten Zeit des NS-Regimes die Familiengüter in Ostpreußen. Gleichzeitig setzt er sich als Mitglied des Ostpreußischen Bruderrates der Bekennenden Kirche für verhaftete Pfarrer ein. Er ist verheiratet mit Maria Agnes von Borcke, mit der er eine Tochter und drei Söhne hat. Als Reserveoffizier wird er 1939 zum Stellvertretenden Stabschef im Generalkommando in Kö nigsberg ernannt. Nach dem deutschen Überfall auf Polen in den aktiven Heeresdienst übernommen, gelingt es ihm 1943, auf eigenen Wunsch als Generalmajor wieder auszuscheiden, weil er die Kriegführung Hitlers

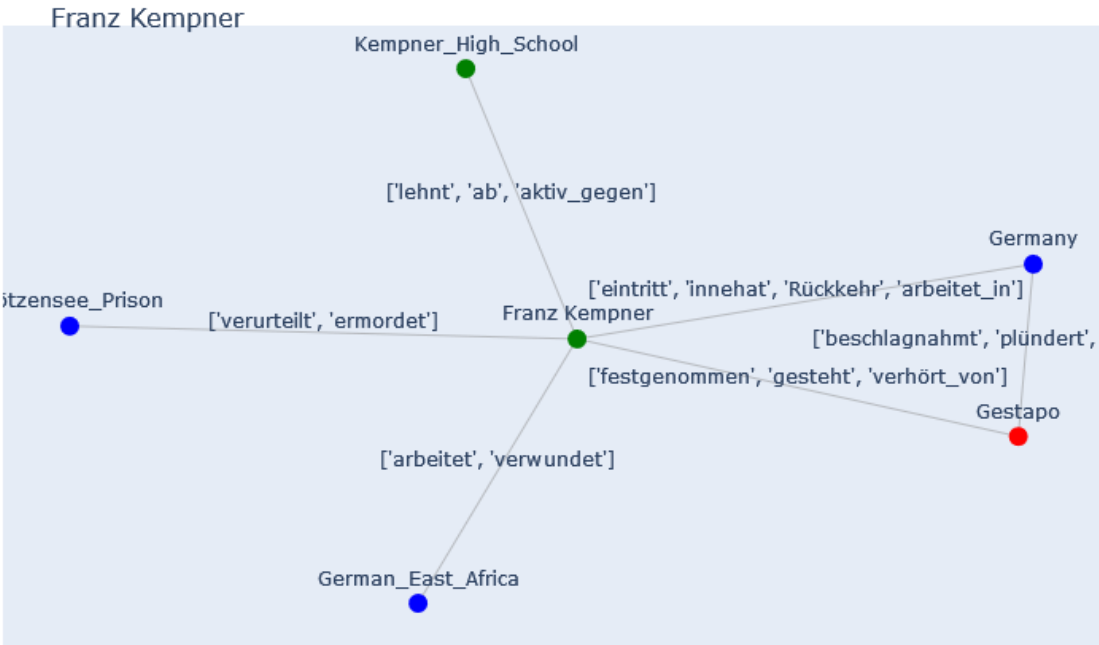

Abbildung 4.3: Resultierender Teilgraph von Franz Kempner

```
        nicht hinnehmen kann. Zusammen mit seiner Frau
        beteiligt er sich an der Umsturzplanung der
        Widerstandsgruppen um Ludwig Beck und Carl Goerdeler
        und stellt sich für den Wehrkreis I (Königsberg) als
        Politischer Beauftragter zur Verfügung. Nach dem
        Attentat auf Hitler werden er und seine Frau verhaftet
        . Während Agnes Gräfin zu Dohna die Haft im Frauen-KZ
        Ravensbrück überlebt, verurteilt der Volksgerichtshof
        Heinrich Graf zu Dohna am 14.September 1944 zum Tode.
        Noch am selben Tag wird er in Berlin-Plötzensee
        ermordet.",
}

{
    "Name": "Carlo Mierendorff",
    "Lebensdaten": "24. März 1897 - 04. Dezember 1943",
```

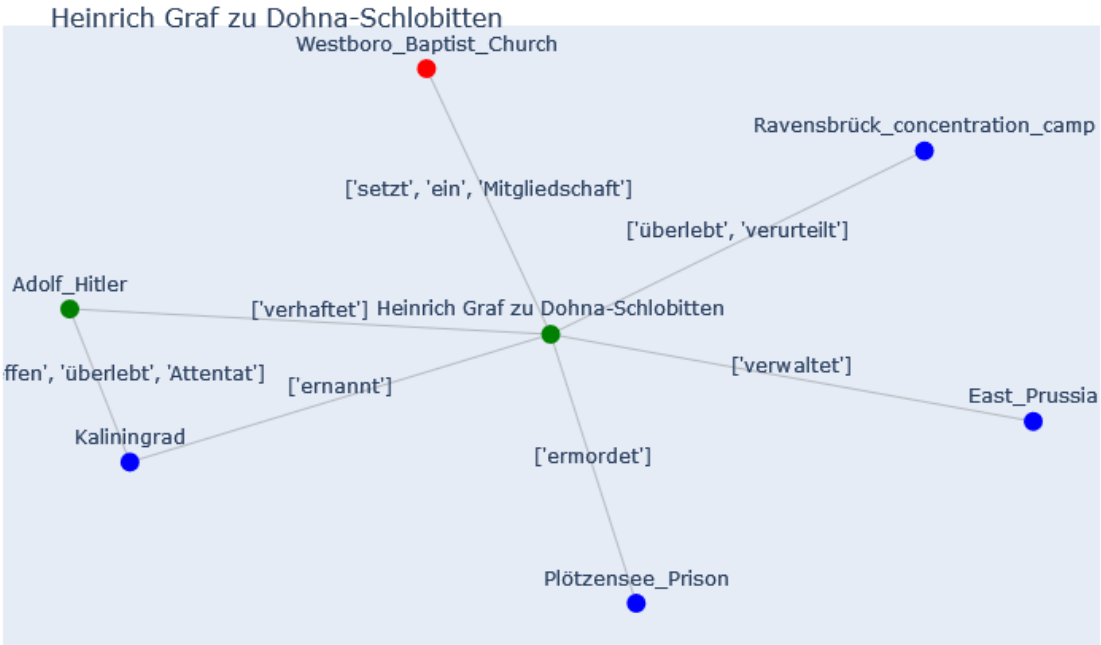

Abbildung 4.4: Resultierender Teilgraph von Heinrich Graf zu Dohna-Schlobitten


```
"Beschreibung": "Carlo Mierendorff nimmt am Ersten
   Weltkrieg als Freiwilliger teil und studiert von 1918
   bis 1922 Philosophie und Volkswirtschaft. 1923
   promoviert er in Heidelberg über die
   Wirtschaftspolitik der KPD. 1920 tritt er der SPD bei,
    wird als zuverlässiger und aktiver Sozialdemokrat
   1929 Pressechef und einer der engsten Mitarbeiter des
   hessischen Innenministers Wilhelm Leuschner. 1930 wird
    Mierendorff in den Reichstag gewählt. Carlo
   Mierendorff gilt vor 1933 als einer der erbittertsten
   Gegner der NSDAP und ihres Propagandisten Joseph
   Goebbels. Schon 1931 kann er den Blick der Ö
   ffentlichkeit auf die Boxheimer Dokumente lenken, die
   detaillierte Pläne für einen nationalsozialistischen
   Staatsstreich enthalten. Mierendorff kehrt 1933 trotz
   vieler Warnungen seiner Freunde von einer Reise aus
```

```
        der Schweiz zurück, wird verhaftet, misshandelt und
        bis 1938 in Konzentrationslagern gequält. Nach seiner
        Entlassung findet er erneut Verbindungen zu
        politischen Gesinnungsfreunden, so auch, wie sein
        Freund Theodor Haubach, zum Kreisauer Kreis, wo
        Mierendorff die sozialpolitische Diskussion
        entscheidend beeinflusst. Es gelingt ihm, die Gegensä
        tze katholischer und sozialistischer Anschauungen zu ü
        berbrücken. Im Juni 1943 verfasst er den Aufruf für
        eine Sozialistische Aktion als Sammlungsbewegung des
        Widerstands. Seine Pläne finden ein plötzliches Ende,
        als Carlo Mierendorff am 4.Dezember 1943 in Leipzig wä
        hrend eines alliierten Bombenangriffs ums Leben kommt
        .",
}
```

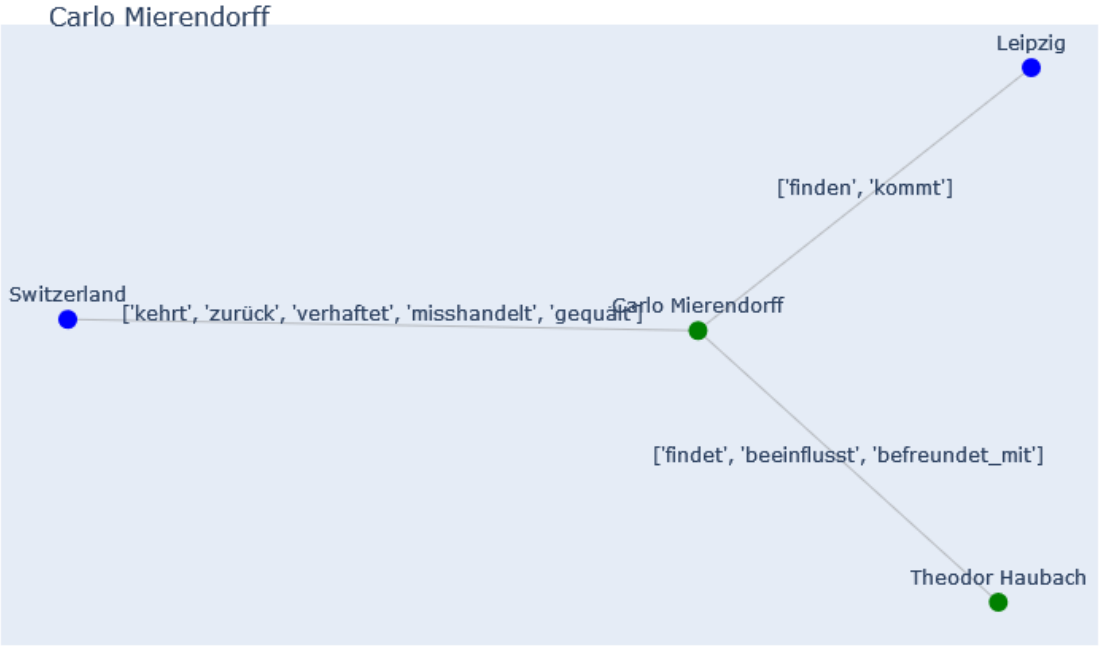

Abbildung 4.5: Resultierender Teilgraph von Carlo Mierendorff

```
{
    "Name": "Roland von Hößlin",
    "Lebensdaten": "21. Februar 1915 - 13. Oktober 1944",
    "Beschreibung": "Roland von Hößlin, der aus einer
       Augsburger Familie stammt, wandelt sich als junger
       Offizier unter dem Eindruck der NS-Kriegführung vom gl
       äubigen Anhänger zum entschiedenen Gegner Hitlers. Als
        Kommandeur einer Panzer-Aufklärungsabteilung des
       Afrikakorps wird er im Juli 1942 verwundet und mit dem
        Ritterkreuz ausgezeichnet. Hößlin, der als Major im
       17.Kavallerie-Regiment Bamberg Regimentskamerad
       Stauffenbergs ist, wird durch diesen im April 1944 in
       die Umsturzpläne der Verschwörer eingeweiht. Als
       Kommandeur der Panzer-Aufklärungs-Ersatz-Abteilung 24
       in Insterburg/Ostpreußen (Wehrkreis I) soll er im
       Falle eines erfolgreichen Umsturzes in Ostpreußen
       wichtige Gebäude besetzen und Maßnahmen gegen die
       NSDAP leiten. Hößlin wird am 23.August 1944 in
       Meiningen/Thüringen verhaftet und am 13.Oktober 1944
       nach seiner Entlassung aus der Wehrmacht vom
       Volksgerichtshof zum Tode verurteilt. Noch am selben
       Tag wird er in Berlin-Plötzensee ermordet.",
}

{
    "Name": "Friedrich Karl Klausing",
    "Lebensdaten": "24. Mai 1920 - 08. August 1944",
    "Beschreibung": "Friedrich Karl Klausing will
       Berufsoffizier werden und tritt im Herbst 1938 als
       Fahnenjunker in das angesehene Potsdamer
       Infanterieregiment 9 ein. Nach Beginn des Zweiten
       Weltkriegs wird er zunächst in Polen und Frankreich
       eingesetzt und nimmt im Winter 1942/43 an den Kämpfen
       bei Stalingrad teil. Dort wird er schwer verwundet und
        nach einer weiteren Verwundung 1943 zum Innendienst
```

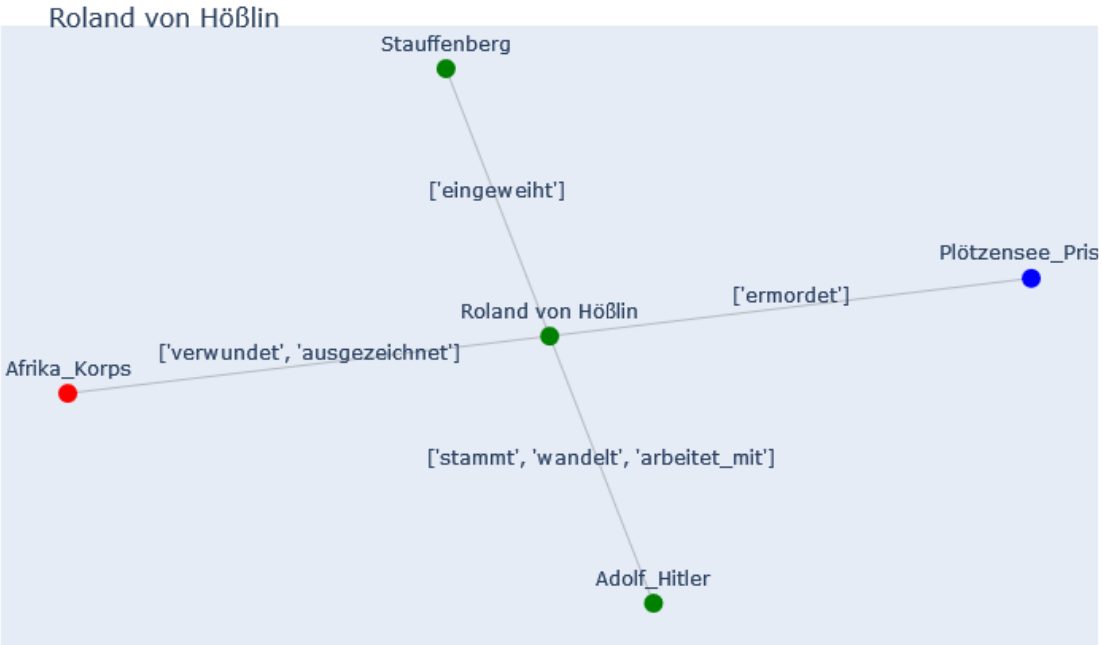

Abbildung 4.6: Resultierender Teilgraph von Roland von Hößlin

```
beim Oberkommando der Wehrmacht nach Berlin versetzt.
Fritz-Dietlof Graf von der Schulenburg kann ihn dort f
ür seine Verschwörungspläne gewinnen. Am 15.Juli 1944
begleitet Klausing Stauffenberg als Adjutant in das
ostpreußische Führerhauptquartier. Das zunächst für
diesen Tag geplante Attentat kann aber nicht ausgefü
hrt werden. Am 20.Juli 1944 hält er sich im
Bendlerblock, der Zentrale der Verschwörer in Berlin,
auf und ist dort für die Übermittlung der Walküre-
Befehle mitverantwortlich. In der Nacht vom 20. auf
den 21.Juli kann Klausing zunächst entkommen und sich
bei Freunden verstecken. Am nächsten Morgen stellt er
sich jedoch der Gestapo und wird im ersten
Schauprozess gegen die Verschwörer vom
Volksgerichtshof am 8.August 1944 zum Tode verurteilt.
 Friedrich Karl Klausing wird noch am selben Tag in
```

```
        Berlin-Plötzensee ermordet.",
}
```

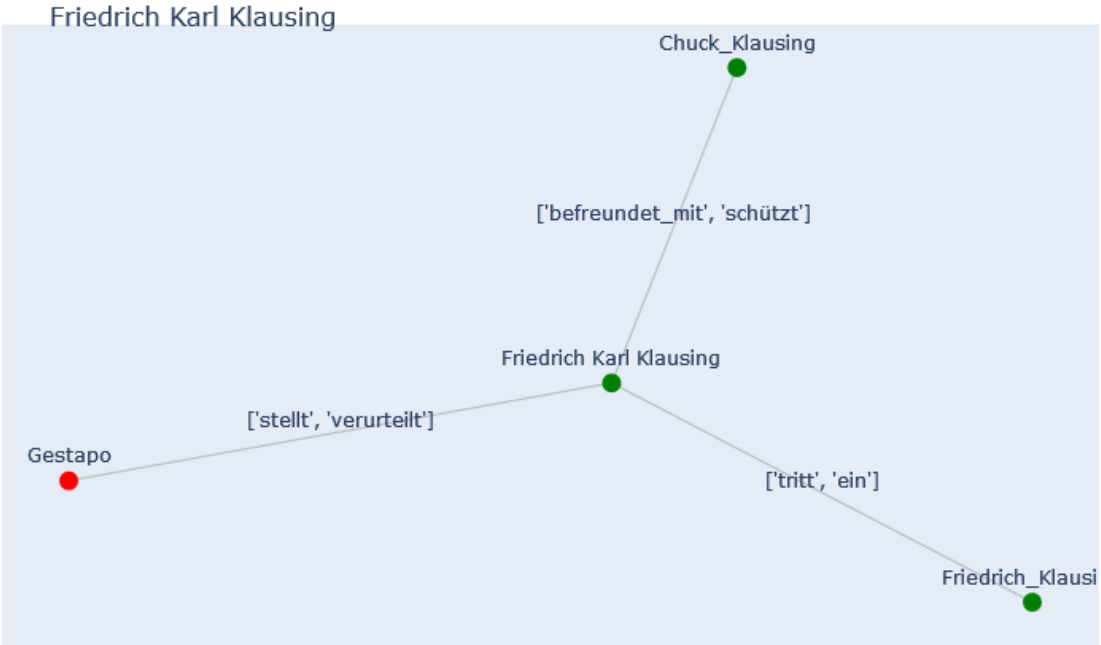

Abbildung 4.7: Resultierender Teilgraph von Friedrich Karl Klausing

```
{
    "Name": "Ewald von Kleist-Schmenzin",
    "Lebensdaten": "22. März 1890 - 09. April 1945",
    "Beschreibung": "Der Jurist und Gutsbesitzer Ewald von
       Kleist-Schmenzin steht der Deutschnationalen
       Volkspartei nahe und bekennt sich zu einem
       monarchistisch und christlich geprägten
       Konservativismus. In der Endphase der Weimarer
       Republik bekämpft er entschieden den
       Nationalsozialismus. Im Mai und Juni 1933 wird er
       zweimal verhaftet, nach kurzer Zeit aber wieder
       freigelassen. Zur Zeit des Münchener Abkommens reist
       Kleist-Schmenzin im Auftrag der Verschwörergruppe um
```

den damaligen Generalstabschef des Heeres Ludwig Beck nach London. In den Jahren 1942 und 1943 trifft er sich mit Carl Goerdeler und sagt seine Unterstützung f ür den Staatsstreich zu. Später wird Kleist-Schmenzin in die Pläne von Stauffenberg eingeweiht und billigt auch das Attentat, an dem sich sein Sohn Ewald Heinrich aktiv beteiligt. Er selbst ist für den Umsturz als Politischer Beauftragter für den Wehrkreis II (Stettin) vorgesehen. Kleist-Schmenzin wird am Tag nach dem gescheiterten Attentat vom 20.Juli festgenommen, am 15.März 1945 vom Volksgerichtshof zum Tode verurteilt und am 9.April 1945 in Berlin-Plö tzensee ermordet. Das Ermittlungsverfahren gegen seinen Sohn Ewald Heinrich, der sich am 20.Juli als Ordonnanzoffizier im Bendlerblock bei den Verschwörern aufhält, wird am 12.Dezember 1944 eingestellt. Er kann, an die Front versetzt, überleben.",
}

{
"Name": "Franz Leuninger",
"Lebensdaten": "28. Dezember 1898 - 01. März 1945",
"Beschreibung": "Franz Leuninger erlernt zunächst das Maurerhandwerk und leitet in der Weimarer Republik verschiedene Bezirksorganisationen der christlichen Gewerkschaften. Er ist verheiratet mit Anna Paulina Meuser, mit der er drei Söhne hat. 1930 zieht er für die Zentrumspartei ins Breslauer Stadtparlament ein und kandidiert im März 1933 auch für den Reichstag. Leuninger ist ein entschiedener Gegner des Nationalsozialismus. Nach der Zerschlagung der Gewerkschaften im Frühsommer 1933 übernimmt er die Geschäftsführung der gemeinnützigen Siedlungsgesellschaft Deutsches Heim. Auf diese Weise kann er die Verbindungen zu Verfolgten und Gegnern des

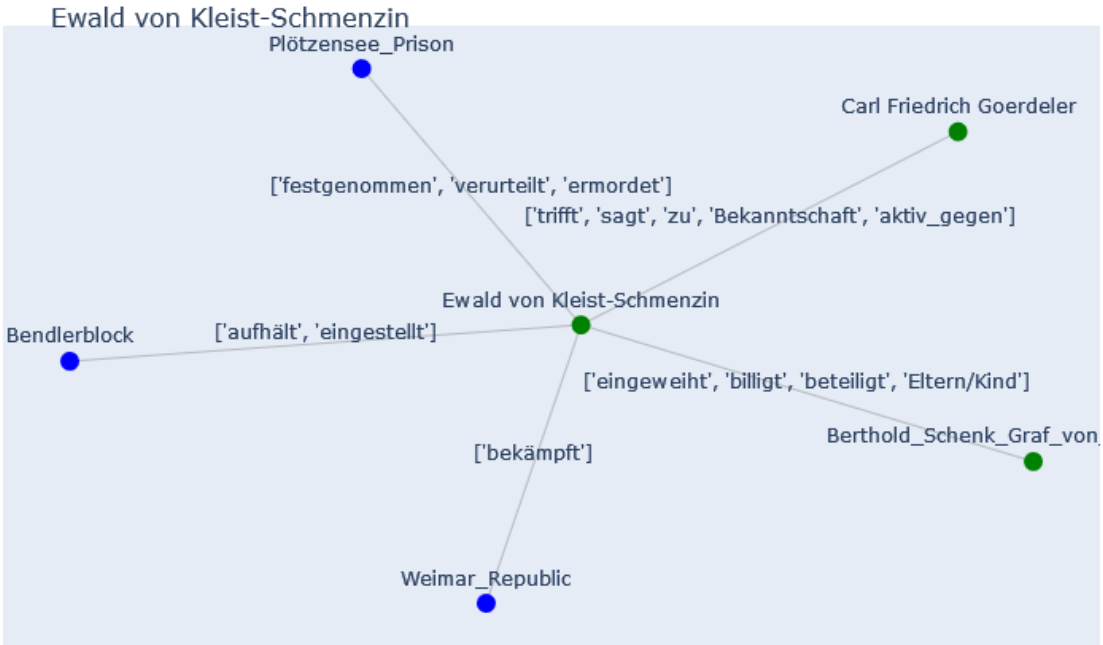

Abbildung 4.8: Resultierender Teilgraph von Ewald von Kleist-Schmenzin

```
 NS-Regimes aufrechterhalten und kommt später in
Kontakt mit den Widerstandsgruppen um Carl Goerdeler
und Ludwig Beck. Die Verschwörer gewinnen Leuninger fü
r das Amt des Oberpräsidenten der Provinz Schlesien.
Nach dem Attentat vom 20.Juli 1944 wird er am 26.
September 1944 festgenommen und für mehrere Monate im
Berliner Zellengefängnis Lehrter Straße inhaftiert, am
 26.Februar 1945 vom Volksgerichtshof zum Tode
verurteilt und am 1.März 1945 in Berlin-Plötzensee
ermordet.",
}

{
    "Name": "Adolf-Friedrich Graf von Schack",
    "Lebensdaten": "03. August 1888 - 15. Januar 1945",
    "Beschreibung": "Nach dem Abitur am humanistischen
```

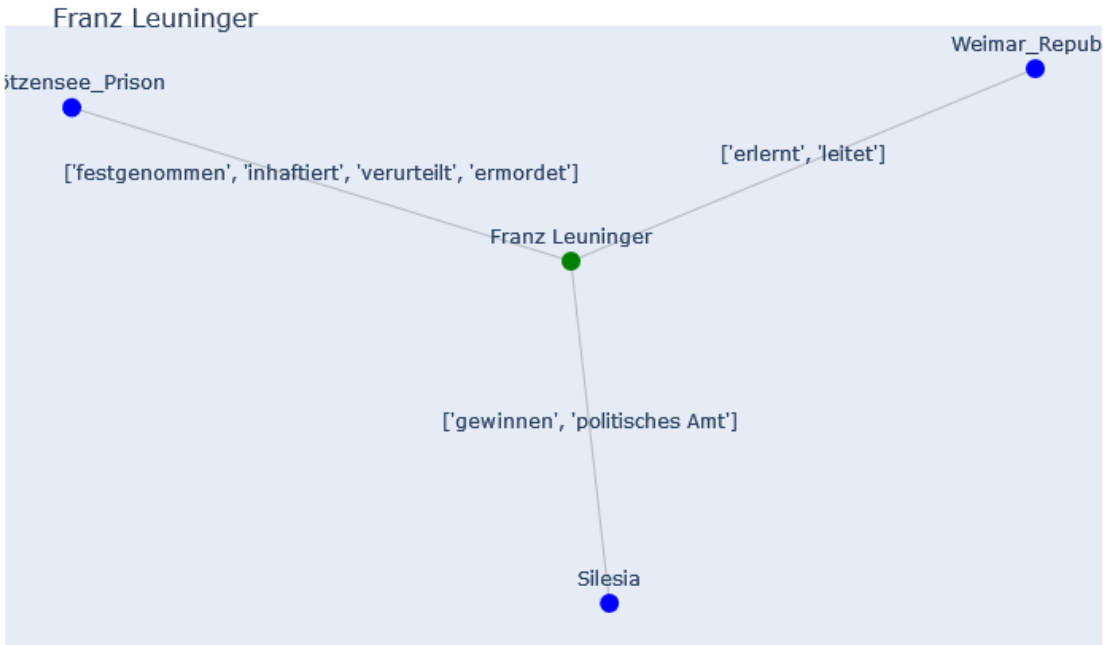

Abbildung 4.9: Resultierender Teilgraph von Franz Leuninger

```
Gymnasium in Boppard am Rhein tritt Adolf-Friedrich
Graf von Schack 1909 in das 2.Garderegiment zu Fuß ein
. Er nimmt als Offizier am Ersten Weltkrieg teil und
wird schwer verwundet. Weitere Fronteinsätze folgen
bis zur Verabschiedung 1920. 1928 heiratet er Else
Dorothea Freiin von Werthern. Aus der Ehe stammen drei
 Söhne und eine Tochter. Ab Januar 1944 ist Graf
Schack in der Berliner Stadtkommandantur als Leiter
der Organisationsabteilung unter Generalleutnant Paul
von Hase tätig. Schack arbeitet am 20.Juli 1944 eng
mit den Verschwörern zusammen. Als deutlich wird, dass
 der Umsturzversuch scheitert, gelingt es von Schack,
belastende Papiere zu vernichten. Adolf-Friedrich Graf
 von Schack wird am 21.Juli 1944 festgenommen und am
10.Oktober 1944 vom Volksgerichtshof zum Tode
verurteilt. Bis zu seiner Erschießung am 15.Januar
```

```
        1945 in Brandenburg-Görden wird er im Gefängnis Tegel
        festgehalten.",
}
```

Abbildung 4.10: Resultierender Teilgraph von Adolf-Friedrich Graf von Schack

```
{
    "Name": "Hermann Kaiser",
    "Lebensdaten": "31. Mai 1885 - 23. Januar 1945",
    "Beschreibung": "Obwohl der Gymnasiallehrer Hermann
       Kaiser früh der NSDAP beitritt, wendet er sich später
       vom Nationalsozialismus ab. Zu Beginn des Krieges wird
        er als Reserveoffizier eingezogen und 1940 zum
       Oberkommando des Heeres versetzt, wo er als Hauptmann
       die Führung des Kriegstagebuchs beim Stab des
       Befehlshabers des Ersatzheeres, Friedrich Fromm, ü
       bernimmt. Hier findet er Anschluss an die militärische
        Opposition um Ludwig Beck und Carl Goerdeler. In
```

```
        seinen privaten Aufzeichnungen überliefert Kaiser
        viele Begegnungen und Gespräche zwischen den
        Verschworenen. Er erklärt sich bereit, nach dem
        Umsturz das Amt eines Staatssekretärs im
        Kultusministerium zu übernehmen. Außerdem ist er als
        Verbindungsoffizier im Wehrkreis XII (Wiesbaden)
        vorgesehen. Nach dem Scheitern des Attentats vom 20.
        Juli 1944 wird Kaiser mit seinen beiden Brüdern
        verhaftet, seine Tagebücher beschlagnahmt und von der
        Gestapo als wichtige Quelle über die Verschwörung
        ausgewertet. Hermann Kaiser wird am 17.Januar 1945 vom
         Volksgerichtshof zum Tode verurteilt und bald darauf
        in Berlin-Plötzensee ermordet.",
}
```

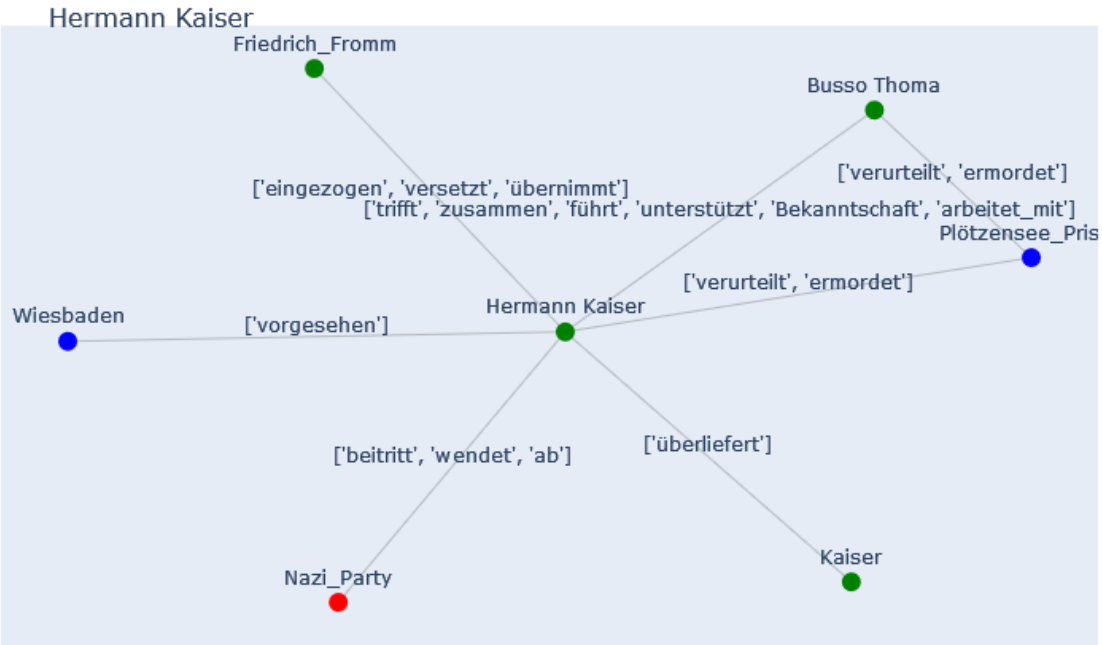

Abbildung 4.11: Resultierender Teilgraph von Hermann Kaiser

Eigenständigkeitserklärung

Ich versichere, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe, insbesondere sind wörtliche oder sinngemäße Zitate als solche gekennzeichnet. Mir ist bekannt, dass Zuwiderhandlung auch nachträglich zur Aberkennung des Abschlusses führen kann