Masterarbeit

Semantische Personensuche auf Hochschulwebsites

Johannes Güttler

04. August 2017

Albert-Ludwigs-Universität Freiburg im Breisgau
Technische Fakultät
Institut für Informatik

Bearbeitungszeitraum
10. 04. 2017 – 04. 08. 2017

Betreuerin
Prof. Dr. Hannah Bast

Gutachter
Prof. Dr. Hannah Bast
Prof. Dr. Peter Fischer

Erklärung

Hiermit erkläre ich, dass ich diese Abschlussarbeit selbständig verfasst habe, keine anderen als die angegebenen Quellen/Hilfsmittel verwendet habe und alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten Schriften entnommen wurden, als solche kenntlich gemacht habe. Darüber hinaus erkläre ich, dass diese Abschlussarbeit nicht, auch nicht auszugsweise, bereits für eine andere Prüfung angefertigt wurde.

Inhaltsverzeichnis

Zusammenfassung

Die Arbeit zeigt wie Personendaten aus einem Webcrawl extrahiert und als Grundlage für eine Suchmaschine aufbereitet werden können. Es wird vorgestellt, wie sich die Texte deutscher Hochschulwebsites aus einem vorhandenen Webcrawl extrahieren lassen. Dazu wird die ungeordnete Menge der Daten mehrerer CommonCrawl-Webcrawls verarbeitet. Mittels Named Entity Recognition werden Personennamen extrahiert und in eine Ontologie überführt. Diese wird mit zusätzlichen Informationen wie zum Beispiel der persönlichen Homepages angereichert.
Die Ontologie und extrahierten Websitetexte werden für die Suchmaschine Broccoli aufbereitet. Sie lassen sich beschränkt auf geografische Gebiete und zusätzlicher Volltextsuche in den Websitetexten durchsuchen. Die Qualität der Ergebnisse variiert stark. Sie ist nicht nur von der Art der Suchanfrage abhängig, sondern auch stark von der Darstellung von Personen auf der entsprechenden Hochschulwebsite.

1 Einleitung

1.1 Motivation

Viele Projekte der letzten Jahre fördern und nutzen die Idee des Semantic Web. Einige Suchmaschinen binden semantische Daten in ihre Ergebnisse ein und große umfassende Wissensdatenbanken entstehen. Auch die semantische Personensuche wird verstärkt bearbeitet. Für die konkrete Suche nach Wissenschaftlern existieren wenige Lösungen. Soweit dem Autor bekannt bietet nur die Suchmaschine AMiner eine umfassende Wissenschaftlersuche. Die hier durchsuchbaren Personennamen basieren auf Publikationsverzeichnissen und sind auf Personen beschränkt für die eine Homepage ermittelt werden kann. Eine Wissenschaftlersuche, die auf Daten der Universitäten und Hochschulen beruht, existiert nach Wissen des Autors nicht. Dies gilt auch für eine Bundesland-orientierte Suche in Deutschland und die Möglichkeit, die Suche mit einer Volltextsuche zu verknüpfen. Ein direktes Durchsuchen der Websitetexte mit Volltextsuche würde die Expertensuche nicht auf die bereitgestellten Themen oder Stichwörter beschränken.

1.2 Ziele

Auf Grundlage der Websites deutscher Universitäten und anderer Hochschulen soll eine semantische Personensuche ermöglicht werden. Die Personen sollen ihrer jeweiligen Hochschule zugeordnet werden und der Datensatz so auch über geografische Suchanfragen, also nach Bundesländern oder Städten, durchsucht werden können. Soweit möglich sollen die Personen auf ihre persönliche Seite auf der Universitätswebsite verlinkt werden.
Neben den semantischen Verknüpfungen sollten die Texte in denen eine Person genannt wird im Zusammenhang mit der Person mittels Volltextsuche verfügbar sein. Dafür müssen die Daten für die Suchmaschine Broccoli aufbereitet werden.

1.3 Aufbau der Arbeit

Im folgenden Abschnitt werden verwandte Arbeiten vorgestellt. Kapitel 2 behandelt einige Grundbegriffe die im Verlauf der Arbeit verwendet werden.

Kapitel 3 beschreibt in welcher Form Daten, vor allem auf Hochschulwebsites, vorliegen und wie diese aus einem Webcrawl extrahiert werden können. Verfahren zur Extraktion von Personen und weiteren Information aus dem Datensatz werden in Kapitel 4 vorgestellt. Kapitel 5 beschreibt die Implementierung der Informationsextraktion. Diese wird in Kapitel 6 evaluiert. Schlussendlich wird die Evaluation in Kapitel 7 bewertet und Anstöße für zukünftige Arbeiten gegeben.

1.4 Verwandte Arbeiten

Tang et al. stellen in [TZY07] ein System vor, um anhand von gegebenen Namen von Wissenschaftlerinnen und Wissenschaftlern persönliche Profile der Personen zu extrahieren. Genutzt werden dafür persönliche oder Organisationswebsites auf denen die Personen vorgestellt werden. Als Voraussetzung dient die Feststellung, dass knapp 71% aus einer Testmenge von fast 450000 Personen eine solche Website zugeordnet werden kann.
Als Ausgangsdaten wird die DBLP Datenbank der Universität Trier genutzt, in der Publikationen aus dem Bereich Informatik gesammelt werden. Anhand der Autorenliste wird eine Suchanfrage über die Google API dahingehend ausgewertet, welches der ersten Ergebnisse am Wahrscheinlichsten eine Seite ist, auf der die gesuchte Person beschrieben wird. Die Website wird in einzelne Tokens zerlegt, die dann als Daten, wie E-Mail-Adresse etc., klassifiziert werden sollen. Tokens sind definierte „Spezialwörter“ wie “<image>“, die über reguläre Ausdrücke gefunden werden. Zur Klassifizierung wird jedes Wort mit einem oder mehreren Tags versehen. Das Tagging wird über ein Conditional Random Field gesteuert. Das heißt Tags werden bedingt von den umgebenden Wörtern vergeben.
Der zweite Teil der Arbeit beschäftigt sich mit der Namensdisambiguierung (Zwei Personen mit gleichem Namen). Auch hier wird mit Hidden Markov Random Fields ein probabilistischer Ansatz verwendet. Über vordefinierte Constraints wie „Autoren gehören der gleichen Organisation an“, „gleiche E-Mail-Adresse“, Zitationen und andere wird die Distanz zwischen Personen und Veröffentlichungen und Personen mit gleichem Namen berechnet.
Zur Auswertung wird eine Menge von 1000 Personennamen untersucht. Es werden nur Personen berücksichtigt, für die die Methode eine Homepage identifiziert. Dies sind im Testdatensatz 898. Erreicht wird mit dieser Methode ein durchschnittlicher F1-Wert von 83.37% unter Auswertung der kompletten Personenprofile.
Die Ergebnisse aus [TZY07] werden in [TZY+08] genutzt um ein „soziales Netzwerk“ zu erstellen, das nicht nur verschiedene Wissenschaftlerinnen und Wissenschaftler mit ihren Fachgebieten beinhalten soll, sondern auch Verknüpfungen zu einem Netzwerk wie die Beziehung zwischen (Co-) Autoren in gemeinsamen Veröffentlichungen. Die gewonnenen Daten können mit der Suchmaschine AMiner für Expertensuche oder geografische Wissenschaftlersuche genutzt werden.

In [AGMU03] wird ein Verfahren vorgestellt, Informationen aus Webseiten zu extrahieren, das ohne Trainingsdaten auskommt. Ziel ist es, Daten in ähnlich aufgebauten Seiten, wie sie zum Beispiel bei Onlineshops vorkommen wieder zu erkennen. Dazu werden Seitentemplates gesucht, die alles enthalten sollen, das nicht zu den zu extrahierenden Daten gehört. Dazu gehören zum Beispiel häufig (oder immer) vorkommende Schlagwörter wie „Preis“ oder „Beschreibung“, jedoch auch anderer Text, der sich von Seite zu Seite unterscheiden kann. Übrig bleiben dann nur die zu extrahierenden Daten.
Die zwei Hauptschwierigkeiten dieses Vorgehens geben die Autoren wie folgt an: Erstens die Erkennung verschachtelter Informationen, die selbst bei ähnlich aufgebauten Seiten schwer zu erkennen sind. Zum anderen die Kategorisierung eines Textabschnittes (Wortes) als Template oder als Datum. Mit dem im Weiteren vorgestellten Framework erzielen die Autoren sehr gute Ergebnisse für immer gleich aufgebaute Seiten, wie Onlineshops. Sobald die einzelnen Seiten stärker voneinander abweichen werden die Ergebnisse allerdings schlechter. Wie in Kapitel 3 beschrieben weichen die Formate von Personenseiten auf Universitätswebsites auch innerhalb einer Universität so stark voneinander ab, dass dieses Verfahren sich für die wenigsten in dieser Arbeit gesuchten Seiten anwenden lassen würde.

Auf die Extraktion von Personennamen aus Nachrichtentexten wird in [BT08] eingegangen. Vorgestellt wird die Extraktion von Personen aus türkischen Nachrichtentexten. Da den Autoren keine türkische Grammatik zur Verfügung stand wird im ersten Teil der Arbeit auf das Finden von Verben der indirekten Rede eingegangen. Anhand typischer Vorkommen in der türkischen Grammatik werden die Verben in vier Gruppen eingeteilt. Die Satzstellungen werden dann in Pattern übersetzt, anhand derer Personen erkannt werden sollen. (Beispiel: „`[Title]? + [W*]? + [PN] + [W*]? + [, | (<E>ise) | (<E>de) | (<E>da)] + [W*] + [RVF1] + [.]`“, mit `[PN]` = Personenname und `[RVF1]` = personenbezogenes Verb). Es wird so eine Präzision von 78,13 (Recall:86,91; F-Maß: 81,97) erreicht.

In [PM08] wird ein Verfahren vorgestellt, um Namensteile schon extrahierter Personen als Vor- oder Nachnamen zu erkennen. Dabei haben die Autoren den Anspruch, das System für beliebig-sprachige Texte und Namen einsetzen zu können. Jeder Namensteil wird einer der drei Kategorien *part*, *abbrv* oder *token* zugeordnet, wobei unter *part* Namenszusätze (von, mc, du, al, ...) verstanden werden. Aus den verschiedenen möglichen Kombinationen lassen sich dann die Namen extrahieren die entweder der Kategorie „Vorname“ oder „Kein Vorname“ zugeordnet werden sollen. Im ersten Schritt werden pro Name / Token verschiedene (Wahrscheinlichkeits-)Werte berechnet: Die generelle Wahrscheinlichkeit, dass der Name als Nachname auftritt, ein Maß für das Auftreten mit umgebenden Wörtern, ein Wahrscheinlichkeitsmaß für die lexikalische Zugehörigkeit (zum Beispiel häufige Suffixe bei Nachnamen), die Wahrscheinlichkeit, dass an dieser Position des Namens ein Nachname steht und die wahrscheinliche Anordnung von Vor- und Nachname. Da keine entsprechend vollständigen Listen zur Verfügung stehen werden für unbekannte Tokens jeweils

Google-Suchanfragen in Kombination mit den häufigsten 20 Vor- und Nachnamen durchgeführt, aus denen die oben genannten Werte berechnet werden. Es wird damit eine Genauigkeit von 90% erreicht, wobei in diesem Fall ausschließlich italienische Nachrichtenseiten verwendet werden.

2 Grundlagen

2.1 Semantische Suche

Die semantische Suche unterscheidet sich von der Volltext- oder Stichwortsuche, die von gängigen Suchmaschienen bekannt ist. Grundlage der Volltextsuche ist, dass nach Wörtern oder Wortkombinationen in Texten beziehungsweise Dokumenten gesucht wird. Zu diesem Zweck wird ein Suchindex über die Wörter oder auch Wortteile erstellt. Wortteile der Länge n werden n-gram genannt und schließen alle zusammenhängenden Abschnitte eines Wortes der Länge n ein.
In der semantischen Suche wird nach Entitäten gesucht, die gegebenenfalls über definierte Relationen miteinander in Verbindung stehen. Entitäten sind in Klassen und Unterklassen organisiert, eine zu definierende Ontologie bildet die Relationen beziehungsweise Eigenschaften der Entitäten ab. Das Finden dieser Semantischen Informationen und Zusammenhänge stellt die Schwierigkeit beim Erstellen eines solchen Suchindexes dar.

2.2 Ontologien und Semantic Web

Ontologie bedeutet in diesem Zusammenhang die Darstellung von Wissen mit Hilfe von Klassen, Entitäten und Beziehungen (Relationen) zwischen diesen Klassen. Entitäten sind Instanzen der Klassen. Beispielsweise ordnet die Struktur „Alan Turing *is-a* Person“ die Entität Alan Turing der Klasse Person zu.
Zur Erstellung von Ontologien können verschiedene Daten herangezogen werden. Die einfachste Möglichkeit ist das Generieren anhand von Tabellen beziehungsweise Datenbanken, in denen die relevanten Informationen schon vorhanden sind. Zum Beispiel eine Liste deutscher Hochschulen, in der nicht nur der Hochschulname, sondern auch Adresse, Website und weitere Informationen enthalten sind. Tatsächlich sind solche strukturierten Daten oft nicht verfügbar. Auch auf Webseiten finden sich in der Regel Informationen in Form von natürlichsprachigem Text [Bas13].
Davon abgesehen existieren verschiedene Ansätze auch diese Daten in strukturierter und maschinell auswertbarer Form darzustellen. Da reines HTML in dieser Hinsicht wenige Möglichkeiten bietet, Webseiten aber ohnehin im HTML-Format vorliegen bietet sich an dieses zu erweitern. Dazu gibt es zwei etablierte Ansätze. Die sogenannten Microformats (μF) integrieren Metadaten in HTML-Attribute. Die zu benutzenden Vokabeln sind dabei strikt vorgegeben und es können keine URIs zur

Entitätenidentifikation verwendet werden [Hau09].
Das Resource Description Framework (RDF) wurde von einer Arbeitsgruppe (RDF Working Group) des W3C entwickelt und bietet die Möglichkeit Metadaten in strukturierter Form in HTML (oder anderswo) einzubetten. Dabei werden im wesentlichen Tripel der Form Subjekt - Prädikat - Objekt verwendet, die einen gerichteten Graphen bilden und als Ontologie interpretiert werden können. Zur eindeutigen Identifikation werden URIs verwendet, wobei Subjekt und Prädikat immer URIs sind und das Objekt ein URI oder ein Literal sein kann. Ist kein eindeutiger Identifier bekannt können Subjekt und Objekt auch anonyme Knoten sein [Hau09]. RDF kann mit der *Web Ontology Language (OWL)* erweitert werden. Es lassen sich dann auch kompliziertere Beziehungen wie transitive Relationen ausdrücken oder Restriktionen definieren.

2.3 Natürliche Sprachverarbeitung

Da Websites in den wenigsten Fällen semantische Verknüpfung ihrer Inhalte bereitstellen müssen gesuchte Entitäten und weitergehende Informationen zunächst im natürlichsprachigen Text erkannt werden. Für diese Verarbeitung von natürlichsprachigem Text (***N**atural **L**anguage **P**rocessing*) wird jeder Satz in seine grammatikalische Struktur zerlegt. Es werden Phrasen und Bausteine der Form Subjekt - Verb - Objekt extrahiert [SNG17]. Die Erkennung von Entitäten kommt oft auch ohne die Definition der kompletten Satzstruktur aus. Obwohl verschiedene Methoden existieren sind probabilistische Ansätze am weitesten verbreitet. Anhand eines händisch erzeugten Ausgangs-Datensatzes wird die wahrscheinlichste Satz-Zerlegung generiert. Funktionsweisen werden in Kapitel 4 vorgestellt.

Stanford Core NLP

Das Stanford Core NLP Toolkit [MSB+14] ist eine Kombination verschiedener Bausteine zur Verarbeitung natürlichsprachiger Texte. Ziel des seit 2006 entwickelten Frameworks ist es, eine einfach zu benutzende Schnittstelle zu allen enthaltenen Verarbeitungstools zu erhalten. Das Framework ist komplett in Java geschrieben und ist über eine Java API zugänglich. Über die Möglichkeit einen Server bereitzustellen können auch beliebige POST-Anfragen verarbeitet werden. Multithreading wird nativ unterstützt, verteilte Hardware nicht. Mittlerweile existieren Wrapper für verschiedene Sprachen.
Zur Steuerung der Funktionalität dient ein Satz von Annotatoren: tokenize, cleanxml, ssplit, truecase, pos, lemma, gender, ner, regexner, parse, sentiment, dcroef. Der „Tokenizer“ (*tokenize*) liefert die logisch zusammenhängenden Einheiten des Satzes. Die Named Entity Recognition (*ner*) erkennt neben Personen-, Orts- und Organisationsnamen auch numerische Entitäten wie Geldbeträge oder Daten. Die Komponente des Part-of-speech Taggers (*pos*) liefert die Wortart jedes Wortes und Zeichens.

Neben weiteren Tools wird mit *cleanxml* zum Beispiel auch die Möglichkeit geboten XML-Tags aus Texten zu entfernen. Alle dieser Annotatoren werden nur für englische Texte unterstützt. Weitere Sprachpakete mit eingeschränktem Umfang existieren für Arabisch, Chinesisch, Französisch und Deutsch.

2.4 Suchmaschine Broccoli

Broccoli ist eine an der Universität Freiburg entwickelte semantische Suchmaschine [BBBH12]. Sie kombiniert die semantische Suche auf Ontologien mit Volltextsuche, so dass die Ergebnismenge beispielsweise zunächst auf eine Objektklasse eingeschränkt werden und dann innerhalb dieser Menge nach Schlüsselwörtern gesucht werden kann.
In der Vorverarbeitung wird dafür der Rohdatensatz nicht nur indiziert, wie es bei einer klassischen Volltextsuche der Fall wäre. Zusätzlich wird eine Entitätserkennung ausgeführt, die nicht nur den reinen Entitätennamen, sondern auch verknüpfte weitere Nennungen erkennt (zum Beispiel eine Person bei Nennung des Nachnamens, oder einem Bezug mit Personalpronomen). Die Dokumente beziehungsweise Texte werden weiterhin in sogenannte Kontexte unterteilt, die jeweils einem zusammenhängenden Satzteil entsprechen. Daraus wird ein erweiterter Index, ähnlich einem Inverted Index erstellt. Dieser enthält neben den relevanten Kontexten des Index-Eintrages auch die Entitäten, die diesem Kontext zugeordnet sind. Weiterhin wird eine Ontologie erstellt, die jede Entität Klassen zuordnet oder über Relationen mit Eigenschaften verknüpft.
Suchanfragen bilden ausgehend von einem Wurzelknoten eine Baumstruktur mit einem oder mehreren Ästen. Klassen und Entitäten bilden dabei die Knoten, die über Realtionen aus der Ontologie verbunden sind. Die Volltext-Komponente wird über die Relation *occurs-with* eingebunden.

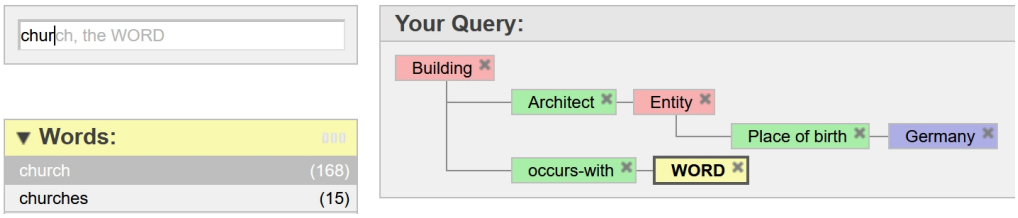

Abbildung 2.1: Kombination von sematischer- und Volltextsuche in Broccoli

Der in [BBBH12] beschriebene Prototyp der Suchmaschine verarbeitet Daten der englischsprachigen Wikipedia. Entsprechende Aufbereitung der Daten vorausgesetzt lässt sich der Index aber für beliebige Daten und Ontologien konstruieren. Die Aufbereitung der Daten in dieser Arbeit ist in Abschnitt 5.2 beschrieben. Abbildung 2.1 zeigt eine Suchanfrage in Broccoli. Gesucht werden Kirchen, die von Architekten aus

Deutschland entworfen wurden. Der Suchbaum führt hier über zwei Stufen. Gesucht werden alle Entitäten vom Typ „Person“, welche die Relation „Place of birth“ zur Entität Deutschland besitzen. Da keine konkrete Relation für Gebäudearten besteht wird weiterhin nur nach Gebäuden gesucht, die zusammen mit dem Stichwort „church“ in Texten vorkommen.

3 Daten

Obwohl die Idee des Semantic Web immer weitere Verbreitung findet [BLO13] bieten die wenigsten Websites semantische Informationen an. Die semantisch verarbeitbaren Informationen beschränken sich so oft auf HTML-Tags, aus denen sich aber nur begrenzt Inhalte extrahieren lassen.
In diesem Kapitel wird zunächst die Idee der semantischen und verknüpften Daten beschrieben. In weiteren Abschnitten folgt ein Überblick, in welcher Form personenbezogene Daten von Hochschulpersonal auf den jeweiligen Websites vorliegt und wie diese Daten in Rohform extrahiert werden können.

3.1 Personendaten

Auf der Suche nach Wissenschaftlerinnen und Wissenschaftlern kommen verschiedene Quellen in Frage. Über einige wurden Wikipedia-Artikel veröffentlicht, teilweise existieren Listen, wenn bestimmte Fachgebiete oder Spezialisierungen gesucht werden. Um eine aktuelle und aktualisierbare Übersicht zu erhalten führt der Weg an den Websites der Universitäten nicht vorbei. Diese überlassen das Bereitstellen von Personendaten beziehungsweise Vorstellen einzelner Mitarbeiter oft den Fakultäten oder Lehrstühlen, was zu einem breiten Angebot unterschiedlichster Darstellungsformen führt.
Einzelne Hochschulen stellen aktuelle Listen des wissenschaftlichen Personals bereit, die allerdings nicht standardisiert sind. Zur Nutzung dieser Listen lässt sich ein manuelles Auswerten der URLs und Strukturen der einzelnen Listen nicht vermeiden.

3.1.1 Daten auf Universitätswebsites

Betrachtet man die Websites der, gemessen an der Zahl der Studierenden, zehn größten Universitäten Deutschlands[1,2] fällt vor allem auf, dass sich selbst innerhalb

[1] FernUniversität in Hagen, Universität zu Köln, Ludwig-Maximilians-Universität München, Johann Wolfgang Goethe-Universität Frankfurt am Main, Rheinisch-Westfälische Technische Hochschule Aachen, Westfälische Wilhelms-Universität Münster, Ruhr-Universität Bochum, Universität Duisburg-Essen, Universität Hamburg, Friedrich-Alexander-Universität Erlangen-Nürnberg.[Hoc]

[2] Stichprobenartige Überprüfung verschiedener Fakultäten aller genannten Hochschulen, Stand Mitte Dezember 2016)

von Fakultäten die Repräsentation von Personendaten sehr unterschiedlich gestaltet. Während bei manchen Universitäten auf fakultäts- oder sogar universitätsweite einheitliche Darstellung geachtet wird, finden sich anderswo auch innerhalb von Fakultäten eigene und vor allem verschiedene Designs. Eine automatische Erkennung von Seitentemplates für jede Hochschule, wie in [AGMU03] beschrieben, wird so unmöglich gemacht, auch weil viele der Seiten die Informationen nur in ausformulierter Textform enthalten.
Gleiches gilt für die URLs: Während auf der Seite mit der URL „.../mitarbeiter/prof.dr.max_mustermann“ in der Regel ein Professor vorgestellt wird, verrät „uni-xy.de/2938/index.htm“ diese Information nicht. Die Problematik dahinter ist in Unterabschnitt 3.2.1 genauer beschrieben.

Auf den Personenseiten finden sich meist grundlegende Daten wie Titel, Namen und Fachgebiet der Person. Ob und in welcher Form weitere Daten vorhanden sind unterscheidet sich wiederum von Person zu Person. So stellen einige Mitarbeiterinnen und Mitarbeiter ihre kompletten Lebensläufe, Publikationen und Informationen zu ihren spezialisierten Forschungsgebieten bereit, bei anderen erfährt man nur den Namen. Das Forschungsgebiet muss dann anhand des Lehrstuhls oder anderen Kontextinformationen erschlossen werden. Lebensläufe und andere Informationen sind gegebenenfalls nur verlinkt, was bedeutet, dass unter der betrachteten URL nicht alle gewünschten Informationen zu finden sind. Das Verfolgen von Links zu Lebenslauf, Beschreibung der Forschungsgebiete, usw. wäre - sofern die Zieladressen im Korpus vorhanden sind - bei Nutzung der Metadaten aus den WAT-Dateien (siehe Abschnitt 3.2) möglich.

Davon unabhängig betreibt zum Beispiel die Universität Frankfurt ein Qualitätsinformationssystem (QIS), über das sich Mitarbeiterdaten fakultätsübergreifend in Tabellenform abrufen lassen. Über das Feld „Experte für “ lassen sich dann auch Mitarbeiter des gesuchten Gebietes finden. (Eine semantische Aufarbeitung der Antworten nach außen erfolgt hier aber nicht, der Benutzer erhält eine HTML-Antwort, bei der nur wenige Tags aussagekräftige IDs besitzen.)

3.2 CommonCrawl

CommonCrawl[Com16] ist eine Non-Profit-Organisation, die seit 2011 einen eigenen Webcrawler betreibt und den Datensatz öffentlich zur Verfügung stellt. Die Ergebnisse einzelner Crawls lassen sich als Rohdaten (Format: Web ARChive, WARC), Daten inclusive der Metadaten wie HTTP-Headern und Links (WAT) oder reine Textinhalte (WET) abrufen. Für jede URL wird ein Eintrag (*Record*) mit den entsprechenden Metadaten angelegt und eine variierende Anzahl von Einträgen in einem *Segment* gespeichert. Eine Sortierung nach Domain findet dabei nicht statt.
Alle URLs sind zudem über den Common Crawl Index durchsuchbar. Es existiert

hierzu eine API, über die sich URLs, beziehungsweise ganze Domains abfragen lassen. Die Ergebnisse liefern im JSON-Format für jede URL den Dateipfad des Segments, das die Daten der URL enthält und weitere Informationen wie Abrufdatum, Länge des Eintrages und Position im Segment. Die Crawls werden von Amazon Webservices bereitgestellt und sind öffentlich abrufbar.
Für die Universität Frankfurt werden hier beispielsweise im Crawl 12/2016 fast 200 000 verschiedene URLs gefunden, für die Uni Freiburg sind es nur knapp 12 000.

Die 193 306 Seiten (URLs) der Universität Frankfurt[3] sind durch die chaotische Struktur des Crawls auf 191 780 verschiedene Segmente verteilt (Uni Freiburg: 11683 Seiten in 11 016 Segmenten). Manche Hochschulen, wie die Universität Frankfurt verlinken Einträge ihrer Bibliotheken oder Bereitstellungen von digitalisierten Schriften und Bildern auf eigene Seiten, wodurch in diesem Fall Seiten der Universitätsbibliothek ca. 98% der URLs ausmachen. Bei der Universität Freiburg sind es umgekehrt nur 2%.

3.2.1 URL-Auswertung

In einem naiven Ansatz kann man davon ausgehen, dass Personen wie Professorinnen und Professoren und andere wissenschaftliche Mitarbeiterinnen und Mitarbeiter der Lehrstühle eine eigene Seite auf der jeweiligen Lehrstuhl-Homepage besitzen und in der Regel einen Titel wie „Prof.“, „Dr.“, oder „Ph.D.“ tragen. Dann genügt es die indizierten URLs der verschiedenen Hochschulen auszuwerten und nur diejenigen Texte tatsächlich herunterzuladen, in deren URL ein solcher Identifier vorkommt. Diese beiden Kriterien werden auch in [TZY+08] als einer von mehreren Klassifikatoren verwendet.
Wie in Unterabschnitt 3.1.1 beschrieben funktioniert dieser Ansatz nur sehr begrenzt, da bei vielen Hochschulen die Namen und Titel nicht in der URL auftauchen. Außerdem sollen auch Personen gefunden werden, für die keine persönliche Seite existiert. Aus diesen Gründen muss der gesamte Inhalt der Websites mit einbezogen werden.

3.3 Datengewinnung

3.3.1 Rohdaten

Wie in Abschnitt 3.2 beschrieben müssen die gesamten textuellen Seiteninhalte für jede URL heruntergeladen werden. Die Größe eines CommonCrawl-Segments ist variabel. Bei WET-Dateien liegt sie - in komprimierter Form - im Durchschnitt bei ca

[3] Anzahl der indizierten Seiten im Crawl vom Dezember 2016. Darunter können sich Formularantworten und URLs befinden, die sich nur durch verschiedene GET-Variablen unterscheiden.

130 bis 150 MB. In dem oben genannten Crawl 12/2016 werden für die 401 deutschen Hochschulen [Hoc] insgesamt 1 413 481 URLs gefunden, deren Seiteninhalt in 91 506 Segmenten gespeichert ist. Insgesamt fallen hier also ca. 12 TB an Daten an. Wird die Suche auf die drei vorherigen Crawls (August, September und Oktober 2016) ausgeweitet werden für 389 Hochschulen insgesamt 1 656 065 URLs gefunden. Diese verteilen sich auf 206 925 Segmente mit einer Gesamtgröße von ca. 27,6 TB. Gleichlautende URLs werden dabei jeweils durch die neuere Version ersetzt.
Die mehr als doppelt so große Zahl an URLs bei Verwendung mehrerer Crawls deutet darauf hin, dass in unterschiedlichen Crawl-Vorgängen unterschiedlichen Verlinkungen gefolgt wird. Es werden so bei jedem Lauf auch Seiten erreicht, die bei vorherigen oder nachfolgenden Durchgängen aufgrund fehlender direkter Verlinkung oder Einschränkungen in der Suchtiefe nicht erreicht wurden. Obwohl die Gefahr besteht veraltete Daten mit einzubeziehen sollten deshalb mehrere Crawls betrachtet werden.
Auffällig ist, dass selbst unter Einbeziehung von vier Durchläufen 12 Hochschulen nicht in den Ergebnissen auftauchen. Bei neun handelt es sich dabei um kleinere Hochschulen mit etwa 30 bis zu etwa 430 Studierenden. Drei der Hochschulen sind mit etwa 800-1100 Studierenden etwas größer. Bei zwei davon ist in der verwendeten Hochschulliste eine falsche Toplevel-Domain angegeben, bei einer eine falsche URL. Für die richtigen Seiten finden sich jeweils Einträge. Auch für die neun kleineren Hochschulen wird bei acht jeweils eine (meist verkürzte[4]) Web-Adresse verwendet, wodurch für die angegebene Second-Level-Domain im CommonCrawl Index keine Einträge gefunden werden. Eine Hochschule wurde von einer in den Ergebnissen vorhandenen Hochschule übernommen. Die korrigierten URLs werden entsprechend zur Datengewinnung miteinbezogen.

3.3.2 Verfügbarkeit und Streaming

Die Segmente können bei Amazon Public Datasets heruntergeladen werden. Dort liegen die Segmente im Gzip-Format vor. Da theoretisch nicht die gesamte Datei gelesen werden müsste läge es nahe den in den Index-Einträgen mitgelieferten Offset zu nutzen, um nur die relevanten Abschnitte der Segmente zu lesen. Dies wird durch die Eigenschaften des Gzip-Formates verhindert, das in der aktuellen Version keinen wahlfreien Zugriff ermöglicht. Dadurch muss die gesamte Datei heruntergeladen oder gestreamt werden.
Da in jedem Segment nur wenige Einträge relevant sind genügt es die Dateien zum Bearbeitungszeitpunkt zu laden und nur die relevanten Einträge zu speichern. Da es selbst bei guter Datenverbindung nicht sinnvoll ist die Daten herunterzuladen müssen die Dateien lokal an ihrem Speicherort ausgewertet werden. Dies ist über Amazon Web Services (AWS) möglich, das einen (pseudo-) direkten Zugriff auf die

[4] Ein prominentes Beispiel für eine verkürzte URL ist die Universität Erlangen-Nürnberg, die in der verwendeten Liste der Hochschulrektorenkonferenz mit *https://www.uni-erlangen.de* gelistet ist. Mittlerweile wird aber ausschließlich die Adresse *https://www.fau.de/* verwendet. Diese Domain wurde allerdings schon vor der Auswertung korrigiert.

Daten ermöglicht.

3.3.3 MapReduce

MapReduce ist ein von Google Inc. entwickeltes Programmiermodell zur parallelen Verarbeitung großer Datenmengen [DG08]. Die Implementierung automatisiert dabei die Parallelisierung der Teilschritte auf beliebig großen Rechenclustern und behandelt dabei auch auftretende Fehler in einzelnen Knoten der Cluster.

Konzept

Während das Framework die Parallelisierung und das Fehlermanagement übernimmt muss der Nutzer vor allem zwei Funktionen implementieren: *map* und *reduce*. Ziel ist, aus einer Menge von *key/value*-Paaren als Eingabe wieder eine Menge von *key/value*-Paaren als Ausgabe zu generieren.
Die *map*-Funktion erhält die Eingabe-Paare $(k1, v1)$ und verarbeitet diese zu Zwischenergebnissen in der gleichen Form: $(k2, v2)$. Die Paare werden nach Schlüsseln sortiert und als Paare aus Schlüssel und Wertliste $(k2, list(v2))$ an die *reduce*-Funktion weitergereicht, wo aus der Wertliste für jeden Schlüssel in der Regel ein (oder kein) Ausgabewert produziert wird.

$$\begin{aligned} \text{map: } & (k1, v1) && \rightarrow && (k2, list(v2)) \\ \text{reduce: } & (k2, list(v2)) && \rightarrow && (k3, list(v3)) \end{aligned}$$

Es können beliebige Datentypen als Schlüssel und Werte genutzt werden. Da es sich hier in der Regel um große Datenmengen handelt werden die Wertlisten jeweils nicht im Speicher gehalten, sondern als Iterator bereitgestellt.

Die einzelnen Schritte werden verteilt auf einem Rechencluster ausgeführt. Während die meisten Knoten dieses Clusters einfache „Arbeiter“-Knoten sind gibt es einen „Master“-Knoten, der die Koordinierung der Schritte übernimmt. Die benutzerspezifische Implementierung der *map*- und *reduce*-Funktion wird dabei auf alle Knoten gestreut. Das Framework (beziehungsweise der Master) regelt dabei die Zuweisung der Datenpakete an die Knoten und Wiederholung von fehlgeschlagenen Schritten (*jobs*). Abbildung 3.1 zeigt den Ablauf eines MapReduce-Vorgangs. Die Ausgangsdaten werden - sofern sie nicht schon als kleine Dateien vorliegen - in etwa gleich große Dateien aufgeteilt. Im original MapReduce-Paper [DG08] werden 16 bis 64 MB vorgeschlagen. Je nach Aufgabe und Größe der Input-Menge können die Eingabedaten-Teile aber auch deutlich größer sein. Die Knoten bekommen vom Master ein Datenpaket zugewiesen und führen den *map*-Schritt aus. Die Ergebnisse, einfache Schlüssel-Wert-Paare, werden gespeichert. Dabei findet ein in der Grafik unsichtbarer Schritt, der „Shuffle“-Schritt statt. Die Daten werden anhand der Schlüssel sortiert, so dass jeder Schlüssel nur noch einmal vorhanden ist. Über

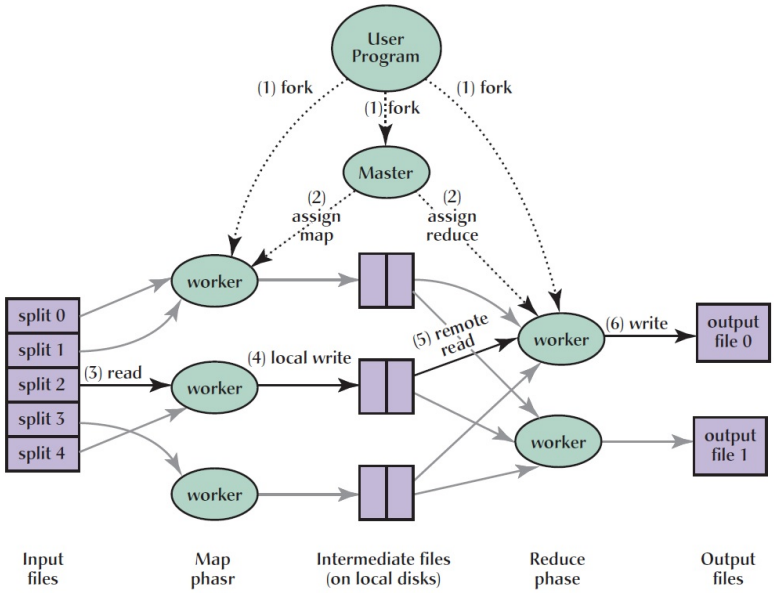

Abbildung 3.1: MapReduce Pipeline[DG08]

diesen Schlüssel lässt sich dann auf einen Iterator zugreifen, der alle Werte enthält die von *map*-Vorgängen mit diesem Schlüssel assoziiert wurden.
Die Schlüssel-Iterator-Paare werden dann vom Master-Knoten auf die Knoten verteilt, die den *reduce*-Schritt ausführen. Hier kommen die gleichen Knoten zum Einsatz, die schon den vorhergehenden Schritt ausgeführt haben. Die geteilte Darstellung in der Grafik dient nur der Übersichtlichkeit. Die Zwischenergebnisse werden insofern reduziert, dass für jeden Schlüssel nur noch ein Wert gespeichert wird.

Es ist möglich das Konzept zu variieren indem für Ein- und Ausgabewerte nur Schlüssel genutzt werden. Als Eingabemenge für die *map*-Funktion kann auch der Schlüssel weggelassen und nur eine Liste aus Werten als Eingabemenge dienen, wie es in Abschnitt 3.3.3 beschrieben wird. Werden als Ausgabe der *map*-Funktion allerdings nur die Schlüsselwerte genutzt (zusammen mit Dummy-Werten, die in der *reduce*-Funktion nicht gelesen werden) könnte, bei Eliminierung mehrfach vorkommender Schlüssel, auch „einfaches“ verteiltes Rechnen angewandt werden. Gegebenenfalls lohnt es sich trotzdem, eine MapReduce-Implementierung zu nutzen, da das Framework dem Nutzer viele Aufgaben abnimmt.

Implementierung

Es existieren verschiedene Implementierungen des MapReduce-Verfahrens. Die wohl bekannteste ist Apache Hadoop, das als freie Software vorliegt und sich auf den bei AWS buchbaren Rechen-Instanzen verwenden lässt. Außerdem arbeitet Hadoop auch mit dem S3-Dateisystem („Simple Storage Service“), das für die Public Datasets genutzt wird. Mittels der Python-Bibliothek mrJob („MapReduce-Job“) lassen sich

Einstellungen für Cluster setzen und auf AWS-Recheninstanzen ausführen.

Mapper Bei der Verarbeitung von CommonCrawl-Daten wird hier die *map*-Funktion genutzt, um aus den gewonnenen Indexdaten die Segmente herauszufiltern, die Hochschul-Daten enthalten. Als Eingabe dient jeweils eine Zeile aus den gesammelten Indexdaten. Dies entspricht einem JSON-Objekt, das neben der URL auch den Dateipfad zu demjenigen Segment enthält, in dem die unter der URL extrahierten Daten gespeichert sind. Der enthaltene Offset kann wie in Unterabschnitt 3.3.2 beschrieben nicht (direkt) genutzt werden. Daher besteht das Schlüssel-Wert-Paar der *Mapper*-Ausgabe aus Segment-Pfad und URL des Eintrages.

Reducer Aus dem Schlüssel ergibt sich der Pfad des Segments. Da Wahlfreier Zugriff nicht möglich ist wird die Bibliothek GzipStream verwendet. Diese erlaubt es Gzip-Dateien als Stream einzulesen und wurde im Speziellen für WARC-Dateien entwickelt. Die Header Einträge der hier verwendeten WET-Dateien haben das selbe Format wie in WARC-Dateien. Der Unterschied liegt im Umfang der Header und dem Inhalt des Datenabschnittes. Anhand der im Header aufgeführten URL wird bei relevanten Einträgen ein URL-Text Paar gebildet.

Zusammengefasst sehen die *map*- und *reduce*-Schritte also folgendermaßen aus:

$$\begin{aligned} \text{map: } & (\_, \mathit{segment\_name}) && \rightarrow && (\mathit{segment\_name}, \mathit{list}(\mathit{url})) \\ \text{reduce: } & (\mathit{segment\_name}, \mathit{list}(\mathit{url})) && \rightarrow && (\mathit{url}, \mathit{list}(\mathit{website\_text\_der\_url})) \end{aligned}$$

3.3.4 Umsetzung

AWS bietet verschiedene Möglichkeiten Rechen-Instanzen und -Cluster zu generieren. So ist es möglich ein Rechencluster im gleichen Rechenzentrum zu generieren, in dem auch die Amazon Public Datasets wie CommonCrawl vorgehalten werden. Dafür werden virtuelle Recheninstanzen, sogenannte EC2-Instanzen („Elastic Cloud Compute“) genutzt.
Da der Masterknoten „nur“ koordinierende Funktion hat kann für diesen eine weniger rechenstarke Instanz genutzt werden als für die Arbeiter-Knoten. Für diese Arbeit wurden eine sogenannte M3 large-Instanz für den Masterknoten und 24 M3 xlarge-Instanzen für die Arbeiter verwendet:

Typ	vCPU	RAM	Speicher (GB)	Physischer Prozessor	Takt (GHz)
m3.large	2	7,5	1 x 32 SSD	Intel Xeon E5-2670 v2	2,5
m3.xlarge	4	15	2 x 40 SSD	Intel Xeon E5-2670 v2	2,5

Es wurden ca. 545 000 Segmente als Eingabemenge genutzt. Der Durchlauf dauert in oben genannter Konfiguration knapp 37 Stunden.

Anmerkung: Genau genommen ist bei dieser Konfiguration der *map*-Schritt sehr klein. Die wesentliche Aufgabe liegt im verteilten Lesen der Segmente. Die Wahl von MapReduce beruht also vor allem darauf, dass es die einfachste Variante ist auf Amazon-Systemen viele Eingaben verteilt zu verarbeiten.

4 Informationsextraktion

Informationsextraktion bedeutet in diesem Zusammenhang das Finden von Entitäten und Relationen zwischen Entitäten. Die Informationsextraktion lässt sich daher in die beiden Problemstellungen der Entitätserkennung und der Erkennung von Relationen gliedern [Hä13]. Da in dieser Arbeit nur wenige und klar definierte Relationen verwendet werden, konzentriert sich das Kapitel auf die Entitätserkennung und Modelle zur Repräsentation natürlicher Sprache und ihr zugeordneten Wortarten. Insbesondere wird das Modell beschrieben das der Stanford Named Entity Recognition zugrunde liegt. Die Stanford NER wird in der Umsetzung (Kapitel 5) zur Entitätserkennung genutzt.

4.1 Natürliche Sprachverarbeitung

Die natürliche Sprachverarbeitung (*Natural language processing, NLP*) beschäftigt sich mit der Analyse natürlichsprachiger Texte. Ziel ist es Textbestandteile zu benennen oder zu extrahieren.
Beim sogenannten **Part-of-speech-Tagging** (*POS-Tagging*) sollen die einzelnen Satzteile als grammatikalische Komponenten benannt werden. Für jedes Wort wird anhand des Wortes selbst und des Kontextes die Wortart bestimmt. POS-Tagging basiert in der Regel auf statistischen Methoden auf Grundlage manuell erstellter Trainingsdaten. Für verschiedene Sprachen werden Tagsets verwendet, die die Eigenschaften der Sprache widerspiegeln. Im Deutschen wird häufig das STTS (Stuttgart-Tübingen-Tagset) verwendet. Es enthält elf Haupt-Wortarten, die in insgesamt 54 Tags unterteilt sind. Beispiele sind NN für Nomen, NE für Eigennamen, ADJA für attributive Adjektive oder ADV für Adverbien. Beispiel 4.1 zeigt einen Satz der so getaggt wurde.

Beispiel 4.1 Satz 9476 aus dem TIGER-Datensatz[1]

Der	Freiburger	Wissenschaftler	ist	einer	der	deutschen
ART	ADJA	NN	VAFIN	PIS	ART	ADJA

Pioniere	auf	diesem	Gebiet	.
NN	APPR	PDAT	NN	$.

Das Taggen einzelner Wörter funktioniert in einigen Sprachen, in denen Wörter ihre Bedeutung durch Suffixe definieren, nicht oder nur begrenzt (agglutinierende Sprachen).
Ein weiterer wichtiger Teil der maschinellen Sprachverarbeitung ist das Erkennen von Sätzen beziehungsweise Satzgrenzen (*Sentence splitting*) und ihrer grammatikalischen Struktur (*Sentence Parsing*). Das für die Erstellung von Ontologien notwendige Erkennen von Entitäten als Teil der natürlichen Sprachverarbeitung wird im nächsten Abschnitt behandelt.

4.2 Named Entity Recognition

Ein entscheidender Bestandteil dieser und ähnlicher Arbeiten ist die Erkennung und Kategorisierung von Entitäten, die *Named Entity Recognition* (NER). Sie bezeichnet die Erkennung von Entitäten wie Personen, Unternehmen, geografischen Entitäten wie Städten oder Zahlen wie Geldbeträgen.
Es gilt dabei nicht nur die Entitäten als solche zu registrieren, sondern auch der richtigen Kategorie zuzuordnen, was bei Eigennamen von Personen, Städten und Organisationen oft nicht einfach ist. Bei Personennamen kommt die Unterscheidung von Vor- und Nachnamen hinzu, wobei einige Namen sowohl als Vor- als auch als Nachname verwendet werden [PM08]. Anders als beim Part-of-speach-tagging (POS) soll bei der Named Entity Recognition explizit nicht jedes Wort definiert, sondern nur Entitäten getaggt werden. Die Erkennung basiert in der Regel auf definierten oder gelernten Features wie Großschreibung, Vorkommen in Wörterbüchern, charakteristischen Wortteilen (oft in chemischen Stoffen oder Medikamenten) und ähnlichem. Weiterhin können umgebende Wörter oder Tags als Feature genutzt werden. Wird ein (Vor-)Name erkannt, folgt mit höherer Wahrscheinlichkeit ein weiterer Name als auf ein anderes Wort.
Grob kann man die Ansätze zur Named Entity Recognition in regelbasierte Verfahren und solche unterteilen, die mit maschinellem Lernen arbeiten [CLR13]. Oft werden diese kombiniert.

4.2.1 Regel-basierte Ansätze

Regelbasierte Entitätserkennung in Texten beruht auf der Festlegung spezieller Satzstrukturen und gegebenenfalls Wortlisten. Die Entität wird dann aufgrund ihrer Position im Satz oder ihres Vorkommens in der Liste als solche erkannt. Der Vorteil regelbasierter Systeme ist die Nachvollziehbarkeit der Ergebnisse und die Unabhängigkeit von Trainingsdaten. Dadurch lassen sie sich auf neue oder veränderte Anforderungen anpassen, ohne dass das Modell neu trainiert werden muss [CLR13].

[1] Deutscher Textkorpus aus Nachrichtentexten mit POS-Tags. `http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/tiger.en.html`, abgerufen 28.07.2017

Regeln können in verschiedenen Formen auftreten. In [CKL+10] wird eine Sprache entwickelt, in der Regeln für Named Entity Recognition definiert werden können. Oft werden reguläre Ausdrücke (mit-)verwendet ([PM08], [CCM+99]).

4.2.2 Maschinelles Lernen

Beim maschinellen Lernen erkennt der Algorithmus Muster in vorhandenen Daten und wendet diese auf den Zieldatensatz an. Man unterscheidet zwischen verschiedenen Lernmethoden wie dem überwachten Lernen (*supervised learning*), dem unüberwachten Lernen (*unsupervised learning*) und anderen. Algorithmen für überwachtes Lernen lernen anhand von Trainingsdaten eine (wahrscheinliche) Struktur der Daten. In der Regel sind dies Label wie POS-Tags, die auf Grundlage des gelernten Modells auf die realen Daten angewandt werden. Es lässt sich so eine hohe Genauigkeit erreichen. Der Trainingsdatensatz muss für jede Veränderung in den Labels angepasst werden. Auch wenn sich die grobe Struktur der Eingabedaten ändert muss der Algorithmus neu trainiert werden. Allerdings werden lernende Algorithmen auf ähnlichen zu den Trainingsdaten strukturierten Eingabemengen meist bessere Ergebnisse liefern als regelbasierte Ansätze (zum Beispiel: trainiert zur Findung geografischer Entitäten in Nachrichtentexten und angewendet auf Publikationen).

Algorithmen für unüberwachtes Lernen wird keine feste Kategorisierung vorgegeben. Sie unterteilen die Daten nach ähnlichen Mustern selbst. Da keine Vorgaben gemacht werden können auch die gefundenen Mengen / Cluster keiner eindeutigen Kategorie wie „Organisationsname“ oder eindeutigen POS-Tags zugeordnet werden. Der Vorteil unüberwachten Lernens ist, dass das System nicht auf neue Eingabemengen angepasst werden muss. Hierbei kommen häufig neuronale Netze zum Einsatz.

In der Praxis werden die beiden Ansätze im teilüberwachten Lernen (*semi-supervised learning*) kombiniert. Das System kommt dann mit einem kleineren Trainingsdatensatz aus, aus dem gegebenenfalls zunächst ein größerer Satz an Trainingsdaten generiert wird. Bei Bedarf kann im Zwischenschritt das Ergebnis kontrolliert und korrigiert werden.

Im Maschinellen Lernen werden diverse unterschiedliche Methoden eingesetzt. Häufig genannt werden Support Vector Maschinen, Hidden Markov Modelle und ihre Weiterentwicklung Maximum Entropy Markov Modelle und Conditional Random Fields.

4.2.3 Modelle für Maschinelles Lernen

Hidden Markov Modell

Ein **Hidden Markov Modell** (HMM) ist ein spezieller endlicher Automat mit stochastischen Übergängen (Transitionen). Sie basieren auf Markov-Ketten. Markov-Ketten sind stochastische Prozesse mit diskreter Zeit. Der nächste Zustand wird von

einer begrenzten Menge Vorgängerzustände bedingt. Ihnen liegt die Feststellung zugrunde, dass eine eingeschränkte Menge an Vorgängerzuständen ausreicht um die Eintrittswahrscheinlichkeit von Folgezuständen zu berechnen und nicht alle vergangenen Zustände mit einbezogen werden müssen. Dies wird als *Markov-Eigenschaft* bezeichnet. Hidden Markov Modelle verwenden Markov-Ketten erster Ordnung, das heißt der Übergang in den nächsten Zustand wird ausschließlich durch den aktuellen Zustand bedingt. Die Indexmenge der Zustände wird als $I = \mathbb{N}_0$ angenommen, der Zustandsraum ist abzählbar [WW16].

Definition 4.2 *Ein stochastischer Prozess $\mathcal{Y} = (Y_t, t \in \mathbb{N}_0)$ auf einem Wahrscheinlichkeitsraum (ω, F, P) mit abzählbarem Zustandsraum (E, ε) heißt* ***Markov-Kette erster Ordnung*** *wenn für alle $t \in \mathbb{N}_0$ und alle Zustände $y_0, y_1, ..., y_{t-1}, y_t, z \in E$ gilt:*

$$P(Y_{t+1} = z | Y_t = y_t, Y_{t-1} = y_{t-1}, ..., Y_0 = y_0) = P(Y_{t+1} = z | Y_t = y) \tag{4.1}$$

Jeder Zustand aus E kann in jeden anderen Zustand übergehen. Für alle $t \in \mathbb{N}_0$ ergibt sich die *Übergangswahrscheinlichkeit* von Zustand y in Zustand z zum Zeitpunkt t:

$$p_{yz}(t) := P(Y_{t+1} = z | Y_t = y_t)$$

Die Besonderheit von Hidden Markov Modellen ist, dass der aktuelle Zustand als solcher nicht bekannt ist, sondern lediglich seine Ausgabe. Die Ausgabe ist ein Zustand $\mathcal{X}_t$, der durch die nicht beobachtbaren Zustände generiert wurde. Das System soll also das wahrscheinlichste Modell unbekannter Zustände $\mathcal{Y}_t$ finden, die die dem Nutzer vorliegenden bekannten Zustände bedingen [Fin14]:

Definition 4.3 *Ein Hidden Markov Modell ist ein zweistufiger Prozess. Die erste Stufe ist ein stochastischer Prozess $\mathcal{Y} = (Y_t, t \in \mathbb{N}_0)$ mit abzählbarem Zustandsraum (E, ε) und Zuständen $y_0, ..., y_t \in E$. Die zweite Stufe ist ein Prozess $\mathcal{X} = (X_t, t \in \mathbb{N}_0)$ mit Zuständen (Ausgaben) $x_0, ...x_t$:*

$$\begin{aligned}
&(1)\ P(Y_t = y_t | Y_{t-1} = y_{t-1}, ..., Y_0 = y_0) = P(Y_t = y_t | Y_{t-1} = y_{t-1}) \\
&(2)\ P(X_t = x_t | X_{t-1} = x_{t-1}, ..., X_0 = x_0, Y_t = y_t, ..., Y_0 = y_0) = P(X_t = x_t | Y_t = y_t)
\end{aligned} \tag{4.2}$$

Übertragen auf einen Text bedeutet dies: beobachtbar ist der vorhandene Text in Form von Ausgaben (Wörtern, $x_0, ...x_t$). Das zugrundeliegende Satzmuster mit Wortarten $y_0, ..., y_t$ ist unbekannt. Jedem Zustand y soll aufgrund der Ausgabe ein Label $l_0, ...l_m$ zugeordnet werden. Bekannte Algorithmen hierfür sind beispielsweise der Vorwärts-Rückwärts-, Viterbi- oder Baum-Welch-Algorithmus.

Maximum Entropy Markov Modelle

Die Beschränkung auf einen einzigen Zustand vereinfacht das Hidden Markov Modell zwar, reicht aber oft für die Textverarbeitung nicht aus. Ein Wort aus dem Namen eines unbekannten Unternehmens lässt sich ohne weiteren Kontext wie umgebende Wörter oder der Position im Text nur schwer als solches erkennen [MFP00]. Auch Textformatierung oder Wortendungen können interessant sein. Die genannten Features sind dabei (teilweise) abhängig voneinander.
Diesen Umstand sollen die in [MFP00] eingeführten **Maximum Entropy Markov Modelle** (MEMM) einbeziehen. Die Funktionen der HMM werden durch eine einzige Funktion ersetzt, die den Aktuellen Zustand auf den vorherigen und die aktuelle Ausgabe zurück führt:

$$P(y_t|y_{t-1}, x_t) \tag{4.3}$$

Der Unterschied der Abhängigkeiten ist in Abbildung 4.1 dargestellt. Die bedingten Wahrscheinlichkeiten der Zustände sind durch ein exponentielles Modell beschrieben das, gegeben den letzten Zustand und die Beobachtung, eine Wahrscheinlichkeitsverteilung über die möglichen nächsten Zustände ausgibt. Auf diese Weise kann ein wahrscheinlichster Pfad berechnet werden. HMM und MEMM teilen die Einschränkung, dass die Pfadberechnung sequenziell erfolgt und zukünftige Zustände oder Beobachtungen keinen Einfluss auf den aktuellen Zustand, beziehungsweise seine Wahrscheinlichkeitsverteilung haben. Um beim Graphenmodell zu bleiben handelt es sich um gerichtete Graphen. Es ist intuitiv, dass bei der Entscheidung über die Art eines Wortes das oder die darauffolgenden Wörter von Interesse sein können. Dieses Problem ist als *label bias-Problem* bekannt. Eine Lösung wird im nächsten Abschnitt vorgestellt.

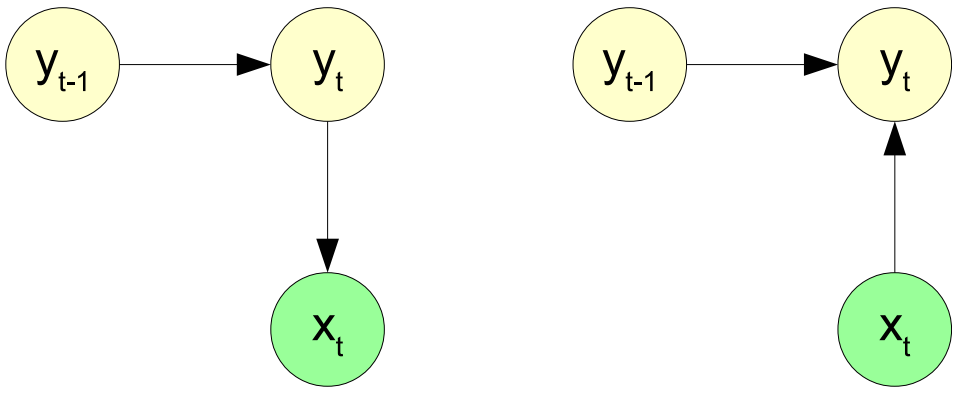

Abbildung 4.1: Abhängigkeitsgraphen für Hidden Markov Modelle (a) und Maximum Entropy Markov Modelle (b) [MFP00]

4.3 Conditional Random Fields

4.3.1 Prinzip

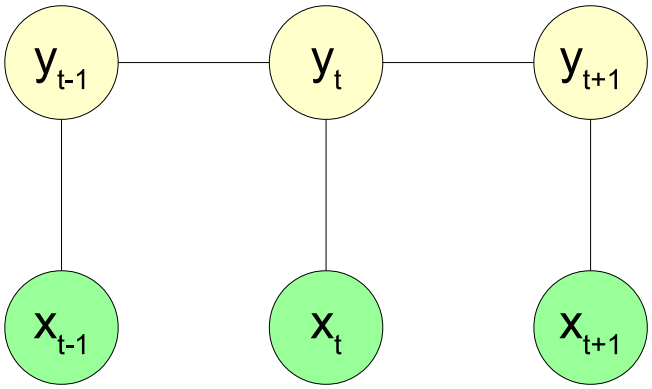

Abbildung 4.2: Graphisches Modell von Conditional Random Fields. Wie bei MEMM und im Gegensatz zu HMM werden die Beobachtungen x nicht vom Modell generiert [LMP+01]

Conditional Random Fields werden in [LMP+01] als Ansatz vorgestellt, der das *label bias-Problem* löst. Zukünftige Zustände sollen in die Wahrscheinlichkeitsverteilung des aktuellen Zustandes einfließen. Um dies zu erreichen wird eine einzige Wahrscheinlichkeitsverteilung für die gesamte Sequenz, gegeben die Beobachtungssequenz, berechnet. Im Unterschied zu HMM und MEMM handelt es sich hier nicht mehr um einen gerichteten, sondern einen ungerichteten Graphen. Der Zeitpunkt $t+1$ wird als letzter Zeitpunkt angenommen, y_{t+1} beschreibt also den letzten Zustand. Ein Beispiel des Graphen ist in Abbildung 4.2 zu sehen.
Betrachtet man die Sequenzen als Vektoren, also $(y_0, ...y_t, y_{t+1}) = \mathbf{y}$ und $(x_0,...,x_t,x_{t+1}) = \mathbf{x}$ lässt sich die Abhängigkeit vereinfacht mit

$$p(\mathbf{y}|\mathbf{x})$$

beschreiben. Lafferti et al. [LMP+01] definieren CRFs im Allgemeinen so:

Definition 4.4 *Sei $G = (V, E)$ ein Graph, so dass $\mathbf{Y} = (\mathbf{Y}_v)_{v \in V}$ und $\mathbf{Y}$ von den Knoten in G indiziert wird. Dann ist $(\mathbf{X}, \mathbf{Y})$ ein Conditional Random Field wenn die durch $\mathbf{X}$ bedingten Zufallsvariablen $\mathbf{Y}_v$ die Markov-Eigenschaft im Bezug auf folgenden Graphen erfüllen:*
$p(\mathbf{Y}_v|\mathbf{X}, \mathbf{Y}_w, w \neq v) = p(\mathbf{Y}_v|\mathbf{X}, \mathbf{Y}_w, w \sim v)$. w und v sind Nachbarn in G wenn $w \sim v$ gilt.

Da es sich um einen ungerichteten Graphen handelt bedeutet die Markov-Eigenschaft in diesem Fall eine Abhängigkeit von (ausschließlich) den Nachbarknoten. G wird

für diese Anwendung als Kette angenommen.

Um das Modell zu vervollständigen werden die *Features* f_k und g_k eingeführt. Ein Feature ist eine Eigenschaft wie die Großschreibung eines Wortes, ein spezieller Tag, oder eine Kombination. f_k steht für ein Kanten-Feature, g_k für ein Knoten-Feature. Kanten-Features stellen die beschriebene Abhängigkeit von benachbarten Knoten dar. Um ein Modell für einen Datensatz zu lernen müssen die Parameter $\theta = (\lambda_1, \lambda_2, ...; \mu_1, \mu_2, ...)$ gelernt werden, die die Features gewichten:

$$p_\theta(\mathbf{y}|\mathbf{x}) \propto \exp\left(\sum_{e \in E,k} \lambda_k f_k(e, \mathbf{y}|_e, \mathbf{x}) + \sum_{v \in V,k} \mu_k g_k(v, \mathbf{y}|_v, \mathbf{x})\right) \tag{4.4}$$

$\mathbf{y}|_e$ und $\mathbf{y}|_v$ sind die Knoten, die zum Teilgraphen von e oder v gehören, also auf die sich die Features beziehen.

4.3.2 Stanford NER

Das Named Entity Recognition Modul der Standford NLP-Gruppe [FGM05] (*Stanford NER*) basiert auf Conditional Random Fields. Es ist Teil des in Abschnitt 2.3 beschriebenen Stanford Core NLP. Trainiert wird das Modell mittels des Viterbi-Algorithmus.
Das genutzte CRF-Modell bezieht zunächst nur die benachbarten Zustände mit ein. Ein Zustand und sein Vorgängerzustand bilden eine *Clique*. Eine Clique ist ein zusammenhängender Teilgraph eines ungerichteten Graphen in dem alle Zustände (Knoten) miteinander verknüpft sind. In diesem Fall umfasst eine Clique also zwei Zustände. Jeder Clique c wird ein Potential ϕ_c zugeordnet. Das Potential ist eine Tabelle, die jeder möglichen Label-Belegung der Clique einen Wert zuordnet. Diese werden anhand der Beobachtungssequenz $\mathbf{x}$ berechnet, beziehen sich aber hier nur auf die lokalen Abhängigkeiten innerhalb der Clique. Aus den Potentialen aller Cliquen lässt sich die Wahrscheinlichkeit der Zustandssequenz $\mathbf{y} = y_0, ..., y_N$ ableiten:

$$P_{CRF}(\mathbf{y}|\mathbf{x}) = \propto \prod_{i=1}^{N} \phi_i(y_{i-1}, y_i) \tag{4.5}$$

Die Wahrscheinlichkeitsverteilung für einen Zustand ist abhängig von den Potentialen der Cliquen in denen er liegt, hier also zwei. Wie in den vorherigen Kapiteln beschrieben reicht dies aus um die Verteilung in Abhängigkeit von der gesamten Sequenz (und den Beobachtungen) zu berechnen.

$$P_{CRF}(y_i|y_0, ..., y_{i-1}, y_{i+1}, y_N, \mathbf{x}) \propto \phi_i(y_{i-1}, y_i)\phi_{i+1}(y_i, y_{i+1}) \tag{4.6}$$

In [FGM05] wird vorgestellt, wie das Modell mit Gibbs-Sampling optimiert werden kann. Der Vorteil liegt darin, dass auch weiter entfernt liegende Features mit einbezogen werden können. Dadurch können Referenzen wie verkürzte Nennung von Personen oder Organisationen der vorherigen Nennung zugeordnet und korrekt getaggt

werden. Die Autoren beziehen sich dabei also auf Wörter, die an unterschiedlichen Stellen unterschiedlich (oder mit zwei verschiedenen Labeln) getaggt werden. Daraus lässt sich die Wahrscheinlichkeit ableiten, dass ein Wort auch ein bestimmtes anderes Label haben kann (Beispiel: Eine als Personenname erkannte Wortfolge kann an anderer Stelle als Organisationsname auftreten). Der im Training berechnete Wert kann so korrigiert werden.

4.4 Relationen

Relationen sind Beziehungen zwischen einer oder mehrerer Entitäten. In der Regel wird hier von zwei Entitäten also *Entität1 - Relation - Entität2* (oder *Subjekt - Prädikat - Objekt*) ausgegangen. Entitäten sollten in solchen Tripeln gleich benannt sein, um zu ermöglichen verschiedene Fakten über die gleiche Entität zu verknüpfen [Bas13].
Am einfachsten lassen sich Relationen aus Datenbanken und Tabellen extrahieren. Auf diese Weise werden auch in dieser Arbeit Städte, Bundesländer und Universitäten mit Websites als Fakten aus einer Tabelle extrahiert. Beispielsweise „Albert-Ludwigs-Universität Freiburg - *City* - Freiburg i. Br.“. Darüber hinaus existieren Wissensdatenbanken wie *Yago* oder *DBpedia*.
Oft liegen die Daten aber nicht in solch strukturierter Form vor. Bei teilweise strukturierten Daten (und bekannten zu extrahierenden Relationen) bieten sich reguläre Ausdrücke an [WHS16]. Die Relationen und Entitätsklassen müssen dafür von Hand festgelegt werden. Sollen viele verschiedene Relationen aus großen (textuellen) Datensätzen extrahiert werden, liegt die Idee nahe diese zu lernen. In [BCS+07] wird ein System vorgestellt, das nicht nur Relationen lernen kann, sondern auch Klassenstrukturen. Dabei gelingt es nach Angaben der Autoren auch Unterscheidungen zu machen, die von einem regulären Ausdruck wahrscheinlich als korrekte Relation erkannt werden würde.

4.5 Weitere Informationen

Neben der expliziten Extraktion und Klassifizierung von Textteilen lassen sich noch andere Informationen aus Texten ziehen. Beispielsweise können Texte verschlagwortet werden indem nach relevanten Worten gesucht wird. Eine einfache Variante ist die Berechnung von BM25 Scores für jeden Text. Dies funktioniert relativ gut und kann beliebig optimiert werden.
Für die Wissenschaftlersuche kann es interessant sein den gefundenen Personen ein Wissenschaftsfeld zuzuordnen. Es bieten sich die von der OECD definierten *Fields of Science and Technology (FOS)* an, die sechs Hauptzweige mit insgesamt 42 Unterkategorien definieren. Die FOS tauchen oft in URLs oder Titeln der Hochschulwebsites und persönlichen Homepages auf, meist in Form der Unterkategorien wie „Institut

für Informatik - Lehrstuhl für ...“.
In der verwendeten Suchmaschiene Broccoli wird davon ausgegangen, dass relevante Eigenschaften und Zusatzinformationen meist in der Nähe der Entität zu finden sind [BBBH12]. Werden also Texte nicht als Ganzes betrachtet, sondern in sinnvoller Unterteilung in Abschnitte (*Kontexte*) kann davon ausgegangen werden, dass Personen (unter anderem) in Verbindung mit ihren Fachgebieten genannt werden. Dieser Ansatz wird auch in dieser Arbeit angewandt und evaluiert.

5 Umsetzung

5.1 Daten

Da sich die monatlichen Web-Crawls der Common Crawl-Organisation nur zu ca. 50% überlappen [Com16] müssen mehrere Crawls einbezogen werden. Gleichzeitig sollten die Daten nicht zu alt sein, da die gesuchten Daten einem ständigen Wandel unterliegen. Für die Auswertung werden daher die Crawls November 2016 bis Mai 2017 verwendet. Diese werden mittels des bereitgestellten Index nach den Websites der 401 deutschen Hochschulen[Hoc] durchsucht. Es ergeben sich hieraus über 8,9 Millionen URLs.
Bei der unbearbeiteten Auswertung dieser URLs stechen einige Universitäten mit sehr vielen Treffern hervor, bei denen es sich meist um Einträge in Bibliotheksseiten und Publikationsverzeichnissen handelt. Diese Menge an Einträgen kann die Ergebnisse verfälschen:

- Personen, die eigentlich an anderen Hochschulen tätig sind tauchen überproportional oft im Publikationsverzeichnis der betrachteten Hochschule auf.
- Personen, denen keine Homepage zugeordnet werden kann werden gegebenenfalls der falschen Hochschule zugeordnet, da sie dort oft in Publikationsverzeichnissen genannt werden.
- In den Verzeichnissen werden oft Mitarbeitende genannt, die nicht zum gesuchten Personenkreis der deutschen Hochschulen gehören.
- In einigen Publikationsverzeichnissen wird pro Autor eine persönliche Seite generiert. Diese kann vom Programm als Homepage der Person interpretiert werden.
- Websites von (Universitäts-)Bibliotheken enthalten naturgemäß eine große Zahl von Personen, die nicht zum gesuchten Personenkreis gehören.

Um diese Fehler zu vermeiden werden fünf häufig genutzte Subdomains und 31 hochschulspezifische Adressen aus den Ergebnissen herausgefiltert. Bei den Subdomains handelt es sich um drei Synonyme für Bibliotheken („bibliothek“, „ub“ und „ulb“) und zwei für Publikationsverzeichnisse („edoc“ bzw. „edocs“). Die hochschulspezifischen Adressen werden anhand der häufigsten Subdomains in der URL-Liste ermittelt.
Die verbleibenden Texte haben einen Umfang von 4 239 183 URLs mit einer Gesamtgröße von 14 GB.

5.2 Verarbeitung

5.2.1 Vorverarbeitung

Die extrahierten Texte werden mittels des Stanford CoreNLP verarbeitet. Als Standard wird das deutsche Sprachmodell verwendet, bei englischen Texten das englische. Dabei werden die Texte in Sätze zerlegt. In einigen Fällen funktioniert dies nicht. Zum Beispiel bei Aufzählungen auf Websites, die keine vollständigen Sätze enthalten. In diesen Fällen werden lange Wortketten in der weiteren Verarbeitung aufgeteilt.
Der Output des Stanford Parsers liegt im JSON-Format vor und kann in der weiteren Verarbeitung als Python-Dictionary (*dict*) eingelesen werden. Die verarbeiteten Texte werden im Format „URL [*Tab*] JSON“ mit einer Zeile pro URL/Text gespeichert.

5.2.2 Personenprofile

Hochschulen und ihre Websites, Städte und Bundesländer werden über Listen eingelesen, so dass die verschiedenen URLs jeweils einer Hochschule zugeordnet werden können. Die gespeicherten Daten aus Unterabschnitt 5.2.1 werden zeilenweise in Threads verarbeitet. Ein Text besteht dabei aus einem (JSON-)Dictionary mit mehreren Sätzen, die jetzt wie oben beschrieben gegebenenfalls noch einmal unterteilt werden müssen. Die Satztrennung wird so durchgeführt, dass sie nicht direkt vor oder nach erkannten Personen passiert. Wie in [BBBH12] wird davon ausgegangen, dass relevante Informationen in der Regel „in der Nähe “von Personen im Text stehen. Das heißt eine Person wird unter anderem zusammen mit ihrem Fach- und / oder Spezialgebiet auftauchen. Dies ist außerdem ein notwendiger Faktor um falsch positive Treffer bei der Volltextsuche zu minimieren: Auf vielen Seiten (Seiten für Vortragsankündigungen, Ehrungen, Mitarbeiter der Hochschule und viele andere) stehen Personen in einem Text mit beliebigen anderen Fachdisziplinen als der eigenen.
Es ergeben sich 35 284 036 Kontexte, also Sätze oder Satzteile. Pro Kontext werden im Verarbeitungsthread die Personen identifiziert und mit akademischen Titeln verknüpft. Da die Ausgabe des Stanford Parsers keine Namenskombination als vollständigen Namen ausgibt, sondern einzelne Wörter als Namen taggt findet dieser Schritt hier statt. Die akademischen Titel werden in der Initialisierung aus einer Liste eingelesen. Zunächst werden dann alle aufeinander folgenden Personennamen und Titel in eine Liste / einen String geschrieben. Ein solcher Namensstring ist abgeschlossen wenn als nächstes ein Wort steht, welches kein Name ist, der Satz endet oder ein Zeilenumbruch gelesen wird. Zwar könnte die Bedingung des Zeilenumbruches Namen trennen, der Datensatz enthält aber häufig Namenslisten in denen ein Name pro Zeile steht. Dies schließt auch Kommata mit ein, da Namen oft in der Form „Nachname, Vorname“ geschrieben werden. Für eine einzige Person ist der

Schritt relativ trivial, da dann ohne Komma die Schreibweise „Vorname Nachname“ angenommen werden kann. Mit Komma entsprechend umgekehrt. In Personenlisten kann es vorkommen, dass eindeutige Trennzeichen wie Zeilenumbrüche verloren gehen (zum Beispiel bei der Textextraktion aus HTML) oder nicht vorhanden sind (zum Beispiel Komma-separierte Listen).
Der entstandene Namensstring, der eine oder mehrere Personen repräsentiert, wird daher mittels eines Regulären Ausdrucks ausgewertet. Ähnlich wie in [BT08] werden dafür verschiedene mögliche Anordnungen von Namen und Titeln definiert. Im Unterschied zu [BT08] besteht der Namensstring bereits ausschließlich aus Personennamen, weshalb hier keine Verben einbezogen werden. Der Fokus liegt vielmehr darauf, eine vollständige Abdeckung des Namensstrings (mit Personenzuordnungen) zu erreichen.
Folgende regulären Ausdrücke werden verwendet:

Regex 1 `(?:(?:TITLE) )*(?:(?!NZS)[\w-]+) (?:\w\. )?(?:(?:NZS) )?[\w-]+`

Regex 2 `(?:(?:NZS) )?[\w-]+, (?:(?!NZS)[\w-]+)(?: \w\.)?,?(?: (?:TITLE))*`

Regex 3 `(?:(?:TITLE) )*(?:(?:NZS) )?[\w-]+, (?:(?!NZS)[\w-]+)(?: \w\.)?`

Mit (?:) als (*non-capturing*) Gruppen und (?!*regex*) als Ausschluss von *regex*. Wie üblich bezeichnet [\w-] eine Kombination aus Buchstaben und „-“, „+“ steht für mindestens ein und „?“ für höchstens ein Vorkommen. Der Platzhalter TITLE steht jeweils für eine Veroderung aller akademischen Titel aus der eingelesenen Liste, NZS für die möglichen **N**amens**z**u**s**ätze (von, zu, van, ...). Wichtig ist, die Namenszusätze in Verbindung mit den Vornamen auszuschließen, damit diese nicht als Vorname erkannt werden. Die Ausdrücke werden im folgenden an einem Beispiel erläutert. Als Namensstring dient die Kombination

> Prof. Dr. Erika Gabler, Dr. Mustermann, Max, von Schmidt-Schneider, Peter R., Ursula zu Müller

Regex 1 erkennt die Form [Titel] Vorname Nachname, im String also als alleinstehender Ausdruck

> Prof. Dr. Erika Gabler, Dr. Mustermann, Max, von Schmidt-Schneider, Peter R., Ursula zu Müller

Offensichtlich wird die Person Peter R. von Schmidt-Schneider nicht korrekt erkannt. Die Kombination Nachname, Vorname [, Titel] deckt **Regex 2** ab:

> Prof. Dr. Erika Gabler, Dr. Mustermann, Max, von Schmidt-Schneider, Peter R., Ursula zu Müller

Neben Peter R. Schmidt-Schneider wird auch *Max Mustermann* erkannt, allerdings fehlt der Titel. Diese Kombination ([Titel] Nachname, Vorname) wird schlussendlich von **Regex 3** abgedeckt.
Werden die regulären Ausdrücke kombiniert lässt sich der gesamte Namensstring abdecken:

Prof. Dr. Erika Gabler, Dr. Mustermann, Max, von Schmidt-Schneider, Peter R., Ursula zu Müller

Im Programm werden die drei Gruppen der Treffer, also Titel, Vorname, Nachname als benannte Gruppen extrahiert. Die Liste der so gewonnenen Personennamen wird ergänzt durch unvollständige Namen aus dem Seitentitel wie „Prof. Gabler“, sofern diese einem gefundenen Namen zugeordnet und so ergänzt werden können. Die Information des Vorkommens im Seitentitel wird im Verlauf zur Zuordnung der Homepage verwendet.
Die Namen werden auf Plausibilität geprüft, um grobe Fehler wie Sonderzeichen auszuschließen. Es werden Personeninstanzen angelegt, die den Rückgabewert der Textverarbeitungs-Threads bilden. Die verschiedenen Personenlisten werden zusammengeführt und Attribute ergänzt.

Um irrelevante und viele falsch erkannte Namen auszuschließen werden alle Personen mit nur einem Eintrag entfernt. Die Evaluation hat gezeigt, dass dies den Recall-Wert nicht verändert. Mindestens für die evaluierten Personenlisten scheint dies also angemessen zu sein. Es verbleiben 395 021 Personen. Die entstehende Ontologie hat 1 235 142 Einträge, von denen sich 1 234 249 auf Personen beziehen.

5.2.3 Relationen

Eine Person kann neben dem Namen folgende Eigenschaften besitzen, die sich als Relationen in der Suchmaschine widerspiegeln:

- Akademischer Grad
- Universität / Hochschule
- Geschlecht
- Homepage (bei der Hochschule)
- Subdomain (der Homepage)

Beliebig viele akademische Titel werden bei der Personenerkennung extrahiert. Herangezogen wird bei der Suche der höchste Grad, ein Prof. Dr. wird also als „Professor“ gefunden. Das Geschlecht wird anhand des Vornamens ermittelt und basiert auf Ausgaben der Website `https://genderize.io`. Als Herkunft des Namens, die bei der Zuordnung eine Rolle spielen kann, wird Deutschland verwendet. Die API gibt dann das wahrscheinlichste Geschlecht für die Kombination aus Name und Region aus.
Die persönliche Homepage der Person wird aufgrund von Vorkommen in der URL und Seitentiteln ermittelt. Wird eine Homepage gefunden bedingt diese die Hochschule, die der Person zugeordnet wird. Anderenfalls wird die Hochschule angenommen, auf deren Seiten die Person am häufigsten genannt wird. Außerdem wird die Subdomain der Homepage gespeichert.

Um eine geographische Zuordnung zu ermöglichen werden anhand der Hochschulliste [Hoc] die Hochschulen als Entitäten mit den Eigenschaften „Stadt“ (City) und „Homepage“ angelegt. Die Städte besitzen eine Relation zum Bundesland (State) in dem sie liegen. Ein Beispiel für die Kombination verschiedener Relationen zeigt Abbildung 5.1.

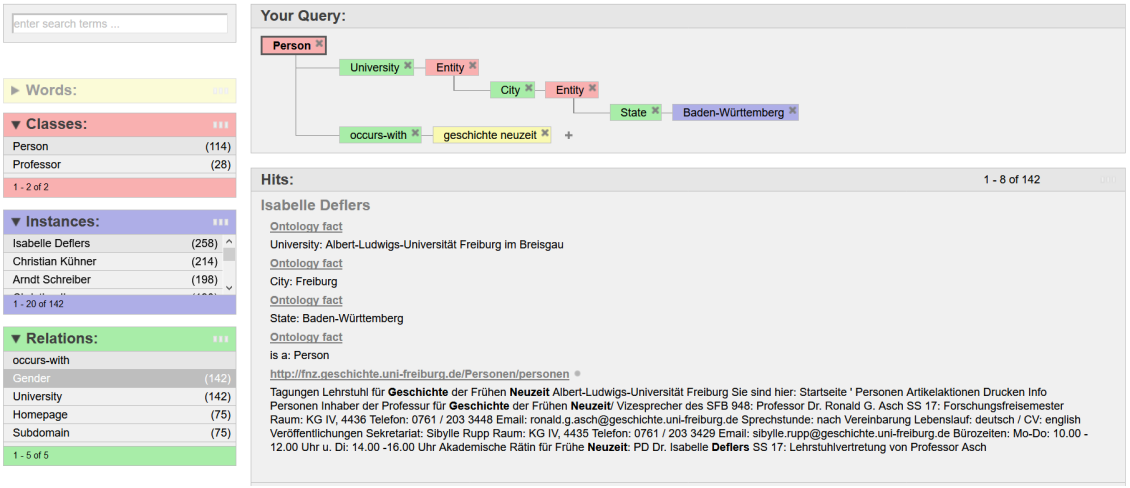

Abbildung 5.1: Beispiel für die Kombination verschiedener Relationen. Über die geografische Lage der Hochschule wird nach Bundesland gefiltert. Zusätzlich wird die Volltextsuche genutzt. Es werden dann verschiedene Personen vorgeschlagen, auf die diese Eigenschaften zutreffen. Im Kasten *Relations* werden weiter Relationen vorgeschlagen.

5.3 Laufzeit und verwendete Hardware

Im Programmablauf werden die gespeicherten getaggten Daten sequenziell gelesen und in 96 Threads verarbeitet. Jeder Thread verarbeitet eine Zeile, was einer URL entspricht. Mit ca. 43% hat das Einlesen der Daten den größten Anteil an der Laufzeit, da die Daten im JSON-Format erst interpretiert werden müssen. Die in den Threads extrahierten Personen werden in einem separaten Thread zusammengeführt. Gleichzeitig werden die Websitetexte von einem weiteren Thread in die verschiedenen Dateien für die Suchmaschine geschrieben. Nach Verarbeitung der Texte werden Homepages und Hochschulen der Personen festgelegt und nach doppelten Einträgen gesucht (beispielsweise wenn eine Person einmal mit zweitem Vornamen, einmal ohne angelegt wurde). Abschließend muss einmal die Wortliste (eine der Eingaben für die Broccoli-Datenbank) aktualisiert werden, da hier die Personen als Entität mit den Namen verknüpft werden, die sie repräsentieren. Die Laufzeit ist damit linear abhängig von der Anzahl der verarbeiteten Texte.

Für die Ausführung wird ein Dell PowerEdge R610 mit zwei Intel(R) Xeon(R) CPU E5649 verwendet. Die CPUs haben eine Taktfrequenz von 2.53GHz, je sechs Kerne, zwölf Threads und 12MB Cache. Es stehen 96GB RAM zur Verfügung. Die Ein- und Ausgangsdaten liegen auf einem Raid 5 mit 48TB.
Die Laufzeit beträgt 27 Stunden, von denen 20 Minuten auf die Nachbearbeitung, also das Zusammenführen, Aktualisieren und Löschen von Personen fallen.

5.4 Ground Truth

Als Ground Truth werden per Hand erstellte und in Listen vorhandene Daten verwendet. Die Gound Truth bildet die Grundlage für die spätere Evaluation und besteht jeweils aus der erwarteten Personenmenge für verschiedene Suchanfragen. Für einzelne Anwendungsfälle existieren Listen, wie zum Beispiel eine Liste aller Informatikprofessorinnen Deutschlands [IF17]. Vergleichbare Listen sind selten (hier verwendet: Liste von Lehrstühlen, die sich unter anderem mit Skandinavistik beschäftigen, Liste aller Physikprofessorinnen in Deutschland). Gerade Personenlisten sind kaum verfügbar und meist älter als zwei Jahre.
Die hier verwendeten Daten beziehen immer alle Personen mit ein, die „von Hand“ in den entsprechenden Fachbereichen aktuell (Mai / Juni 2017) gefunden werden können. Dies schließt oft auch studentische und Mitarbeiterinnen und Mitarbeiter der Sekretariate der Lehrstühle mit ein. Diese werden oft zusammen mit den festen Angestellten genannt und sind in den Rohdaten kaum von wissenschaftlichen Mitarbeitern zu unterscheiden. Nicht eingeschlossen werden unter anderem:

- Einträge in Publikationsverzeichnissen.
- Artikel, Newsseiten und Vortragsankündigungen in denen Personen wie Dozenten genannt werden.
- Generell Personen, die außerhalb der entsprechenden Lehrstühle in Verbindung mit den Suchbegriffen auftauchen.

Die für Anfrage 9 (Personen im Fachbereich Psychologie in Schleswig-Holstein, siehe Tabelle 6.1) werden beispielsweise die Psychologie-Lehrstühle der Universitäten Lübeck und Flensburg einbezogen. Des weiteren wird eine Professorin der Fachhochschule Kiel aufgeführt, die eine Professur für Psychologie und Gruppendynamik innehat. Diese ist keinem eigenen Fachbereich Psychologie angegliedert, sondern dem der Sozialen Arbeit und Gesundheit zugeordnet. Eine Stichwortsuche nach „Psychologie“ auf den Hochschulseiten der anderen Hochschulen in Schleswig-Holstein ergibt weitere Treffer, die sich jedoch auf Vorträge oder Gastdozentinnen und -dozenten einzelner Veranstaltungen beziehen. Diese werden nicht in der Ground Truth aufgeführt. Diese „Ausreißer“ können in den Suchergebnissen auftauchen, wenn ihnen keine andere deutsche Hochschule zugeordnet wird.
Die erstellten Listen enthalten insgesamt 998 Personen. Um die Unvollständigkeit

des Datensatzes einzubeziehen werden die Rohdaten nach diesen Personen durchsucht. Jeder Name wird dafür in gebräuchlichen Kombinationen per regulärem Ausdruck im ungeparsten Datensatz gesucht. Die Kombinationen sind:

- „Vorname Z. Nachname“; Mit Z. als optionale Abkürzung eines zweiten Vornamens
- „Nachname, Vorname“; Komma optional
- Bei vollständig bekanntem zweiten oder dritten Namen: Vollständiger Name

Es verbleiben 865 Namen. Entfernt werden vor allem Namen aus größeren Listen wie der oben erwähnten Anfrage 9 (11 von 50 entfernt) oder der Liste der Personen im Bereich Sportwissenschaften in Bayern (69 von 226 entfernt).

6 Evaluation

In diesem Kapitel werden einige Suchanfragen vorgestellt, anhand derer das Gesamtsystem bewertet wird. Die Ergebnisse werden im folgenden Abschnitt 6.1 beschrieben. In Abschnitt 6.2 werden einige der Fehler dargestellt, die zu den Ergebnissen führen und exemplarisch zwei Anfragen genauer ausgewertet. Auf dieser Grundlage werden die Werte in Abschnitt 6.3 bewertet.

6.1 Ergebnisse

Zur Auswertung werden verschiedene Suchanfragen ausgewertet, für die im Voraus eine Ground Truth erstellt wurde. Insgesamt handelt es sich dabei um 19 Anfragen. Die Ground Truth besteht jeweils aus einer Personenliste mit 3 bis 225 Personen. Andere Entitäten wie Städte und Hochschulen werden über vorhandene Listen erstellt und sind deshalb für die Auswertung nicht relevant.
In Tabelle 6.1 sind die Anfragen aufgeführt, die zur Auswertung herangezogen werden. Die Beispiele sind über verschiedene Fachgebiete und Bundesländer verteilt.

Nr.	Suchanfrage	*Relation* Entität
1	Alle Professorinnen und Professoren der Informatik an der Uni Freiburg	*University* Universität Freiburg *is-a* Professor *Subdomain* Informatik
2	Alle ProfessorInnen der Geschichte der Uni Hamburg	*University* Universität Hamburg *is-a* Professor *occurs-with* Geschichte
3	Professorinnen und Professoren Fachbereich Informatik in Ulm	*is-a* Professor *University –> City* Ulm *occurs-with* Informatik
4	Mitarbeiterinnen und Mitarbeiter am Institut für Hydrologie Freiburg mit eigener Homepage	*University* Universität Freiburg *Subdomain* Hydrologie
5	Professorinnen und Professoren an der Fakultät für Chemie und Pharmazie Uni Regensburg	*University* Universität Regensburg *is-a* Professor *occurs-with* Chemie
6	Mitarbeiterinnen und Mitarbeiter am Institut für Hirnforschung Uni Bremen mit eigener Homepage	*University* Universität Bremen *Subdomain* Brain

7	Mitarbeiterinnen und Mitarbeiter am Institut für Sprachwissenschaften der Universität Würzburg	*University* Universität Würzburg *occurs-with* Sprachwissenschaft
8	Alle Professorinnen und Professoren an der Hochschule für Musik Saarland	*University* HfM Saarland *is-a* Professor
9	Personen mit Fachbereich Psychologie in Schleswig-Holstein	*University* –> *City* –> *State* Schleswig-Holstein *occurs-with* Psychologie
10	Alle Professorinnen der Physik in Deutschland	*is-a* Professor *Gender* female *occurs-with* Physik
11	Mitarbeiterinnen und Mitarbeiter am Institut für Staatsphilosophie der Universität Köln mit eigener Homepage	*University* Universität Köln *Subdomain* Staatsphil
12	Alle Professorinnen der Mathematik in Deutschland	*is-a* Professor *Gender* female *occurs-with* Mathematik
13	Professorinnen und Professoren im Bereich Amerikastudien der Universität Leipzig	*University* Universität Leipzig *is-a* Professor *Subdomain* Americanstudies
14	Alle Professorinnen der Informatik in Deutschland	*is-a* Professor *Gender* female *occurs-with* Informatik
15	Personen mit Fachbereich Ägyptologie in Bayern	*University* –> *City* –> *State* Bayern *occurs-with* Ägyptologie
16	Personen mit Fachbereich Sportwissenschaft in Bayern	*University* –> *City* –> *State* Bayern *occurs-with* Sportwissenschaft*
17	Mitarbeiterinnen und Mitarbeiter im Exzellenzcluster „Hören für alle“/ “hearing4all“	*occurs-with* hearing4all
18	Mitarbeiterinnen und Mitarbeiter am Lehrstuhl für Völkerrecht der Universität Göttingen	*University* Universität Göttingen *occurs-with* Völkerrecht
19	Mitarbeiterinnen und Mitarbeiter am Institut für Biostatistik der Universität Hannover mit eigener Homepage	*University* Universität Hannover *Subdomain* Biostat

Tabelle 6.1: Evaluierte Suchanfragen

Nr.	GT	Found	TP	FP	FN	p	Recall	f	p@5	p@10	AP
1	22	24	16	8	6	0.67	0.73	0.70	1.00	0.80	0.65
2	20	44	2	42	18	0.05	0.10	0.06	0.40	0.20	0.04
3	38	63	8	55	30	0.13	0.21	0.16	0.20	0.30	0.07
4	40	42	36	6	4	0.86	0.90	0.88	0.80	0.90	0.85
5	32	33	13	20	19	0.39	0.41	0.40	0.60	0.80	0.32
6	18	23	16	7	2	0.70	0.89	0.78	1.00	0.90	0.83
7	22	63	12	51	10	0.19	0.55	0.28	0.60	0.60	0.32
8	48	52	27	25	21	0.52	0.56	0.54	0.60	0.50	0.32
9	29	27	6	21	23	0.22	0.21	0.21	0.20	0.20	0.06
10	67	258	22	236	45	0.09	0.33	0.14	0.40	0.40	0.12
11	3	1	1	0	2	1.00	0.33	0.50	-	-	0.33
12	168	310	47	263	121	0.15	0.28	0.20	0.60	0.30	0.10
13	4	7	1	6	3	0.14	0.25	0.18	0.20	-	0.25
14	112	364	48	316	64	0.13	0.43	0.20	0.80	0.90	0.24
15	32	23	6	17	26	0.26	0.19	0.22	0.60	0.40	0.12
16	156	178	29	149	127	0.16	0.19	0.17	0.60	0.50	0.07
17	24	39	6	33	18	0.15	0.25	0.19	0.40	0.40	0.10
18	20	29	8	21	12	0.28	0.40	0.33	0.60	0.40	0.29
19	10	10	4	6	6	0.40	0.40	0.40	0.60	0.40	0.34
Ges.	**865**	**159**	**308**	**1282**	**557**	**0.34**	**0.40**	**0.34**	**0.57**	**0.52**	**0.29**

GT – Personen in der Ground Truth
Found – Gefundene Personen
TP – *true positive* - Korrekte Treffer
FP – *false positive* - Inkorrekte Personen in der Treffermenge
FN – *false negative* - Nicht gefundene Personen
p – Präzision
f – F-Maß

Tabelle 6.2: Ergebnisse der Anfragen aus Tabelle 6.1

Aufgrund der Unterschiedlichkeit werden die vorgestellten Anfragen im Folgenden in drei Kategorien eingeteilt:

1. Anfragen über mehrere oder alle Hochschulen. Die entscheidende Relation ist hier *occurs-with* STICHWORT in Tabelle 6.3.
2. Anfragen an einer Hochschule mit der Relation *occurs-with* STICHWORT in Tabelle 6.4.
3. Anfragen mit der Relation *Subdomain* in Tabelle 6.5. Diese beziehen sich entsprechend nur auf eine Hochschule und Personen, die unter der genannten Subdomain eine persönliche Seite besitzen.

Die Ergebnisse und Mittelwerte über alle Anfragen sind in Tabelle 6.2 dargestellt. Unterschieden wird dabei nach Effektivitäts-Werten (Präzision, Recall und F-Maß)

Nr.	GT	Found	TP	FP	FN	p	Recall	f	p@5	p@10	AP
3	38	63	8	55	30	0.13	0.21	0.16	0.20	0.30	0.07
9	29	27	6	21	23	0.22	0.21	0.21	0.20	0.20	0.06
10	67	258	22	236	45	0.09	0.33	0.14	0.40	0.40	0.12
12	168	310	47	263	121	0.15	0.28	0.20	0.60	0.30	0.10
14	112	364	48	316	64	0.13	0.43	0.20	0.80	0.90	0.24
15	32	23	6	17	26	0.26	0.19	0.22	0.60	0.40	0.12
16	156	178	29	149	127	0.16	0.19	0.17	0.60	0.50	0.07
17	24	39	6	33	18	0.15	0.25	0.19	0.40	0.40	0.10
Ges.	**626**	**1262**	**172**	**1090**	**454**	**0.16**	**0.26**	**0.19**	**0.48**	**0.42**	**0.11**

Tabelle 6.3: Anfragen Kategorie 1: verschiedene Hochschulen, Relation *occurs-with*

Nr.	GT	Found	TP	FP	FN	p	Recall	f	p@5	p@10	AP
2	20	44	2	42	18	0.05	0.10	0.06	0.40	0.20	0.04
5	32	33	13	20	19	0.39	0.41	0.40	0.60	0.80	0.32
7	22	63	12	51	10	0.19	0.55	0.28	0.60	0.60	0.32
8	48	52	27	25	21	0.52	0.56	0.54	0.60	0.50	0.32
18	20	29	8	21	12	0.28	0.40	0.33	0.60	0.40	0.29
Ges.	**142**	**221**	**62**	**159**	**80**	**0.28**	**0.40**	**0.32**	**0.56**	**0.50**	**0.26**

Tabelle 6.4: Anfragen Kategorie 2: eine Hochschule, Relation *occurs-with*

Nr.	GT	Found	TP	FP	FN	p	Recall	f	p@5	p@10	AP
1	22	24	16	8	6	0.67	0.73	0.70	1.00	0.80	0.65
4	40	42	36	6	4	0.86	0.90	0.88	0.80	0.90	0.85
6	18	23	16	7	2	0.70	0.89	0.78	1.00	0.90	0.83
11	3	1	1	0	2	1.00	0.33	0.50	-	-	0.33
13	4	7	1	6	3	0.14	0.25	0.18	0.20	-	0.25
19	10	10	4	6	6	0.40	0.40	0.40	0.60	0.40	0.34
Ges.	**97**	**107**	**74**	**33**	**23**	**0.63**	**0.58**	**0.57**	**0.72**	**0.75**	**0.54**

Tabelle 6.5: Anfragen Kategorie 3: eine Hochschule, Relation *Subdomain*

und Ranking-Maßen, die in Unterabschnitt 6.3.1 und Unterabschnitt 6.3.2 separat betrachtet werden. Über die 19 durchgeführten Anfragen ergeben sich folgende Durchschnittswerte:

Präzision	Recall	F-Maß	Mean Average Precision (MAP)
0.34	0.40	0.34	0.29

Bei der Unterteilung in Anfragetypen bilden sich deutliche Unterschiede in der Präzision und des Recall-Wertes heraus. Besonders deutlich wird dies bei Kategorie 1, bei der alle F-Werte unter 23% liegen. In Kategorie 3 erzielt die Suche dagegen deutlich bessere Ergebnisse. Hier erreicht die Anfrage nach Personen am hydrologischen Institut Freiburg einen F-Wert von 88%.

Ein Faktor der in oben gezeigter Kategorisierung noch nicht einbezogen wurde ist die Relation *is-a* PROFESSOR. Ein direkter Vergleich aller neun Anfragen in denen diese Relation gewählt wird mit den zehn Anfragen in denen ohne Titel gesucht wird ist in Tabelle 6.6 zu sehen.

is-a	GT	Found	TP	FP	FN	p	Recall	f	p@5	p@10	AP
Professor	511	1155	184	971	327	0.25	0.37	0.29	0.53	0.53	0.24
Person	354	435	124	311	230	0.42	0.43	0.40	0.60	0.52	0.33

Tabelle 6.6: Vergleich der Anfragen mit *is-a* PROFESSOR und *is-a* PERSON

6.2 Fehleranalyse

Insgesamt liefert die Suche mit einer durchschnittlichen Präzision von 34% deutlich mehr oder andere Ergebnisse als anhand der Ground Truth erwartet werden. Dass mehr Personen gefunden werden als an den über die Ground Truth fokussierten Lehrstühlen ist aufgrund des Suchprinzips mit der Volltextsuche nicht ungewöhnlich. Durchschnittlich werden 40% der gesuchten Personen gefunden. Anhand Anfrage 6 lassen sich einige der Fehlerfaktoren beispielhaft identifizieren.

Suchanfrage 6

Gesucht werden 18 Personen mit einer Homepage unter der Subdomain „Brain“ an der Universität Bremen. Die Treffermenge enthält 22 Personen, von denen 16 korrekte Treffer sind. Dementsprechend werden zwei Personen nicht gefunden und sechs nicht gesuchte Ergebnisse angezeigt. Eine manuelle Auswertung der falsch positiven Treffermenge ergibt:

- Eine korrekte Person bei der der zweite Vorname als Nachname erkannt wird.

- Einen Schreibfehler im letzten Buchstaben eines Nachnamens, der auf der Lehrstuhl-Homepage mittlerweile korrigiert wurde. In der Ground Truth ist also der korrekte Name aufgeführt, in der Datenbank der falsche.
- Ein ehemaliges Lehrstuhl-Mitglied, hier sind die Daten des Web-Crawls veraltet.
- Eine Person, die in der Ground Truth nicht auftaucht, da sie nicht in der Liste der Mitarbeiter steht. Zusätzlich hat sich der Nachname geändert, was an der URL und E-Mail-Adresse ersichtlich ist.
- Zwei Personen die unter „Freunde des Lehrstuhls“ gelistet sind und der falschen Uni zugeordnet wurden.
- Ein Fehler in der Personenerkennung („Theoretical Neurobiology“)

Nach Verbesserung der Namenserkennung an dieser Stelle und Datenupdate würde sich die Präzision von 0.70 in diesem Fall also noch etwas verbessern.
Der Recall-Wert ist 0.89, wodurch sich ein F-Maß von 0.78 ergibt. Die zugrundeliegende Homepage des Instituts begünstigt die Informationsextraktion. Es existiert eine HTML-Tabelle mit Mitarbeiternamen, die im extrahierten Format in einzelnen Zeilen stehen und gut differenzierbar sind. Die Seite verlinkt auf persönliche Mitarbeiterseiten unter der gleichen Subdomain, die jeweils ausschließlich den Namen der Person als Titel tragen.

Suchanfrage 9

Die Erkennung von Personen eines Lehrstuhls ist ein relativ einfaches Problem. Deshalb soll weiterhin eine Suchanfrage der Kategorie 1 (Alle Hochschulen eines geografischen Gebietes und Relation *occurs-with*) betrachtet werden. Anfrage 9 bezieht sich auf Personen, die in Zusammenhang mit dem Stichwort Psychologie auftreten und einer Hochschule in Schleswig-Holstein zugeordnet sind. Es werden sechs von 29 Personen gefunden, weiterhin 21 falsch-positive und 23 falsch-negative Treffer. Die Präzision liegt also bei 22%, der Recall bei 21% und das F-Maß ebenfalls bei 21%. Zunächst soll die Präzision, also die falsch-positiven Treffer betrachtet werden:

- 5 Personen, die nur zufällig im Zusammenhang mit dem Stichwort in Texten vorkommen.
- 4 Mitarbeiterinnen und Mitarbeiter des Lehrstuhls für Neurowissenschaften (Lübeck), die sich auch mit dem Themenfeld Psychologie befassen.
- 3 Personen mit dem Themengebiet Geschichte, Theorie und Ethik der Medizin. Auch hier spielt die Psychologie als solche eine Rolle.
- 3 Personen, die einer falschen Hochschule zugeordnet werden, aber im gesuchten Fachgebiet tätig sind.
- 2 Mitarbeiter einer fachgebietsübergreifenden Arbeitsgruppe.

- Eine Namensüberschneidung mit einem Psychologie-Professor aus Heidelberg.
- Weiterhin ein erwähnter Autor und zwei Fehler in der Namenserkennung, davon ein Organisationsname.

Obwohl die meisten der Personen tatsächlich in Zusammenhang mit dem gesuchten Fachgebiet auftreten zeigen sich charakteristische Fehlerquellen. Die wichtigste für diese Anfrage ist, dass der Kontext der Personen zu groß gewählt ist. Eine Aufzählung verschiedener Professuren einer Uni kann die Ergebnisse stark verfälschen, da die Personen im Text in der Nähe verschiedener Fachgebiete stehen.
23 Personen wurden nicht gefunden. Davon sind

- 16 nicht in der Datenbank enthalten.
- 4 Personen der falschen Hochschule zugeordnet.
- 3 Personen nicht in einem Kontext mit dem Stichwort zu finden.

Die 16 fehlenden Personen sind größtenteils Mitarbeiterinnen und Mitarbeiter der Institute für Psychologie Lübeck. Die zwei Institute haben Mitarbeiterübersichten und persönliche Seiten für jede Person. Dort steht jeweils der Nachname im Seitentitel. Außerdem gibt es eine Überschrift (<h1>) mit dem vollen Namen der Person. Die Informationen beschränken sich auf Kontaktdaten, Texte finden sich auf den Seiten nicht. Das Fehlen des textuellen Kontextes der Namen erschwert die Erkennung.

Fehlende Daten

Einen Fehler in der Datengewinnung deckt Anfrage 15 auf, die nach Personen im Fachbereich Ägyptologie in Bayern fragt. Die Ground Truth listet 38 Personen der Julius-Maximilians-Universität Würzburg und Ludwig-Maximilians-Universität München, die dort jeweils am Lehrstuhl für Ägyptologie tätig sind. Die Treffermenge umfasst 21 Personen, von denen aber nur sechs zu den Gesuchten gehören. Es zeigt sich, dass der Münchener Lehrstuhl in den Ergebnissen komplett fehlt. In der Hochschulliste ist die Domain der Universität München mit lmu.de angegeben. Allerdings wird auch die Domain uni-muenchen.de genutzt. Lehrstühle benutzen unterschiedliche Secondlevel-Domains, der Lehrstuhl für Ägyptologie die Domain uni-muenchen.de. Da lmu.de dennoch verwendet wird und dementsprechend auch über 30 000 URLs dieser Domain vorhanden sind ist der Fehler zunächst nicht aufgefallen.

Zusammenfassung

Viele Personen die zusätzlich gefunden werden sind im gesuchten Fachgebiet oder überschneidenden Arbeitsbereichen tätig. Die Hauptgründe für das Fehlen von Personen in den Ergebnissen sind falsche Hochschul-Zuordnung oder vollständiges Fehlen in der Datenbank. Fehlerquellen sind weiterhin:

- Falsche oder keine Relation zu Homepage, Titel und Geschlecht.
- Fehlerhafte Namenserkennung (Zweiter Vorname als Nachname, „Verschmelzen“ von Personen in Listen).
- Personen, die nicht an einer (deutschen) Hochschule tätig sind, aber als Autoren, Mitarbeitende in Arbeitsgruppen oder in Vortragsankündigungen genannt werden.

6.3 Interpretation

Da mit den beschriebenen Suchanfragen das Gesamtsystem evaluiert wird bedingen sich die verschiedenen Verarbeitungsschritte. Der Faktor der fehlenden Daten wird wie beschrieben ausgeklammert, indem die Ground Truth auf die im Datensatz vorhandenen Personen reduziert wird. Der erste Faktor ist also, ob eine Person korrekt erkannt wird. Wenn das der Fall ist kann sie in den Suchresultaten fehlen, wenn lokal (Hochschule, Stadt, Bundesland) gesucht wird und die Hochschul-Zuordnung falsch ist. Gleiches gilt für die Subdomain und den akademischen Grad.
Eine Ground Truth zu definieren ist bei dieser Art der Suche nicht trivial, da auch Personengruppen für die Ergebnisse relevant sein können, die nicht an den hier fokussierten Lehrstühlen tätig sind. Sollen Experten für ein Thema gesucht werden wird ohnehin nicht nach dem Fach- sondern dem Spezialgebiet gesucht.
Generell gibt es also viele Personen die zumindest in einigen Fällen für die Ergebnismenge relevant sein können, hier aber nur als *relevant* oder *nicht relevant* gewertet werden. Gegebenenfalls kann es die Evaluation einer solchen Suche verbessern, wenn die Relevanz nicht nur binär, sondern anhand von Relevanzklassen gemessen wird [MRS08].

6.3.1 Präzision und Recall

Die höchsten Werte für Präzision und Recall werden in Kategorie 3 (siehe Tabelle 6.5) erreicht. Eine deutliche Ausnahme bildet hier Anfrage 13 mit 14%, die zusätzlich noch die Relation *is-a* PROFESSOR fordert. Über die Relation *Subdomain* werden hier nur Personen gelistet für die eine Homepage unter der gesuchten Subdomain gefunden wird. Es liegen hier also generell mehr Informationen vor. Zusätzlich ist die Treffermenge klar umrissen, so dass weniger falsch-positive Treffer auftauchen.
Kategorien 1 und 2 fassen Suchanfragen zusammen, die die Volltextsuche (*occurs-with*) verwenden. Hier zeigen sich mit einer Präzision von 5-52% und Recall-Werten zwischen 10 und 56% (F-Maß: 0.14-0.54) deutlich niedrigere Werte als in Kategorie 1. Dabei beträgt der durchschnittliche F-Wert in Kategorie 1 0.19 und in Kategorie 2 0.32. Es werden also bessere Ergebnisse erzielt wenn nur eine Hochschule einbezogen wird.
Die Relation *is-a* PROFESSOR bewirkt eher eine Verschlechterung der Präzision, wie

Tabelle 6.6 zeigt. Die Suchanfragen mit der Relation liefern eine durchschnittliche Präzision von 25%, während ohne die Relation 42% erreicht werden. Die Werte könnten darauf hindeuten, dass zu viele Personen als Professorin oder Professor gelabelt werden.

6.3.2 Ranking

Die Treffer werden nach Anzahl der Kontexte sortiert in denen die Person genannt wird. Wird die Relation *occurs-with*, also die Volltextsuche, genutzt werden nur diejenigen Kontexte gezählt, die die eingegebenen Wörter enthalten.
Kategorie 3 liefert auch hier wieder die besten Ergebnisse. Die in Abschnitt 6.2 vorgestellte Anfrage 6 hat mit einer Average Precision von 0.83 den zweitbesten Wert aller evaluierten Suchanfragen. Eine detaillierte Aufstellung der p@k-Werte zeigt Tabelle 6.7. Die einfache Sortierung funktioniert hier also sehr gut.
Dem Gegenüber steht Anfrage 9 mit einer Average Precision von 0.06. Die genau-

k	1	2	3	4	5	6	7	8	9	10	11	12
p@k	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.89	0.90	0.91	0.83

k	13	14	15	16	17	18	19	20	21	22	23
p@k	0.85	0.86	0.87	0.88	0.88	0.83	0.79	0.80	0.76	0.73	0.70

Tabelle 6.7: p@k-Werte der Anfrage 6

en Ergebnisse der p@k-Werte finden sich in Tabelle 6.8. Bei nur 6 Treffern von 29 gesuchten Personen und 21 falsch-positiven sind die 6 Personen über die Ergebnismenge verteilt. Eine Sortierung ist nicht erkennbar.
Zur Verbesserung der Ergebnisse sollten vor allem diejenigen Personen aus der Tref-

k	1	2	3	4	5	6	7	8	9	10	11	12
p@k	0.00	0.00	0.33	0.25	0.20	0.17	0.29	0.25	0.22	0.20	0.18	0.17

k	13	14	15	16	17	18	19	20	21	22	23	24
p@k	0.23	0.29	0.27	0.31	0.29	0.28	0.26	0.25	0.24	0.23	0.22	0.25

k	25	26	27
p@k	0.24	0.23	0.22

Tabelle 6.8: p@k-Werte der Anfrage 6

fermenge ausgeschlossen werden, die fälschlicherweise im Kontext mit dem gesuchten Wort stehen. Es würde sich hier für Anfrage 9 eine deutliche Verbesserung ergeben.

Ein Faktor der das Ranking wesentlich entscheidet, ist dass trotz Ausschluss der entsprechenden Domains viele Publikations-Nennungen verarbeitet werden. Oft bieten Lehrstühle oder Einzelpersonen auf ihrer Homepage Publikationslisten an die so mit ausgewertet werden. Da oft ähnliche Templates für Literaturverzeichnisse genutzt werden könnte versucht werden diese zu erkennen. Andererseits kann die Anzahl der Publikationen ein Indiz für die Relevanz einer Person (innerhalb eines Fachgebietes) sein. Dies funktioniert allerdings nur wenn die Daten gleichmäßig erhoben werden.

7 Schlussfolgerung und zukünftige Arbeiten

7.1 Schlussfolgerung

Die Arbeit zeigt wie Personendaten aus einem Webcrawl extrahiert werden können und als Grundlage für eine Suchmaschine aufbereitet werden können.
Obwohl die einzelnen Crawls der CommonCrawl-Organisation nicht alle Websites der deutschen Hochschulen enthalten, lässt sich bei Kombination mehrerer Crawls eine zufriedenstellende Abdeckung erreichen. Die Daten lassen sich dann aus den ungeordneten Segmenten extrahieren und weiterverarbeiten.
Mittels des Stanford Parsers lassen sich Personennamen im Text finden. Weitere Informationen zur Person lassen sich aus dem den Namen umgebenden Websitetext extrahieren. Die entstehende Ontologie und Textausschnitte werden für die Suchmaschine Broccoli aufbereitet. Es wird so nicht nur eine semantische Suche mit Hilfe der Relationen, sondern eine zusätzliche Volltextsuche ermöglicht. Eine Expertensuche wird vereinfacht, da direkt nach Stichwörtern gesucht wird, die in der Nähe der Personen im Text stehen.
Die Evaluation zeigt je nach Art der gewählten Suchanfrage mit F-Werten von 0.06 bis 0.88 gemischte Ergebnisse, arbeitet aber auch die Schwachstellen und möglichen Verbesserungen des Systems heraus. Mit einigen Verbesserungen lässt sich also eine solche Suche derart implementieren, dass sie ein hilfreiches Instrument bei der (geografischen) Expertensuche darstellt.

7.2 Zukünftige Arbeiten

Viele Personen werden nicht korrekt erkannt, obwohl ihre Namen im Datensatz vorhanden sind. Hier sollten zukünftige Arbeiten ansetzen und beispielsweise neben dem Titel-Tag der Websites auch andere Tags wie Überschriften einbeziehen. Die Personen müssen für eine geografische Suche der richtigen Universität zugeordnet werden, hier besteht noch Verbesserungsbedarf. Um die Zahl der falsch-positiven Treffer zu verringern sollte evaluiert werden, inwieweit kleinere Kontexte wie in [BBBH12] eine Verbesserung der Ergebnisse bewirkt.
Ein in dieser Arbeit ignorierter Punkt, der im größeren Umfang aber äußerst wichtig

ist, ist die Unterscheidung zweier Personen mit gleichem Namen (*name disambiguation*).
Ergänzend zu Websitetexten können Publikationslisten einbezogen werden. Es ist dann zu beachten, dass hier viele Personen anderer Einrichtungen oder aus dem Ausland genannt werden.
Die Datenbank lässt sich beliebig erweitern. Neben weiteren Daten die sich aus den Homepages extrahieren lassen bietet sich eine Ausweitung des Datensatzes auf Forschungseinrichtungen und Universitätskliniken an.

Literaturverzeichnis

[AGMU03] Arasu, Arvind ; Garcia-Molina, Hector ; University, Stanford: Extracting structured data from Web pages. In: *Proceedings of the 2003 ACM SIGMOD international conference on on Management of data - SIGMOD*, Association for Computing Machinery (ACM), 2003

[Bas13] Bast, Hannah: Semantische Suche. In: *Informatik-Spektrum* 36 (2013), feb, Nr. 2, S. 136–143

[BBBH12] Bast, Hannah ; Bäurle, Florian ; Buchhold, Björn ; Haussmann, Elmar: Broccoli: Semantic full-text search at your fingertips. In: *arXiv preprint arXiv:1207.2615* (2012)

[BCS+07] Banko, Michele ; Cafarella, Michael J. ; Soderland, Stephen ; Broadhead, Matthew ; Etzioni, Oren: Open Information Extraction from the Web. In: *IJCAI* Bd. 7, 2007, S. 2670–2676

[BLO13] Berners-Lee, Tim ; O'Hara, Kieron: The read–write Linked Data Web. In: *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences* 371 (2013), Nr. 1987, S. 20120513

[BT08] Bayraktar, Ozkan ; Temizel, Tugba T.: Person name extraction from Turkish financial news text using local grammar-based approach. In: *2008 23rd International Symposium on Computer and Information Sciences*, Institute of Electrical and Electronics Engineers (IEEE), oct 2008

[CCM+99] Cunningham, Hamish ; Cunningham, Hamish ; Maynard, Diana ; Maynard, Diana ; Tablan, Valentin ; Tablan, Valentin. *JAPE: a Java Annotation Patterns Engine*. 1999

[CKL+10] Chiticariu, Laura ; Krishnamurthy, Rajasekar ; Li, Yunyao ; Reiss, Frederick ; Vaithyanathan, Shivakumar: Domain adaptation of rule-based annotators for named-entity recognition tasks. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing* Association for Computational Linguistics, 2010, S. 1002–1012

[CLR13] Chiticariu, Laura ; Li, Yunyao ; Reiss, Frederick R.: Rule-based information extraction is dead! long live rule-based information extraction systems! In: *EMNLP*, 2013, S. 827–832

[Com16] CommonCrawl. *Common Crawl Website*. Online. 2016

[DG08] Dean, Jeffrey ; Ghemawat, Sanjay: MapReduce: simplified data processing on large clusters. In: *Communications of the ACM* 51 (2008), Nr. 1, S. 107–113

[FGM05] Finkel, Jenny R. ; Grenager, Trond ; Manning, Christopher: Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. In: *Proceedings of the 43rd annual meeting on association for computational linguistics* Association for Computational Linguistics, 2005. – Cite this for Stanford NER, S. 363–370

[Fin14] Fink, Gernot A.: *Markov Models for Pattern Recognition.* Springer, 2014. – ISBN 1447163079

[gen17] *Website genderize.io.* Online. 2017

[Hä13] Hänig, Christian: *Unsupervised Natural Language Processing for Knowledge Extraction from Domain-specific Textual Resources*, Universität Leipzig, PhD thesis, 2013

[Hau09] Hausenblas, Michael: Anreicherung von Webinhalten mit Semantik–Microformats und RDFa. In: *Social Semantic Web.* Springer, 2009, S. 147–158

[Hoc] Hochschulrektorenkonferenz. *Hochschulkompass der Hochschulrektorenkonferenz*

[IF17] Informatica-Feminale. *Website der Informatica Feminale.* Online. 2017

[LMP+01] Lafferty, John ; McCallum, Andrew ; Pereira, Fernando [u. a.]: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001). – Original CRF Paper

[MFP00] McCallum, Andrew ; Freitag, Dayne ; Pereira, Fernando C. N.: Maximum Entropy Markov Models for Information Extraction and Segmentation. In: *Proceedings of the Seventeenth International Conference on Machine Learning.* San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2000 (ICML '00). – Einführung von Maximum Entropy Markov Modellen. – ISBN 1–55860–707–2, S. 591–598

[MRS08] Manning, Christopher D. ; Raghavan, Prabhakar ; Schütze, Hinrich: *Introduction to Information Retrieval.* Cambridge University Pr., 2008. – ISBN 0521865719

[MSB+14] Manning, Christopher D. ; Surdeanu, Mihai ; Bauer, John ; Finkel, Jenny R. ; Bethard, Steven ; McClosky, David: The Stanford CoreNLP Natural Language Processing Toolkit. In: *ACL (System Demonstrations)*, 2014. – Stanford Parser in general Cite this for Stanford CoreNLP, S. 55–60

[PM08] Popescu, Octavian ; Magnini, Bernardo: Language Independent First and Last Name Identification in Person Names. In: *Computational Linguistics and Intelligent Text Processing*. Springer Berlin Heidelberg, 2008, S. 322–333

[SNG17] Stanford-NLP-Group. *The Stanford Parser: A statistical parser*. Online. 2017

[TZY07] Tang, Jie ; Zhang, Duo ; Yao, Limin: Social network extraction of academic researchers. In: *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on* IEEE, 2007, S. 292–301

[TZY+08] Tang, Jie ; Zhang, Jing ; Yao, Limin ; Li, Juanzi ; Zhang, Li ; Su, Zhong: ArnetMiner: Extraction and Mining of Academic Social Networks. In: *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD 08*, Association for Computing Machinery (ACM), 2008

[WHS16] Weikum, Gerhard ; Hoffart, Johannes ; Suchanek, Fabian: Ten Years of Knowledge Harvesting: Lessons and Challenges. In: *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 39 (2016), Nr. 3, S. 41–50

[WW16] Webel, Karsten ; Wied, Dominik: *Stochastische Prozesse*. Gabler, Betriebswirt.-Vlg, 2016. – ISBN 365813884X