Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Francis Opoku

Ontologieextraktion aus natürlichsprachlichen Texten

Fakultät Technik und Informatik
Studiendepartment Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Francis Opoku

Ontologieextraktion aus natürlichsprachlichen Texten

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Bachelor of Science Angewandte Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. rer. nat. Michael Neitzke
Zweitgutachter: Prof. Dr. rer. nat. Michael Köhler-Bußmeier

Eingereicht am: 31. August 2016

Francis Opoku

Thema der Arbeit
Ontologieextraktion aus natürlichsprachlichen Texten

Stichworte
Ontologien, Ontologieextraktion, Evaluation, Konzeptextraktion, Relationsextraktion, Instanzextraktion, Axiomextraktion

Kurzzusammenfassung
Die Idee des *Semantic Web* ist es, Inhalte im World Wide Web für Maschinen *verständlich* zu machen. Dafür ist es notwendig, dass dieses Wissen *formalisiert* wird. Eine Ontologie ist eine solche Formalisierung. Die *Ontologieextraktion aus natürlichsprachlichen Texten* ist die (Semi-)Automatisierung dieses Formalisierungsprozesses. Die Unterstützung der Ontologieerzeugung durch (semi-)automatische Ontologieextraktions-Methoden kann wertvolle Ressourcen sparen und es mehr Menschen und Unternehmen ermöglichen, Ontologien einzusetzen. Diese Arbeit zeigt den aktuellen Forschungsstand in Bezug auf die Ontologieextraktion aus natürlichsprachlichen Texten.

Title of the paper
Ontology Learning from Text

Keywords
Ontologies, Ontology Learning, Evaluation, Concept, Instance, Relation, Axiom, Extraction

Abstract
The idea behind the *Semantic Web* is it to make the World Wide Web *understandable* for machines. To do so knowledge has to be *formalized.* An ontology is such a formalisation. *Ontology learning from text* is the (semi-)automation of this process of formalisation. The support of (semi-)automatic ontology extraction methods may save important resources and enables more people and companies to use ontologies. This work shows the current state of research in ontology learning from text.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Wenn Menschen Informationen austauschen, dann tun sie dies *verbal* und/oder *nonverbal*. Die nonverbale Kommunikation umfasst u.a. die Kommunikation durch Mimik und Gestik. Die verbale Kommunikation erfolgt durch Sprachen, wie z.B. Deutsch und Englisch. In der Regel ist das Verstehen von Text für den Menschen keine große Herausforderung. Informationen, die in Texten natürlicher Sprache, wie z.B. Deutsch und Englisch, enthalten sind, können von Menschen leicht *ermittelt*, in einen bestimmten *Kontext eingeordnet* und *verstanden* werden. Maschinen haben diesbezüglich größere Schwierigkeiten. Die *Ermittlung* von natürlichsprachlichen Informationen aus Texten erfolgt durch Maschinen meistens in Form eines einfachen Vergleichs von Zeichenketten (Hitzler u. a., 2008, S. 10). Auf der untersten Ebene des Verstehens geht der Mensch genauso vor, wenn er Informationen in Texten sucht. Wenn ein Mensch in einer Suchmaschine die Suchbegriffe *lukrative Beschäftigung* eingibt, erhält dieser Suchergebnisse, in denen diese Wörter auftreten. Doch auch wenn diese Wörter in den Suchergebnissen enthalten sind, bedeutet dies nicht, dass die Suchergebnisse tatsächlich Antworten darauf liefern, welche lukrativen Beschäftigungen es gibt. Eine bessere Suchmaschine würde auch die Begriffe in die Suche einbeziehen, die der Suchphrase nahestehen, wie z.B. *Beruf, Gehalt, Verdienst, Lohn, Arbeitszeiten* etc. Dieses Vorgehen würde bedeuten, dass die Suchmaschine fähig wäre, die Suchphrase in einen *Kontext* einzuordnen. Eine weitere Steigerung wäre es, wenn die Suchmaschine fähig wäre, die Suche zu *verstehen*. Ein Beruf, in dem der Netto-Verdienst bei $5000€$ im Monat liegt, muss nicht lukrativ sein, wenn dafür 18 Stunden täglich gearbeitet werden müssen, an allen Tagen eines Monats. Eine solche Suchmaschine würde also auch Informationen über Suchbegriffe miteinander kombinieren können, in diesem Fall also *Gehalt* und *Arbeitszeiten*.

Das aktuelle Problem für Maschinen besteht darin, dass die natürlichsprachlichen Informationen, die sie finden sollen, in keiner für sie einfach zu verarbeitenden Form vorliegen. Sie sind eingebettet in elektronische Ressourcen, wie z.B. HTML (Hypertext Markup Language) und PDF (Portable Document Format). Diese Dokumente dienen jedoch hauptsächlich der Präsentation der Inhalte für Menschen. Die Informationen, die darin enthalten sind und von Maschinen verstanden werden, beziehen sich u.a. darauf, wo und wie bestimmte Inhalte der Dokumente angezeigt werden sollen, aber nicht darauf, in welchem Zusammenhang die natürlichsprachlichen Begriffe stehen. Im Kontext des World Wide Web (im Folgenden einfach nur „Web“ genannt) gab und gibt es deshalb Bestrebungen, Informationen auf Webseiten stärker zu formalisieren, sodass Softwareprogramme diese leicht interpretieren können und dadurch in die Lage versetzt werden, zielgerichteter Informationen zu suchen und zu finden. Die Idee, die dahinter steht, ist die der *semantischen Suche*, das Ziel ist ein *semantisches Web* (engl. semantic web) (Hitzler u. a., 2008, S. 10). In diesem Zusammenhang gibt es Standardisierungsbemühungen, allen voran die des *World Wide Web Consortium* (W3C). Abbildung 1.1 zeigt eine mögliche Darstellung verwendeter Technologien zur Realisierung des Semantic Web.

Dabei wurden RDF (*Resource Description Framework*) und OWL (*Web Ontology Languange*) explizit für den Zweck eines Semantic Web entwickelt (Hitzler u. a., 2008, S. 11f). RDF ist eine Sprache zur Beschreibung von Ressourcen im Web, mit der Inhalte durch Angabe von Tripeln aus Subjekt, Prädikat und Objekt annotiert werden können (Jakus u. a., 2013, S. 23). Das Tripel *(Unternehmen, verkauft, Produkt)* enthält beispielsweise das Subjekt *Unternehmen*, das Prädikat *verkauft* und das Objekt *Produkt*. RDF Schema (RDFS) ist eine Erweiterung von RDF um die Fähigkeit, terminologisches Wissen einer Domäne auszudrücken (Hitzler u. a., 2008, S. 67), z.B. Subklassenbeziehungen zwischen den *Konzepten* einer Domäne wie zwischen *Buch* und *Sachbuch*. OWL besitzt eine größere Ausdrucksstärke als RDFS und basiert auf einer Beschreibungslogik, die eine formale Semantik besitzt und deshalb die Möglichkeit logischer Schlussfolgerungen anbietet. Darauf aufbauend können dann formale Beweise geführt werden, die es ermöglichen, Wissen aus einer *Ontologie* abzuleiten und Aussagen zu überprüfen. Die oberste Schicht kann dann Sicherheitsaspekte umsetzen, wie beispielsweise die Entscheidung, ob einem in der *Proof*-Schicht erbrachten Beweis gefolgt werden soll

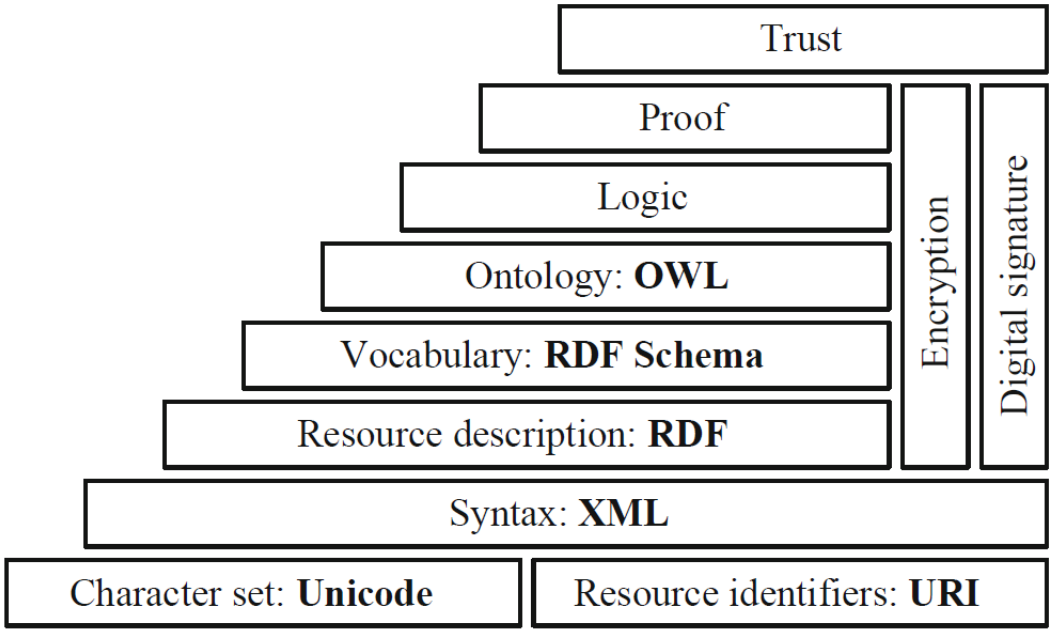

Abbildung 1.1: Der Semantic Web technology Stack, (Jakus u. a., 2013, S. 22)

oder nicht (Jakus u. a., 2013, S. 24). Die untersten Schichten dienen der Kodierung der Dokumente (XML, Unicode) und der Identifikation der Ressourcen (URI).

Um das Wissen einer Domäne zu formalisieren, muss dies in einer für Maschinen geeigneten Weise erfolgen. Eine Datenstruktur, die sich dafür anbietet, ist die Datenstruktur der *Ontologie*. Eine Ontologie besteht im Wesentlichen aus *Konzepten* und *Beziehungen* zwischen diesen Konzepten und stellt eine *Wissensbasis* dar. Im obigen Beispiel könnten die Suchbegriffe als Konzepte der Domäne *Arbeit* aufgefasst werden. Dann könnten z.B. die foglenden Beziehungen zwischen den Konzepten einer Domäne existieren:

(Architekt, ist ein, Beruf), (Maurer, ist ein, Beruf), (Beruf, ist eine, Arbeit), (Gehalt, in Form von, Geld), (Lohn, ist gleich, Gehalt), (Verdienst, ist gleich, Gehalt), (Arbeit, beinhaltet, Gehalt), (Arbeit, beinhaltet, Arbeitszeit), (Arbeit, ist eine, Beschäftigung)

Es ist möglich, Konzepten *Attribute* zuzweisen. So wäre es z.B. möglich, dem Konzept *Gehalt* das Attribut *Höhe* zuzuweisen und dem Konzept *Arbeitszeit* die Attribute *Von*

und *Bis*. Zusätzlich könnte eine einfache Funktion berechnen, wann etwas lukrativ ist. Dann wäre die folgende Kette von Schlussfolgerungen möglich, wenn die Suchbegriffe *lukrative Beschäftigung* in eine Suchmaschine eingegeben werden:

- Arbeit ist eine Beschäftigung → Suche auch nach *Arbeit*
- Arbeit beinhaltet Arbeitszeit → Suche auch nach *Arbeitszeit*
- Arbeit beinhaltet Gehalt → Suche auch nach *Gehalt*
- Lohn ist gleich Gehalt *und* Verdienst ist gleich Gehalt → Suche auch nach *Lohn* und *Verdienst*
- Beruf ist eine Arbeit → Beruf impliziert Gehalt, Lohn, Verdienst.
- Maurer ist ein Beruf, Architekt ist ein Beruf → Maurer und Architekt haben Gehalt, Verdienst, Lohn
- Ermittlung der Suchergebnisse…
- Wenn in den Suchergebnissen *Arbeitszeit, Gehalt, Architekt* etc. steht:
  $$\text{Stundenlohn} = \frac{\text{Gehalt.Höhe}}{\text{Arbeitszeit.Bis} - \text{Arbeitszeit.Von}}$$
- Stundenlohn $\geq 17€ \rightarrow$ lukrativ.
- Sortierung der Suchergebnisse, sodass die lukrativstien Beschäftigungen als erstes zu sehen sind.
- Anzeigen der Ergebnisse.

Diese schematische Darstellung verdeutlicht die Vorgehensweise der logischen Schlussfolgerungen.

Im Allgemeinen können Konzepte als jene Begriffe einer Domäne aufgefasst werden, die in dieser Domäne eine gewisse *Bedeutung* haben (Jakus u. a., 2013, S. 7f). Der Vorteil von Ontologien ist der, dass es für sie Beschreibungslogiken (engl. *Description Logics*) gibt, die eine formale Semantik haben (Krötzsch u. a., 2014, S. 3) und mit denen das Wissen einer Domäne formal beschrieben werden kann. Diese formale Semantik ermöglicht es, dass Softwareanwendungen aus dem Wissen, welches durch eine Ontologie formalisiert wurde, Folgerungen schließen können. Dies bedeutet,

sie können gewisse Sachverhalte überprüfen und gegebenenfalls neues Wissen aus diesen ableiten. Ein weiterer Vorteil ist der, dass sich Wissen auf diese Art leicht zwischen Computern austauschen lässt, die miteinander vernetzt sind. Dadurch wird die Interoperabilität von unterschiedlichen Systemen erhöht (Jakus u. a., 2013, S. 33).

Da der Prozess der Erzeugung einer Ontologie mit steigender Komplexität der jeweiligen Domäne jedoch sehr viel Zeit in Anspruch nehmen kann, wurden und werden Methoden entwickelt, die diesen Prozess unterstützen sollen. Der Forschungszweig, der sich mit diesen Methoden befasst, ist der des *Ontology Learning*. Zu diesem Forschungszweig gehört nach (Lehmann und Völker, 2014, S. Xf) auch der Zweig der Ontologieextraktion aus natürlichsprachlichen Texten, in dem mit Hilfe von Methoden des *Natural Language Processing* und des *Machine Learning* versucht wird, Ontologien voll- und semiautomatisch aus textuellen Ressourcen zu extrahieren.

Diese Arbeit soll den aktuellen Stand der Forschung auf dem Gebiet der Ontologieextraktion aus natürlichsprachlichen Texten aufzeigen. Der Aufbau der Arbeit ist wie folgt: Da vor der Ontologieextraktion klar sein sollte, was eine Ontologie genau ausmacht, wird in Kapitel 2 auf die Konzepte eingegangen, die einer Ontologie zugrundeliegen. In Kapitel 3 wird das Gebiet des Ontology Learning vorgestellt und darauf eingegangen, welche allgemeinen Fragestellungen auf diesem Gebiet existieren. Es wird ersichtlich, dass die Ontologieextraktion aus natürlichsprachlichen Texten einer von mehreren Wegen ist, den Ontologieerzeugungsprozess zu unterstützen. Kapitel 4 behandelt die Methoden, die eingesetzt werden können, um Ontologien aus natürlichsprachlichen Texten zu extrahieren. Um sicherzustellen, dass eine Ontologie gewissen Erwartungen genügt, muss diese evaluiert werden. Das Kapitel 5 befasst sich mit diesem Aspekt. In Kapitel 6 werden einige Methoden aus Kapitel 4 auf das deutsche Grundgesetz angewendet und die Ergebnisse werden besprochen. Die Ergebnisse der Arbeit werden in Kapitel 7 zusammengefasst und es wird ein Fazit gezogen.

2 Ontologien

Der Begriff der Ontologie hat seinen Ursprung in der Philosophie (Lim u. a., 2013, S. 6) und bezeichnet dort die Lehre des Seins oder der Existenz (Jakus u. a., 2013, S. 29). Neben der Anwendung im Bereich der Philosophie wird Ontologien auch in der Informatik eine hohe Aufmerksamkeit beigemessen.

In der Informatik befasst sich der Forschungszweig der künstlichen Intelligenz (KI) damit, Computerprogramme intelligenter zu machen. Eine grundlegende Frage dabei ist, wie Wissen im Computer dargestellt werden kann. Diese Leitfrage beschäftigt Forscher auf dem Gebiet des *Knowledge Engineering*, in welchem Methoden entwickelt werden, Wissen aus Daten zu extrahieren, indem diese Daten gewisse Verarbeitungsschritte durchlaufen, sodass am Ende dieses Prozesses diese Daten in eine für Maschinen und/oder Menschen lesbare Form transformiert werden (Lim u. a., 2013, S.4).

Eine Ontologie stellt eine solche Form der Wissens-Repräsentation dar. Weder in der Informatik noch in der Philosophie erfolgte bisher die Festlegung auf *eine* Definition dessen, was als Ontologie aufgefasst werden kann. Eine mögliche Definition wird in (Jakus u. a., 2013, S. 29) zitiert:

> *„An Ontology is a formal and explicit specification of a shared conceptualization."*

Die Autoren schlussfolgern aus dieser Definition vier Kerneigenschaften, die Ontologien aufweisen:

Konzeptualisierung. Ontologien bestehen aus Konzepten. Was Konzepte sind, wird in Abschnitt 2.2.1 erläutert.

Explizitheit. Konzepte und Relationen, die Konzepte miteinander in Beziehung setzen, werden explizit definiert. Relationen werden in Kapitel 2.2.2 näher betrachtet.

Formalismus. Dieser Aspekt spielt in der Informatik eine wichtige Rolle, da Ontologien in der Informatik von Softwareanwendungen verwendet werden. Dieser Aspekt wird in Abschnitt 2.4 verdeutlicht.

Geteilte Verwendung. Ontologien werden von mehreren Anwendern verwendet. Diese Anwender stimmen hinsichtlich der verwendeten Konzepte und Relationen, die in der Ontologie repräsentiert sind, überein - dies wird auch als *ontological commitment* bezeichnet.

So wie Dokumente nach den Themen, die sie behandeln, klassifiziert werden können, können auch Ontologien nach ihrem Spezialisierungsgrad klassifiziert werden. Dieser Aspekt wird in Abschnitt 2.1 verdeutlicht. Abschnitt 2.2 beschreibt, woraus Ontologien bestehen. Zur Veranschaulichung von Ontologien haben sich bestimmte Formen der Visualisierungen in der Informatik etabliert, die in Abschnitt 2.3 vorgestellt werden. Um Ontologien für Softwareanwendungen nutzbar zu machen, ist die oben erwähnte Eigenschaft des *Formalismus* wichtig. Dieser wird durch *Beschreibungslogiken* gewährleistet, die in Abschnit 2.4 vorgestellt werden.

2.1 Klassifikationen

Es können zwei Arten von Ontologien unterschieden werden: **Allgemeine Ontologien** und **spezifische Ontologien** (Jakus u. a., 2013, S. 31f). Allgemeine Ontologien können auch als *allgemein gehaltene* Ontologien bezeichnet werden, d.h. sie beinhalten kein Wissen, das so speziell ist, dass sich dafür nur eine Anwendungsdomäne finden lässt. Allgemeine Ontologien werden in der Literatur darüber hinaus auch als *top-level, general-, upper-, foundation-, common-* sowie *core-ontologys* bezeichnet. Allgemeine Ontologien weisen unter anderem die Eigenschaft auf, auf Konzepte beschränkt zu sein, die universell, generisch und abstrakt sind (Lim u. a., 2013, S. 8). Aus diesen allgemeinen Konzepten können dann spezifische Ontologien konstruiert werden. Als ein wichtiger Nutzen allgemeiner Ontologien wird unter anderem die *Interoperabilität* genannt. Wenn zwei oder mehr Systeme mit spezifischen Ontologien arbeiten, können diese über die Konzepte einer allgemeinen Ontologie Wissen austauschen. In der Vergangenheit gab es diverse Versuche, solch allgemeine Ontologien einer

breiten Nutzergemeinschaft zugänglich zu machen. In (Lim u. a., 2013, S. 8) werden hier die allgemeinen Ontologien *SUMO* und *OpenCyc* genannt. SUMO fand laut Schoening (2015) allerdings nie wirklich Konsens zwischen allen Beteiligten. OpenCyc liegt zum Zeitpunkt der Arbeit in Version 4.0 vor und umfasst ca. 239.000 Konzepte und Subkonzepte sowie über 2 Millionen Relationen.

Spezifische Ontologien werden in der Literatur auch *domänenabhängige* bwz. *domänenspezifische Ontologien* genannt, da sie Konzepte beinhalten, die von einer bestimmten Domäne, Anwendung, Aufgabe, Aktivität, Methode etc. abhängig sind (Jakus u. a., 2013, S. 32). Eine domänenabhängige Ontologie kann die Erweiterung einer top-level-Ontologie darstellen (Lim u. a., 2013, S. 11). Hier wird außerdem darauf aufmerksam gemacht, dass domänenabhängige Ontologien für Softwareanwendungen besser geeignet sind als allgemeine Ontologien, da Softwareanwendungen in der Regel ebenfalls auf spezifische Domänen zugeschnitten sind.

2.2 Komponenten

2.2.1 Konzepte

Ein wesentlicher Bestandteil von Ontologien sind *Konzepte*. Mit der Frage danach, welche Eigenschaften Konzepte aufweisen, sind die Gebiete der Philosophie, der Psychologie und der Linguistik befasst (Jakus u. a., 2013, S. 5). Neben einigen Unterschieden in der Auffassung dessen, was ein Konzept ausmache, lassen sich jedoch zwei grundlegende Gemeinsamkeiten ausmachen (Jakus u. a., 2013, S. 5f):

Abstraktheit. Konzepte sind abstrakt, da kein Konzept für sich genommen die Unterscheidungsmerkmale aufweist, die die Objekte aufweisen, auf die das Konzept angewendet wird.

Universalität. Konzepte sind universell, bezogen auf die Objekte, auf die sie angewendet werden.

Diese beiden Eigenschaften werden am Konzept *männlich* deutlich: Sowohl Menschen als auch Tiere können männlich sein (Universalität). Gleichzeitig gibt es Unterscheidungsmerkmale zwischen Mensch und Tier, die durch das Konzept *männlich* aber

nicht ausgedrückt werden (Abstraktheit). Im Allgemeinen gibt es keine Einschränkungen hinsichtlich der Objekte, auf die Konzepte angewendet werden, sodass diese Objekte real oder imaginär sowie atomar oder zusammengesetzt sein können (Jakus u. a., 2013, S. 6). Wenn es darum geht zu beurteilen, ob ein bestimmtes Wort, das in einer Domäne beobachtet wurde, ein Konzept ist, dann muss beurteilt werden, ob dieses Wort für diese Domäne „wichtig“ ist - oder ob es so allgemein ist, dass es auch in jedem anderen Kontext auftreten kann. Diese Problemstellung wird in Kapitel 4.3 behandelt.

2.2.2 Relationen

Als *semantische Relationen* zwischen Konzepten werden Beziehungen bezeichnet, die zwischen zwei oder mehr Konzepten bestehen (Jakus u. a., 2013, S. 12). Wenn eine semantische Relation zwei Konzepte miteinander in Beziehung setzt, wird diese Relation als *binäre Relation* bezeichnet - mehrstellige Relationen können jedoch in binäre Relationen umgewandelt werden (Jakus u. a., 2013, S. 12). In Systemen der Wissensorganisation (engl. *Knowledge Organization Systems*) können zwei grundlegende Arten semantischer Relationen unterschieden werden: *Paradigmatische Relationen* und *syntagmatische Relationen* (Peters und Weller, 2008, S. 100):

Paradigmatische Relationen sind semantische Relationen, die fest an die Konzepte gekoppelt sind, die sie in Beziehung setzen.

Syntagmatische Relationen sind semantische Relationen, die in Bezug auf die Konzepte die sie in Beziehung setzen, vom jeweiligen Kontext, in dem sie beobachtet wurden, abhängig sind.

Paradigmatische Relationen haben die Eigenschaft, dass eine einmal beobachtete paradigmatische Beziehung zwischen zwei oder mehr Konzepten, in jedem anderen Kontext auf diese Konzepte angewendet werden kann. Dies ist beispielsweise bei der Relation *artVerwandt* und den Konzepten *Wolf* und *Fuchs* der Fall. Paradigmatische Relationen sind also *kontextunabhängige Relationen*. Syntagmatische Relationen sind *kontextabhängig*. Wird der Satz *„Hans fährt Fahrrad.“* betrachtet können die Konzepte

Mensch und *Vehikel* (*Hans* ist ein *Mensch* und *Fahrrad* ist ein *Vehikel*) in der syntagmatischen Relation *fährt* miteinander in Beziehung stehen. Wenn in einer anderen Situation wieder eines dieser beiden Konzepte beobachtet wird, kann jedoch nicht davon ausgegangen werden, dass der Mensch das jeweilige Vehikel fährt, da diese Relation kontextabhängig war und nicht von einer Situation auf eine andere übertragen werden kann.

Die Autoren Peters und Weller (2008) geben in Abbildung 2.1 einen Überblick über mögliche Relationen, die in Ontologien und Folksonomien (engl. *Folksonomy*) auftreten. Folksonomien stellen alle Schlüsselwörter, die in einem System von dessen Anwendern auf Inhalte angewendet wurden, dar. Dieses Vorgehen wird auch als *soziale Verschlagwortung* (engl. *social tagging*) bezeichnet und kann in Sozialen Netzwerken wie Twitter (Hashtag) und Instagram beobachtet werden.

Nach Abbildung 2.1 können semantische Relationen in drei Kategorien eingeteilt werden: *Relationen der Äquivalenz*, *Relationen der Hierarchie* und *Relationen der Assoziation*. Relationen der Äquivalenz beinhalten *Synonyme, Quasi-Synonyme* und *Gen-Identität*. Zu den Synonymen zählen unterschiedliche Begriffe, die dasselbe ausdrücken. Quasi-Synonyme sind unterschiedliche Begriffe, die streng genommen auch unterschiedliche Bedeutungen haben, aber im jeweiligen Kontext als Synonym aufgefasst werden (Stock, 2009, S. 411). Die Gen-Identität bezeichnet Begriffe, die in einer schwächeren Form als die eigentliche Identität als identisch betrachtet werden, wobei der Begriff der Gen-Identität einen klaren Bezug zur Zeit hat (Stock, 2009, S. 411). Dies bedeutet, bezogen auf Objekte, dass ein Objekt, das über die Zeit verschiedene Stadien, z.B. die Alterung, durchläuft, gen-identisch ist.

Relationen der Hierarchie beinhalten *Hyponyme* und *Meronyme*. Zusätzlich gibt es auch noch die taxonomischen Beziehungen der *Holonyme*, *Hypernyme* und *Antonyme* (Lim u. a., 2013, S. 19). Diese können demnach wie folgt definiert werden:

Hypernyme - W_1 ist ein Hypernym von W_2 bedeutet, dass W_2 in einer „ist-eine-Art-von“ Beziehung zu W_1 steht.

Hyponyme - W_1 ist Hyponym von W_2 bedeutet, dass W_1 in einer „ist-eine-Art-von“ Beziehung zu W_2 steht.

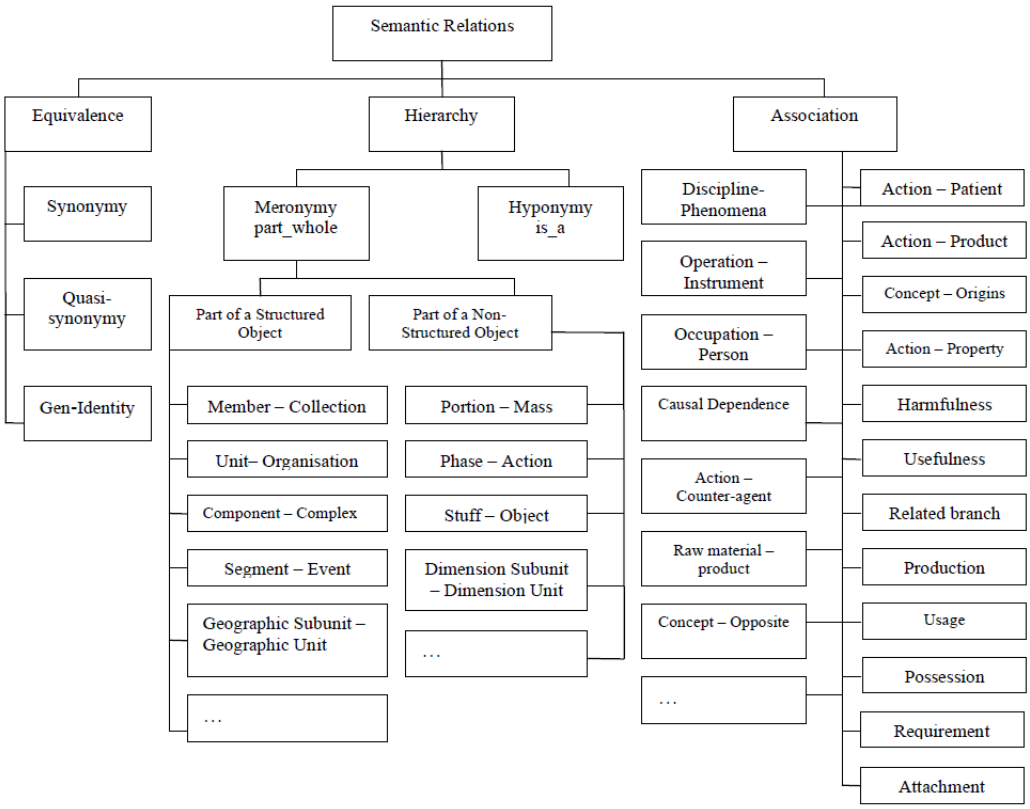

Abbildung 2.1: Semantische Relationen, (Peters und Weller, 2008, S. 105)

Meronyme - W_1 ist Meronym von W_2 wenn W_1 „ein-Teil-von“ W_2 ist.

Holonyme - W_1 ist ein Holonym von W_2 wenn W_2 „ein-Teil-von“ W_1 ist.

Antonyme - Antonyme sind das semantische Gegenteil von Wörtern. Als Beispiele werden hier „hapiness“ und „unhappiness“ sowie „man“ und „woman“ genannt. W_1 ist Antonym von W_2 wenn W_1 nicht W_2 ist.

Abbildung 2.1 zeigt darüber hinaus auf, dass Meronyme Teil von strukturierten wie auch unstrukturierten Objekten sein können, wobei die darunter aufgeführten Objekte als Beispiele angesehen werden können.

Die Relationen der Assoziation stellen Beziehungen zwischen Konzepten dar, die anwendungsabhängig sind. Die in Abbildung 2.1 aufgeführten Relationen beziehen

sich zum einen auf die Arbeit von Clarke (2001) und zum anderen auf eine Reihe von Anwendungsfällen, die in Peters und Weller (2008) hinsichtlich ihrer Beziehungen analysiert wurden. Auch wenn in der Literatur noch nicht viele Versuche unternommen wurden, assoziative Relationen zu spezifizieren, wird der Ansatz von Clarke (2001) in (Peters und Weller, 2008, S. 102) näher vorgestellt und ist in Tabelle 2.1 zu sehen.

Assoziative Relation	**Beispiel**
Terme mit überlappender Bedeutung	Schiffe und Boote
Disziplin und Phänomen	Seismologie und Erdbeben
Prozess und Instrument	Geschwindigkeitsmessung und Tacho
Beschäftigung und Beschäftigte/r	Buchhaltung und Buchhalter
Aktion und Produkt der Aktion	Straßenbau und Straße
Aktion und betroffene Person	Lehren und Studentin
Konzept und dessen Herkunft	Wasser und Wasserfall
Kausale Abängigkeit	Erosion und Verschleiß
Ding bzw. Aktion und Gegenmittel	Schädlinge und Pestizide
Rohes Material und das erzeugte Produkt	Felle und Leder
Aktion und eine damit verbundene Eigenschaft	Präzisionsmessung und Präzision
Konzept und Gegenteil	Toleranz und Vorurteil

Tabelle 2.1: Assoziative Relationen nach (Peters und Weller, 2008, S. 102)

In Tabelle 2.1 fällt auf, dass die assoziativen Relationen der *Terme mit überlappender Bedeutung* auch als Quasi-Synonyme aufgefasst werden können. Relationen des *Konzeptes und Gegenteils* können auf dem Gebiet der Sprachwissenschaft auch als *Antonym* aufgefasst werden.

Generalisierbare Relationen spielen im Bereich der Wissensrepräsentation eine herausragende Rolle (Peters und Weller, 2008, S. 100). Diese Relationen entsprechen paradigmatischen Relationen, die sowohl in allgemeinen als auch in spezielleren Domänen angewendet werden können. Zu den generalisierbaren Relationen zählen demnach *Relationen der Äquivalenz, hierarchische Relationen* und *Assoziative Relationen.*

Wenn eine Ontologie nur hierarchische Beziehungen in Form von Hyponymen/Hypernymen besitzt, dann spricht man statt von einer Ontologie auch von einer **Taxonomie**. Eine wichtige Eigenschaft von Ontologien ist also, dass sie nicht nur Hy-

pernym/Hyponym Beziehungen zwischen den Konzepten zulässt bzw. aufweist, sondern auch Relationen der Äquivalenz und/oder Assoziation. Das geläufigste Beispiel einer Taxonomie ist die Darstellung eines *Stammbaums*.

2.2.3 Weitere Komponenten

Neben *Konzepten* und *Relationen* beinhalten Ontologien oft weitere Komponenten (Jakus u. a., 2013, S. 31):

- Objekte (auch: Individuen, Instanzen oder Entitäten)
- Attribute (auch: Eigenschaften) von Konzepten und Objekten und
- Attributwerte.

Objekte, Individuen, Instanzen oder Entitäten werden synonym verwendet und bezeichnen die „Dinge“, auf die Konzepte angewendet werden können. Wenn beispielsweise *Anna* eine Frau und Mutter ist, dann ist Anna eine Instanz der Konzepte *Frau* und *Mutter*. Attribute und Eigenschaften können bestimmte Attributwerte annehmen und versehen dadurch Konzepte und Instanzen mit zusätzlichen Unterscheidungsmerkmalen und Charakteristiken. Das Konzept *Frau* könnte z.B. das Attribut *name* mit dem Attributwert *Zeichenkette* aufweisen.

2.3 Visualisierungen

Ontologien können durch *Konzeptgraphen* (engl. *conceptual graphs*) und *Semantische Netze* (engl. *semantic networks*) dargestellt werden (Jakus u. a., 2013, S. 33). Die folgenden Unterabschnitte stellen diese beiden Formalismen vor.

2.3.1 Konzeptgraphen

Konzepte werden in Konzeptgraphen in der Regel als Rechtecke und Relationen als Kreise oder Ovale dargestellt (Jakus u. a., 2013; Dengel, 2012, S. 14, S. 88). Pfeile führen von einem Konzept über eine Relation zu einem weiteren Konzept und stellen

[Concept_1] → (relation) → [Concept_2].

Abbildung 2.2: Konzeptgraph nach (Polovina, 2007, S. 2)

gleichzeitig die Leserichtung dar. Diese können über einen textuellen Zeichensatz oder durch Grafiken dargestellt werden (Polovina, 2007, S. 2). Die textuelle Visualisierung wird gelesen als „Die *relation* eines *Concept_1* ist ein *Concept_2*. Zu beachten ist der Punkt am Ende, der das Ende des Konzeptgraphen darstellt. Konzepte können außerdem typisiert sein, sodass

[Employee : Simon].

Abbildung 2.3: Typisierung von Konzepten nach (Polovina, 2007, S. 2)

bedeutet, dass *Simon* dem *Typlabel Employee* entspricht. Darüber hinaus können Konzepte *generisch* sein, sodass die Instanz *Simon* nicht genannt werden muss und stattdessen nur

[Employee] oder [Employee : *]

verwendet wird. Eine wichtige Eigenschaft von Konzeptgraphen ist, dass mit ihnen logische Schlüsse abgeleitet werden können. Dafür wird das Prinzip der *Projektion* verwendet. Die Projektion wird nach (Polovina, 2007, S. 4) in Verbindung mit spezialisierten - also domänenspezifischen - Konzeptgraphen verwendet. Ein Konzeptgraph wird demnach spezialisierter, wenn mindestens einer der folgenden Fälle eintritt:

1. Zur Einschränkung des Anwendungsbereichs des Konzeptgraphen werden mehr Konzepte, Typen und Relationen hinzugefügt, oder
2. der Konzeptgraph erhält statt generischer Konzepte ihr konkretes Pendant, oder
3. Subtypen von Konzepten ersetzen ihre jeweiligen Supertypen, wodurch der allgemeinere Aspekt der Konzeptgraphen verloren geht.

Allgemeinere Konzeptgraphen können durch diese drei Fälle zu spezialisierteren Konzeptgraphen transformieren. Im Allgemeinen kann es mehrere spezialisierte Varianten solcher allgemeinen Konzeptgraphen geben. Umgekehrt kann es für einen spezialisierten Konzeptgraphen auch allgemeinere Konzeptgraphen geben, aus denen der spezialiserte Graph erzeugt wurde. Wenn es einen allgemeinen und einen spezialisierten Konzeptgraphen gibt und der spezialisierte Konzeptgraph aus dem allgemeinen erzeugt wurde, „projiziert“ der allgemeine in den spezialisierten Konzeptgraphen. Der spezialisierte Konzeptgraph ist dann die Projektion des allgemeinen Konzeptgraphen (Polovina, 2007, S. 4). In der logischen Inferenz von Konzeptgraphen geht es dann u.a. darum, zu zeigen, dass ein Graph aus einem anderen durch Projektion geschlussfolgert werden kann und dass zwei Graphen zu einem weiteren Graphen zusammengefügt werden können (Polovina, 2007, S. 7ff).

Durch Konzeptgraphen können beliebige Sätze natürlicher Sprache grafisch in logische Ausdrücke überführt werden sowie alle Ausdrücke der Prädikatenlogik erster Stufe visuell repräsentiert werden (Dengel, 2012, S.88f). Um Inferenzen auf Konzeptgraphen anwenden zu können, werden diese in Beschreibungslogiken überführt (Dengel, 2012, S. 89), die in Abschnitt 2.4 vorgestellt werden.

2.3.2 Semantische Netze

Wie Konzeptgraphen auch, bestehen semantische Netze aus Konzepten und Relationen. In Bezug auf Sowa (2015) werden Klassifikationen semantischer Netze in (Dengel, 2012, S. 75ff) wie folgt beschrieben:

Definitionsnetze kategorisieren Konzepte hierarchisch und entsprechen deshalb Taxonomien oder Partonomien. Die Relationen, die Definitionsnetze aufweisen, entsprechen also nur *ist-ein(e)* Relationen (Taxonomien) oder *hat-Teil* bzw. *ist-Teil-von* Relationen (Partonomien).

Propositionale Netze erweitern taxonomische und partonomische Relationen um assoziative Relationen (vgl. Abschnitt 2.2.2), wie z.B. `X ermittelt-Wert-von Y`.

Implikationsnetze sind azyklische, gerichtete Graphen, die kausale Abhängigkeiten zwischen Ereignissen repräsentieren. Sie können dafür verwendet werden, Ursachen eines Ereignisses zu schlussfolgern, wie dies beispielsweise in *Bayesschen Netzen* erfolgt.

Ausführbare Netze erlauben die Dynamik eines Systems oder eines Prozesses abzubilden, wie dies beispielsweise in *Petri-Netzen* erfolgt.

Lernende Netze transformieren sich mit der Zeit, indem Relationen und Konzepte in ihnen hinzugefügt oder entfernt werden können.

Hybride Netze kombinieren andere Typen semantischer Netze. Als Beispiel wird hier die Kombination eines Baysschen Netzes und eines lernenden Netzes aufgeführt, sodass das Bayssche Netz durch das lernende Netz lernfähig sein könnte.

Diese Aufzählung verdeutlicht das breite Einsatzgebiet semantischer Netze. Diese können wie Konzeptgraphen auch, für die Modellierung von Ontologien verwendet werden, da sie auf dem Modell der Konzeptgraphen basieren (Dengel, 2012, S. 88).

2.4 Sprachen

2.4.1 Beschreibungslogiken

Um die in Kapitel 2 beschriebene Formalismuseigenschaft von Ontologien zu erreichen, werden Beschreibungslogiken verwendet. Diese Beschreibungslogiken stellen oftmals eine Teilmenge der Prädikatenlogik dar und sind im Gegensatz zu dieser entscheidbar (Krötzsch u. a., 2014, S. 3). Eine Menge M ist entscheidbar, wenn es eine algorithmisch arbeitende Maschine gibt, die für ein Element $m \in M$ in endlicher Zeit beantworten kann, ob dieses Element zur Menge M gehört oder nicht (Hoffmann, 2013, S. 290f). Im Jahr 1936 bewies Alan Turing, dass es ein solches Entscheidungsverfahren für die Prädikatenlogik (erster Stufe) nicht gibt (Hoffmann, 2013, S. 307). Beschreibungslogiken, die entscheidbar sind, bieten deshalb den Vorteil, für Aussagen $m \in M$ beantworten zu können, ob m eine Aussage ist, die sich durch die Ontologie M beweisen lässt oder nicht. Die Entscheidbarkeit von Beschreibungslogiken erfolgt allerdings zu dem

Preis geringerer Ausdrucksfähgikeit im Vergleich zur Prädikatenlogik. In der Prädikatenlogik gibt es u.a. *Konstanten* und *Prädikate*. Im Kontext von Beschreibungslogiken entsprechen Konstanten dann den Namen der *Individuen* einer Domäne, unäre Prädikate entsprechen den *Konzepten* einer Domäne und binäre Prädikate entsprechen den *Rollen* bzw. Relationen der Domäne (Krötzsch u. a., 2014, S. 3).

Open World Assumption vs. Closed World Assumption

In Bezug auf Logiken bezeichnet der Begriff *Wissensbasis* (WB) eine Menge von Fakten und Regeln die in einer Datenbank gespeichert sind. Ein Inferenzsystem kann eine solche WB nutzen, um zu überprüfen, ob sich bestimmte Aussagen mit dieser WB beweisen lassen. Wenn eine Aussage bewiesen werden kann, dann wurde diese Aussage aus der WB abgeleitet und kann als neues Wissen, das sich jedoch implizit bereits in der WB befand, dieser hinzugefügt werden. Sollte sich eine Aussage jedoch nicht ableiten lassen, dann stellt sich die Frage, wie damit umgegangen wird. Zwei Vorgehensweisen können unterschieden werden: Die *Open World Assumption* (OWA) und die *Closed World Assumption* (CWA).

In der CWA wird davon ausgegangen, dass sich alles Wissen, das im Zuge des Inferenzverfahrens zu berücksichtigen ist, in der Wissensbasis befindet. Es handelt sich bei der Wissensbasis also um eine „geschlossene Welt“ und was sich nicht beweisen lässt, muss folglich falsch sein. In der OWA wird angenommen, dass jederzeit neues Wissen zu der Wissensbasis hinzukommen kann, das sich nicht bereits aus ihr ableiten lässt. Die OWA geht weder davon aus, dass sich Aussagen, die sich nicht beweisen lassen, falsch sind, noch dass sie wahr sind. In Bezug auf Beschreibungslogiken für Ontologien wird die OWA vertreten (Krötzsch u. a., 2014, S. 12). Einer der Gründe hierfür ist, dass Ontologien im Kontext des World Wide Web mit sich stetig verändernden Informationsressourcen arbeiten, wie bereits in Kapitel 1 erwähnt wurde.

Axiome

Im Kontext von Beschreibungslogiken spielen *Axiome* eine entscheidende Rolle. Ein *Axiom* ist in der Mathematik ein Satz, der als wahr angenommen wird, ohne dass ein Beweis für diesen vorausgesetzt wird (Schubert, 2009, S. 21). Bei Axiomen handelt es

sich um Sätze, auf denen direkt oder indirekt alle weiteren Beweise einer Domäne beruhen. Das Bewusstsein darüber, welche Axiome in einer Beschreibungslogik gültig sind, ist also wesentlich, um beurteilen zu können, wie ausdrucksstark eine Beschreibungslogik ist. Eine mögliche Klassifikation solcher Axiome erfolgt in (Krötzsch u. a., 2014, S. 4ff):

ABox Axiome. Diese Klasse umfasst Axiome, die Aussagen über Konzepte (*concept assertions*) und Rollen (*role assertions*) treffen.

TBox Axiome beschreiben terminologisches Wissen. Durch Teilmengenbeziehungen zwischen Konzepten können hierarchische Beziehungen zwischen diesen Konzepten ausgedrückt werden. Dies wird als *concept inclusion* und manchmal als *subsumption* bezeichnet. Durch die Äquivalenz von Konzepten (*concept equivalence*) kann ausgedrückt werden, dass zwei Konzepte dieselben Instanzen besitzen.

RBox Axiome. Durch RBox Axiome können Subrollenbeziehungen (*role inclusion/role subsumption*) und Rollenäquivalenz (*role equivalence*) ausgedrückt werden. Außerdem gehört zu dieser Kategorie von Axiomen die (komplexe) Rollenkomposition (*(complex) role composition*). Zu den RBox Axiomen zählen darüber hinaus auch Axiome, die die Disjunktheit von zwei Rollen ausdrücken (*disjoint roles*) und Rollencharakteristiken (*role characteristics*) wie z.B. die Reflexivität, beschreiben.

Tabelle 2.2 zeigt einige Beispiele dieser Axiome. Zu beachten ist in Tabelle 2.2, dass Konzeptbezeichnungen mit einem großen Anfangsbuchstaben beginnen und Rollenbezeichnungen mit einem kleinen. Beispiel 5 bedeutet „Alle Brüder sind auch Familienmitglieder“. Beispiel 7 ist so zu verstehen, dass alle Paare von Individuen, die über die Rolle *vorgesetzterVon* miteinander in Beziehung stehen, auch in der Rolle *weisungsbefugterVon* miteinander in Beziehung stehen. Die einfache Rollenkomposition hat die Ergebnismenge $onkelVon \circ vaterVon = \{(x,z)|(x,y) \in onkelVon \wedge (y,z) \in vaterVon\}$. Die komplexe Rollenkomposition liefert als Ergebnismenge $onkelVon \circ vaterVon \sqsubseteq großonkelVon = \{(x,z)|(x,y) \in onkelVon \wedge (y,z) \in vaterVon \wedge (x,z) \in großonkelVon\}$. Das *disjoint* Axiom legt fest, dass niemand Onkel und Bruder desselben Individuums sein kann.

Nr.	Klasse	Axiom	Beispiel
1	ABox	concept assertion	Bruder(bob)
2	ABox	role assertion	onkelVon(luke, bob)
3	ABox	individual inequality	luke $\not\approx$ bob
4	ABox	individual equality	luke $\approx$ bob
5	TBox	concept inclusion	Bruder $\sqsubseteq$ Familienmitglied
6	TBox	concept equivalence	Haus $\equiv$ Gebäude
7	RBox	role inclusion	vorgesetzterVon $\sqsubseteq$ weisungsbefugterVon
8	RBox	role equivalence	vorgesetzterVon $\equiv$ weisungsbefugterVon
9	RBox	role composition	onkelVon $\circ$ vaterVon
10	RBox	complex role composition	onkelVon $\circ$ vaterVon $\sqsubseteq$ großonkelVon
11	RBox	disjoint	disjoint(onkel, bruder)

Tabelle 2.2: Beispiele von ABox, TBox und RBox Axiomen

Konstruktoren

Die Ausdrucksfähigkeit einer Beschreibungslogik kann durch Hinzufügen weiterer Konzept- und Rollenoperationen sowie Quantoren erhöht werden. Tabelle 2.3, Tabelle 2.4 und Tabelle 2.5 fassen die in (Krötzsch u. a., 2014, S. 6ff) weitergeführten Axiome zusammen.

Nr.	Operation	Beispiel
1	Schnittmenge/Konjunktion	Männlich $\sqcap$ Familienmitglied
2	Vereinigung/Disjunktion	Männlich $\sqcup$ Weiblich
3	Komplement	Arbeitslos $\sqcap \neg$ Verheiratet
4	Top Konzept	$\top \sqsubseteq$ Tot $\sqcup$ Lebendig
5	Bottom Konzept	Tot $\sqcap$ Lebendig $\sqsubseteq \bot$

Tabelle 2.3: Boolsche Konzeptkonstruktoren

Die Schnittmenge (Tabelle 2.3) fasst alle Individuen in einer Menge zusammen, die sowohl männlich als auch Familienmitglied sind. Die Vereinigungsmenge fasst jene Individuen in einer Menge zusammen, die weiblich oder männlich sind. Das Komplement ermittelt alle Individuen, die dem Konzept, auf das es angewendet wird, nicht zugehörig sind. Die Schnittmenge von *Arbeitslos* und $\neg$ *Verheiratet* enthielte im Beispiel dann die Individuen, die arbeitslos und nicht verheiratet sind. Das Top-Konzept

ist eine vereinfachte Darstellung von $C \sqcup \neg C$ und bedeutet in seiner Semantik, dass dieses Konzept alle Individuen einer Domäne als Instanz hat (Krötzsch u. a., 2014, S. 7). Im Beispiel bedeutet dies also, dass alle Individuen der betrachteten Domäne tot oder lebendig sind. Das Bottom Konzept ist eine abkürzende Schreibweise für $C \sqcap \neg C$ und hat kein Individuum als Instanz (Krötzsch u. a., 2014, S. 7). Das Beispiel drückt dann also aus, dass kein Individuum tot und lebendig gleichzeitig ist.

Nr.	**Operation**	**Beispiel**
6	Existentielle Restriktion	$\exists$ vaterVon.$\top$
7	Universelle Restriktion	$\forall$ vaterVon.Männlich
8	Zahlen Restriktion (at-least)	≥ 3 vaterVon.Weiblich
9	Zahlen Restriktion (at-most)	≤ 2 vaterVon.Weiblich
10	Lokale Reflexivität	$\exists$ vorgesetzterVon.Self

Tabelle 2.4: Rollenrestriktionen

Die existentielle Restriktion (Tabelle 2.4) wendet die Rolle in ihrem Skopus auf höchstens ein Individuum an, das über diese Rolle erreicht wird. Das Top Konzept im Beispiel (6) würde dann dafür sorgen, dass alle Individuen ermittelt werden, die Vater von jemandem sind. Die Universelle Restriktion im Beispiel (7) würde dies genauso ermitteln, wenn die Einschränkung *Männlich* auf der Ergebnismenge nicht erfolgen würde. Diese hat zur Folge, dass nur solche Väter ermittelt werden, die nur Söhne haben - oder keine Söhne, wie in (Krötzsch u. a., 2014, S. 7) analog zu dem Beispiel angemerkt wird. Die at-least Zahlenrestriktion schränkt die Ergebnismenge von Individuen im Beispiel (8) auf solche Väter ein, die mindestens 3 Töchter haben. Im Beispiel (9) sind in der Ergebnismenge der Väter dann nur solche Väter enthalten, die Vater von höchstens 2 Mädchen sind. Die lokale Reflexivität *Self* bezieht eine Rolle dann auf sich selbst. Im Beispiel bedeutet dies, dass es mindestens ein Individuum gibt, das sein eigener Vorgesetzter ist.

In Tabelle 2.5 kann die Inverse Rolle ausdrücken, dass eine Rolle das Inverse einer anderen Rolle ist, d.h. eine Rolle $r_1(x, y) \equiv \overline{r_2(y, x)}$ bedeutet $\overline{r_2(y, x)} = r_2(x, y) = r_1(x, y)$. Wenn also Manfred der Arbeitgeber von Günther ist, dann ist Günther der Arbeitnehmer von Manfred. Nominale ermöglichen es, mehrere Instanzen in einem Konzept zusammenzufassen.

Nr.	Operation	Beispiel
11	Inverse Rolle	arbeitgeberVon $\equiv \overline{\text{arbeitnehmerVon}}$
12	Universelle Rolle	Alle Individuen einer Domäne in Beziehung
13	Nominale	SG1 $\equiv$ {o'Neil} $\sqcup$ {carter} $\sqcup$ {jackson} $\sqcup$ {teal'c}

Tabelle 2.5: Inverse Rolle, Universelle Rolle und Nominale

Eine Ontologie ist *inkohärent*, wenn und nur wenn es ein Konzept in ihr gibt, zu dem es keine Instanzen (Individuen) gibt (Zhu u. a., 2015, S. 31).

Namenskonventionen

Im Kontext von Beschreibungslogiken orientiert sich deren Namensgebung an den Konstruktoren und Axiomen, die sie erlauben (Krötzsch u. a., 2014, S. 14). Implementierungssprachen für Ontologien basieren zum Teil auf diesen Beschreibungslogiken und in deren Spezifikation wird auf diese Bezug genommen. Die Kenntnis dieses Namenssystems hilft also dabei, die Ausdrucksfähigkeit der Beschreibungslogik, die einer Ontologiesprache zugrundeliegt, zu beurteilen. Einige der gebräuchlichen Buchstaben zur Namensgebung von Beschreibungslogiken sind in (Krötzsch u. a., 2014, S. 14f) aufgeführt und ergeben sich wie folgt:

$\mathcal{I}$ steht für inverse Rollen.

$\mathcal{O}$ steht für Nominale.

$\mathcal{Q}$ steht für qualifizierte Zahlenrestriktionen (qualified number restrictions).

$\mathcal{H}$ steht für Rollenhierarchien (d.h. Rolleninklusionsaxiome ohne Komposition)

$\mathcal{R}$ steht für das Vorhandensein von Rolleninklusionen, lokaler Reflexivität $Self$ und der universellen Rolle U sowie die zusätzlichen Rollencharakteristiken Transitivität, Symmetrie, Asymmetrie, disjunkte Rollen, Reflexivität und Irreflexivität.

2.4.2 Implementierungssprachen für Ontologien

Die Menge gültiger Axiome und Konstruktoren einer Beschreibungslogik ist ein wesentliches Unterscheidungsmerkmal von Beschreibungslogiken. Eine der bekanntesten Implementierungssprachen für Ontologien ist *OWL* (*Web Ontology Langauge*), die es bereits in der Version *OWL 2* gibt.

Auch wenn OWL 2 die Axiome und Konstruktoren, die Beschreibungslogiken zugrundeliegen können, umsetzt, weicht die Namensgebung dieser von denen in Abschnitt 2.4.1 erwähnten teilweise ab. So werden Konzepte als *Klassen* bezeichnet (engl. classes) und Relationen als *Objekt-Eigenschaften* dieser Klassen (engl. object properties). Assoziative Relationen können nicht direkt zwischen zwei Klassen definiert werden, sondern nur über deren Instanzen. Es gibt zwei Kategorien von Klasseneigenschaften/Relationen: *Object Properties* und *Datatype Properties*. Durch Object Properties können Objekte (d.h. Individuen) miteinander in Beziehung gesetzt werden, durch Datatype Properties können Datentypen für Objekte angegeben werden, wie z.B. Zeichenketten (String). Eine OWL-Ontologie besteht grundlegend aus Axiomen, welche von dem zugrundeliegenden Inferenzsystem als wahr angenommen werden und es gilt die Open World Assumption. Wenn einer OWL-Ontologie eine neue Relation, ein neues Konzept oder eine neue Instanz hinzugefügt werden soll, dann muss dies über die Definition von Axiomen erfolgen. OWL 2 bietet u.a. Ausdrucksmöglichkeiten für die Konzept- und Rolleninklusion, Rollen- und Zahlenrestriktionen sowie für die Equality und Inequality von Individuen an. Eine weiterführende Betrachtung von OWL 2 kann Hitzler u. a. (2012) entnommen werden.

OWL 2 gibt es in mehreren Varianten. Der Grund dafür ist der, dass OWL 2 hinsichtlich der Ausdrucksstärke von einigen Interessengruppen als zu komplex betrachtet wurde, um einfach und effizient Ontologien zu implementieren. Aus diesem Grund wurden *OWL 2 DL* und *OWL 2 Full* entwickelt, denen unterschiedliche Semantiken zugrundeliegen. OWL 2 DL entspricht weitestgehend der Beschreibungslogik $\mathcal{SROIQ}$ und bietet deshalb auch die in Abschnitt 2.4.1 beschriebenen Axiome und Konstruktoren an (Hitzler u. a., 2012, Kap. 9). OWL 2 Full liegt eine RDF basierte Semantik zugrunde (RDF = *Resource Description Framework*).

2.5 Zusammenfassung

Ontologien werden für die Wissensrepräsentation verwendet und können in grafischer Notation (Abschnitt 2.3) und sprachlicher Notation (Abschnitt 2.4) beschrieben werden. Sie repräsentieren die Konzepte einer Domäne und die Beziehungen zwischen diesen Konzepten. Die Verwendung einer formalen Semantik für Ontologien erlaubt die Anwendung von Inferenzverfahren, wodurch neues Wissen aus einer Ontologie abgeleitet werden kann. Eine der bekanntesten Implementierungssprachen für Ontologien ist OWL bzw. OWL 2.

3 Ontology Learning

3.1 Abgrenzungen

Wie bereits in Kapitel 1 erwähnt wurde, befasst sich das Gebiet des *Ontology Learning* damit, Methoden zu entwickeln, die den Prozess der Erzeugung von Ontologien unterstützen. Dabei geht es jedoch nicht um die Entwicklung von Vorgehensmodellen zur manuellen Ontologieerzeugung, sondern um die voll- oder semiautomatische Konstruktion von Ontologien, d.h. um das Ableiten von Ontologien aus Daten (Cimiano, 2014; Lehmann und Völker, 2014, S. V, S. IX). In (Lehmann und Völker, 2014, S. Xf) werden vier wesentliche Forschungsfelder des Ontology Learning identifiziert: In dem Forschungsfeld des *Linked Data Mining* werden Prozesse entwickelt, die die Identifizierung von Mustern in RDF-Graphen ermöglichen sollen. Der Unterschied zu natürlichsprachlichen Texten liegt darin, dass „Linked Data“ bereits strukturiert sind, das Problem bestünde jedoch darin, dass die Schemadefinitionen dieser Strukturierung oftmals nicht mit den Daten veröffentlicht werden. Die Namensbezeichnung liegt an dem Umstand, dass die Daten untereinander verlinkt sind. Als ein weiteres Forschungsfeld wird das *Konzeptlernen in Beschreibungslogiken* aufgeführt. Hierbei gehe es darum, Axiome zu erlernen, die bestimmte Schemata definieren, wie z.B. Definitionen von Klassen (siehe Kapitel 2.4). Das Gebiet des *Crowdsourcing* stellt außerdem ein weiteres Forschungsfeld dar, das auch für das Ontology Learning neue Erkenntnisse liefern kann. Der Term *Crowdsourcing* bezeichnet in diesem Zusammenhang die gemeinsame Erzeugung einer Ontologie durch viele Anwender (die „Crowd“). Die *Ontologieextraktion aus natürlichsprachlichen Texten*, womit sich diese Arbeit befasst, wendet Methoden aus dem Bereich des Textmining und des maschinellen Lernens (ML) an, um Ontologien aus Daten abzuleiten, die in Form natürlichsprachlicher Texte vorliegen.

3.2 Probleme

Wie der vorherige Abschnitt gezeigt hat, lässt sich das Lernen von Ontologien nicht auf einen einzigen Bereich beschränken. Trotz der verschiedenen Forschungsschwerpunkte können jedoch auch Gemeinsamkeiten in Bezug auf die zu bewältigenden Herausforderungen gefunden werden. Die wesentlichen Herausforderungen des Ontology Learning - und somit auch der Ontologieextraktion aus natürlichsprachlichen Texten - bestehen in den folgenden sechs Bereichen (Lehmann und Völker, 2014, S. XIf):

Heterogenität. Die Ressourcen, auf denen im Zuge des Lernens einer Ontologie gearbeitet wird, können sich stark in ihrer Beschaffenheit unterscheiden (Format, Qualität, Domäne, etc.). Diesem Bereich ist bisher jedoch zu wenig Aufmerksamkeit gewidmet worden.

Unsicherheit. Eine geringe Qualität der Daten kann zu einer geringeren Qualität der gelernten Ontologie führen. Als Qualität kann beispielsweise der Informationsgehalt, die Korrektheit der Informationen und die Rechtschreibung betrachtet werden. In Bezug auf eine anschließende Auswertung der erlernten Ontologie durch Domänenexperten ist es wichtig, die ermittelten Informationen in der Ontologie mit Annotationen z.B. in Bezug auf ihre Autorenschaft und dem Datum der Informationsermittlung zu versehen.

Schlussfolgerbarkeit. Wie bereits erwähnt wurde, können Ontologien verwendet werden, um Aussagen in Bezug auf das Wissen, das sie modellieren, zu überprüfen und abzuleiten. In diesem Zusammenhang ist es wichtig, dass gelernte Ontologien konsistent und kohärent sind.

Skalierbarkeit. In Bezug auf große Mengen von Daten, die potentiell für das Ontology Learning herangezogen werden können, spielt die Skalierbarkeit eine wichtige Rolle. Hierbei geht es u.a. darum, die Effizienz von Ontology Learning Algorithmen dadurch steigern zu können, dass bereits existierende Ontologien in den Lernprozess eingebunden werden, genauso wie die Einbeziehung mehrerer (auf unterschiedlichen Rechnern befindlichen) Quellen.

Qualität. Die Beurteilung der Qualität einer erlernten Ontologie ist keine einfache Aufgabe. Einer der Gründe ist der, dass selbst die manuelle Erzeugung einer Ontologie einer bestimmten Domäne durch verschiedene Domänenexperten in gewissen Punkten Unterschiede aufweist. Qualitätsmerkmale sind unter anderem die formale Korrektheit, die Vollständigkeit und die Konsistenz.

Interaktivität. Da der Mensch im Post-Processing der automatisch erzeugten Ontologie eingebunden ist, werden gute interaktive Methoden benötigt, um die anschließende Bearbeitung der erzeugten Ontologie zu erleichtern.

Nach (Cimiano, 2014, S. Vff) liegt die Schwierigkeit des Erlernens einer Ontologie im Wesen der Ontologie selbst sowie der zur Verfügung stehenden Daten. Während eine Ontologie einen Weg der Konzeptualisierung der Welt oder einer bestimmten Domäne darstelle, sei das Ergebnis eines Ontology Learning Algorithmus immer das Produkt der Eigenheiten des Datenbestands. Aus diesem Grund könne der Arbeitsaufwand, der mit der Überführung einer Ontologie, die aus einem semi- oder vollautomatischen Verfahren gewonnen wurde, in eine Ontologie, die von Domänenexperten akzeptiert wird, kostspieliger sein, als die vollständige manuelle Erzeugung. Ein weiteres Problem stelle der Mangel an Anwendungen dar, die es ermöglichen, Ontologien zu erlernen. Besondere Aufmerksamkeit widmet Cimiano (2014) der Diskrepanz zwischen dem Bereich des Ontology Learning und des Natural Language Processing (NLP). Während Methoden des NLP auf dem Gebiet des Ontology Learning in großem Maße angewendet werden, würden im Gegenzug (OWL) Ontologien auf dem Gebiet des NLP kaum angewendet werden.

Lehmann und Völker (2014) machen darauf aufmerksam, dass eine erzeugte Ontologie im Anschluss auch gewartet werden müsse, da das Wissen, das durch die Ontologie modelliert ist, Veränderungen ausgesetzt sein kann (Lehmann und Völker, 2014, S. X). Neben den bereits erwähnten Hürden des Ontology Learning, stelle auch der Umstand eine Schwierigkeit dar, dass Ontologien explizit definiert seien, doch das Wissen in textuellen Ressourcen oft vage und implizit ist (Lim u. a., 2013, S. 21). Dem Ontology Learning Algorithmus fehlt deshalb *Hintergrundwissen*, da nicht jede Information, die für das Verständnis eines gegebenen Textes notwendig ist, auch in diesem Text zu finden ist.

3.3 Metriken

In vielen Bereichen des Textmining werden bestimmte Metriken verwendet, um den Erfolg oder Misserfolg verwendeter Methoden anzugeben. Die wichtigsten Metriken in diesem Zusammenhang werden im Folgenden erläutert.

Im Kontext des *Information Retrieval* soll beurteilt werden, mit welchem Grad eine Menge ermittelter Dokumente relevant, bezogen auf eine aktuelle Suchanfrage, ist. Die *Precision* setzt deshalb die relevanten Dokumente (aus der Menge der ermittelten Dokumente) mit der Gesamtanzahl der ermittelten Dokumente in Beziehung. Der so berechnete Quotient gibt Auskunft darüber, wie viel Prozent der ermittelten Dokumente unter allen gefundenen Dokumenten relevant sind. Die Precision ist in Anlehnung an (Bergmann, 2011, S. 32) in Gleichung 3.1 dargestellt.

$$\text{Precision} = \frac{\text{recherchierte relevante Dokumente}}{\text{alle recherchierten Dokumente}} \tag{3.1}$$

Ein weiteres Bewertungsmaß ist der *Recall*, der angibt, wie viele relevante Dokumente von allen zu findenden relevanten Dokumenten gefunden wurden. In Anlehnung an (Bergmann, 2011, S. 32) ergibt sich der Recall wie in Gleichung 3.2 dargestellt.

$$\text{Recall} = \frac{\text{recherchierte relevante Dokumente}}{\text{alle relevanten Dokumente}} \tag{3.2}$$

Zwischen Precision und Recall besteht ein Zielkonflikt (Bodendorf, 2013, S. 97). Dieser äußert sich dadurch, dass ein verbesserter Wert der Precision einen schlechteren Wert des Recalls zur Folge haben kann und umgekehrt. Maßnahmen, die die Precision verbessern, zielen darauf ab, den Devisor zu minimieren. Maßnahmen, die den Devisor der Precision minimieren sollen, schließen evtl. auch relevante Dokumente aus. Wenn die Menge der ausgeschlossenen relevanten Dokumente kleiner als die Menge der ausgeschlossenen irrelevanten Dokumente ist, ist der Wert der Precision folglich besser als zuvor. Der Recall hätte sich jedoch verschlechtert. Umgekehrt könnte der Recall leicht verbessert werden, indem mehr Dokumente ermittelt werden, da sich die Wahrscheinlichkeit, auch relevante Dokumente darin zu finden, mit jedem zusätzlichen Dokument erhöht. Die Precision würde dieses Vorgehen jedoch negativ beeinflussen.

Um Recall und Precision zu ermitteln, muss die Anzahl relevanter Dokumente bekannt sein. Diese Leistung wird oftmals durch einen *Klassifizierer* erbracht. Bei einem Klassifizierer handelt es sich um Software, die Dokumenten eine *Klasse* bzw. ein *Label* zuweist. Die Menge zuweisbarer Klassen kann vom Entwickler festgelegt werden oder automatisch „erlernt“ werden. Beispiele für Klassen sind *relevant* und *irrelevant*, andere Klassen sind jedoch auch möglich.

Im Zuge einer solchen Klassifikationsaufgabe wird auch oft von *true positive, false positive, true negative* und *false negative* gesprochen. Diese vier Begrifflichkeiten drücken aus, ob einem Dokument korrekterweise eine Klasse zugeteilt wurde (true positive) oder ob dies fälschlicherweise geschah (false positive) und ob einem Dokument korrekterweise eine Klasse nicht zugeteilt wurde (true negative) oder ob dies geschah, obwohl es nicht hätte passieren dürfen (false negative). Wenn Precision und Recall im Rahmen einer Klassifikationsaufgabe vergeben werden, dann geschieht dies jeweils pro zu vergebender Klasse.

Die *Accuracy* (dt. Genauigkeit) gibt den Wert an, mit dem Dokumente korrekt klassifiziert wurden, und ist nach (Powers, 2015, S. 3) in Gleichung 3.3 angegeben. Hierbei steht TP für die Anzahl der Dokumente die *true positive* klassifiziert wurden, TN für die Anzahl der Dokumente, die *true negative* klassifiziert wurden und N für die Gesamtanzahl der Dokumente, die klassifiziert wurden.

$$\text{Accuracy} = \frac{[TP + TN]}{N} \tag{3.3}$$

Eine weitere Messung ist die *F-Messung* (engl. F-measure), die Precision und Recall in einer einzigen Messung zusammenfasst. Die F-Messung ist in den Gleichungen 3.4 und 3.5 definiert (Sasaki, 2007, S. 3). Dabei ist P die Precision, R der Recall und β ein Parameter, der je nach gewählter Größe den Recall oder die Precision in der jeweiligen Gewichtung begünstigt oder abschwächt.

$$\mathrm{F}_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R} \qquad (0 \leq \beta \leq +\infty) \tag{3.4}$$

Wenn β kleiner als 1 ist, begünstigt dies die Precision, wenn β größer als 1 ist, den Recall und wenn $\beta = 1$ gilt, entspricht dies einem Gleichgewicht von Precision

und Recall. Letzterer Fall wird deshalb auch oft als F_1-Messung bezeichnet und ist in Gleichung 3.5 angegeben.

$$\mathrm{F}_1 = \frac{2 \cdot PR}{P + R} \tag{3.5}$$

4 Ontologieextraktion

4.1 Methoden

Auf dem Gebiet der Ontologieextraktion hat sich bisher keine Vorgehensweise der Ontologieextraktion als Standardmodell beweisen können. Eine Vorgehensweise, die in der Literatur jedoch sehr oft angewendet wurde, ist der *Ontology learning layer cake* (OLC), welcher in Abbildung 4.1 dargestellt wird. Der OLC stellt den Prozess der Ontologieextraktion als Folge aufeinander aufbauender Schritte dar.

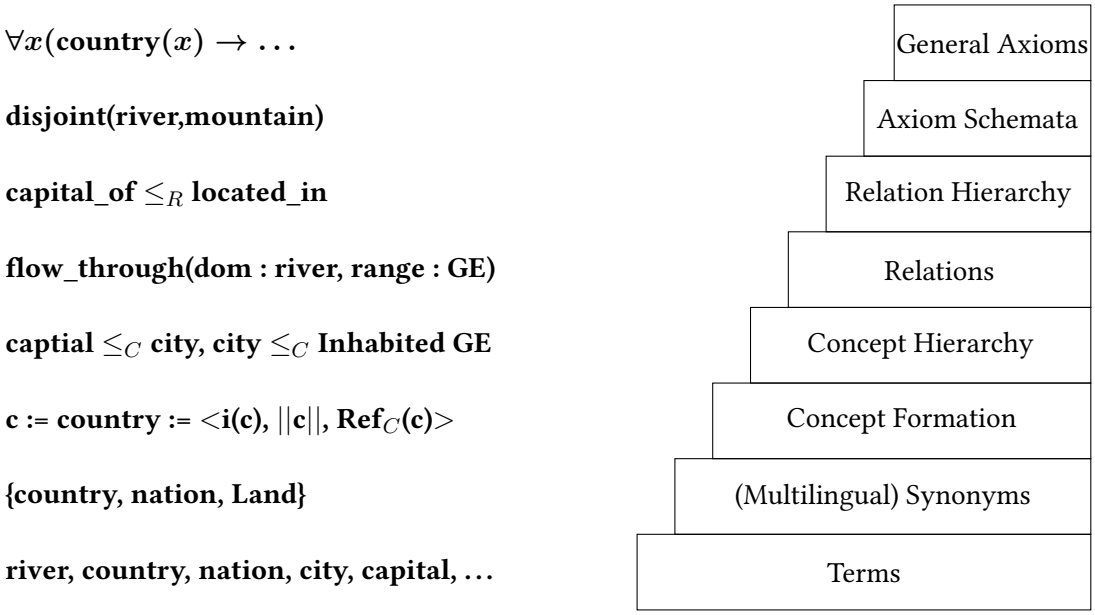

Abbildung 4.1: Ontology learning layer cake nach (Cimiano u. a., 2009, S. 251)

Es werden zunächst Terme und Synonyme von Termen im Text identifiziert. Darauf aufbauend werden Verfahren zur Identifikation von Konzepten (Concept Formation)

und Identifikation von taxonomischen Beziehungen zwischen Konzepten (Concept Hierarchy) angewendet. Aufbauend auf dieser Taxonomie werden dann (hierarchische) Relationen zwischen den Konzepten ermittelt. Wenn sowohl die Relationen als auch die Konzepte ermittelt wurden, werden Axiome ermittelt. Der OLC beschreibt eine Vorgehensweise der Ontologieextraktion. In keinem bekannten Verfahre wurden bisher alle Schritte umgesetzt. Vielmehr behandeln einzelne Arbeiten jeweils einen Schritt im OLC. Der OLC ist stark auf das Lernen taxonomischer Beziehungen ausgerichtet und lernt deshalb TBox Axiome (Cimiano u. a., 2009, S. 250). Da der OLC kein allgemein akzeptiertes Standardmodell der Ontologieextraktion ist, weichen die Verfahren, die in den Abschnitten 4.3, 4.4, 4.5 und 4.6 vorgestellt werden, vom OLC ab. In Frantzi u. a. (2000) werden beispielsweise Terme mit Konzepten gleichgesetzt und in einem Schritt, ohne die Beachtung von Synonymen, extrahiert. Die Extraktion von Instanzen wird im OLC nicht näher betrachtet.

Bevor Texte in diesen oder ähnlichen Schritten analysiert werden, müssen diese für die meisten Algorithmen in einer *Vorverarbeitungsphase* aufbereitet werden. Dieser obligatorische erste Schritt und alle weiteren werden in den folgenden Abschnitten näher betrachtet.

4.2 Vorverarbeitung

Im Bereich des Textmining ist es oft notwendig, dass der *Textkorpus* vor der eigentlichen Aufgabe, in diesem Fall der Aufgabe der Ontologieextraktion, zunächst grundlegende Verarbeitungsschritte durchläuft. Als *Korpus* wird eine Sammlung von Ressourcen bezeichnet, auf denen im Zuge des Verarbeitungsprozesses gearbeitet wird. Der Prozess der Vorverarbeitung beinhaltet Standard-Aufbereitungsmethoden, die die Dokumente im Textkorpus so aufbereiten, dass nachfolgende Algorithmen die Informationen, die in den Dokumenten enthalten sind, leichter verarbeiten bzw. leichter darauf zugreifen können. Dieser Prozess wird auch oft als *pre-processing* bezeichnet und beinhaltet die Methoden, die in den Abschnitten 4.2.1 bis 4.2.4 geschildert werden. Im Allgemeinen müssen jedoch nicht alle Methoden zur Lösung eines gegebenen Problems angewendet werden, wenn die daraus gewonnenen Informationen irrelevant für die Problemlösung sind.

4.2.1 Segmentierung

Vor der eigentlichen Ausführung der Vorverarbeitung müssen die Dokumente von ihrem Speicherort eingelesen werden. Im Zuge dieses Einlesens entspricht der Dokumenteninhalt einem Zeichenstrom. Dieser Zeichenstrom wird in einem ersten Verarbeitungsschritt in *Tokens* zerlegt und darauf aufbauend in einem weiteren Verarbeitungsschritt in *Sätze*.

Tokenisierung

Im Zuge der Tokenisierung wird ein eingehender Zeichenstrom in einzelne *Tokens* zerlegt. Was konkret als Token betrachtet wird, kann im Zuge der Implementierung eines *Tokenisers* festgelegt werden. Üblicherweise werden Zahlen, Symbole, Punktuationen („?“, „.“, etc.) und Wörter unabhängig von ihrer Rechtschreibung als Tokens aufgefasst (Maynard und Bontcheva, 2014, S. 52f). In der Regel werden Tokeniser jedoch selbst implementiert, da viele open source Tokeniser existieren.

Das Verfahren der Tokenisierung kann als Klassifikationsaufgabe betrachtet werden. Die Aufgabe besteht dann darin, aufeinanderfolgende Zeichen als Tokens zusammenzufassen, mit dem Ziel, korrekte Tokens zu erkennen. Ob eine solche Folge von Zeichen korrekt als Token zusammengefasst wurde, beeinflusst den Erfolg darauf aufbauender Verarbeitungsschritte. Das englische Wort *don't* würde je nach Tokeniser in drei Tokens („*do*“, „*'*“ und „t“) oder zwei Tokens („*do*“ und „*n't*“) zerlegt werden, wobei die Variante mit zwei Tokens für das korrekte Funktionieren des POS-Tagging notwendig ist (Maynard und Bontcheva, 2014, S. 53).

Satzteilung

In der Satzteilung werden Tokens zu Sätzen zusammengefasst. Je nach Sprache werden dabei die Zeichen von einem *Sentence Splitter* als Satzende interpretiert, die in der jeweiligen Sprache das Satzende markieren. Im Japanischen gibt es beispielsweise zusätzliche Zeichen, die das Ende eines Satzes markieren, wie z.B. Finalpartikel. Im Deutschen entsprechen diese dann den Punktuationszeichen. Diese relativ einfache Aufgabe kann jedoch dadurch erschwert werden, dass zwischen dem Satzanfang

und dem Satzende Zeilenumbrüche, Tabellen und andere Formatierungen auftreten (Maynard und Bontcheva, 2014, S. 53f).

4.2.2 Part-of-speech Tagging

Im Rahmen des *Part-of-speech-Tagging* (POS-Tagging) arbeitet ein POS-Tagger auf Tokens und weist jedem Token eine Kennzeichnung (Tag) zu. Ein solches Tag gibt Auskunft darüber, welcher Wortart (Nomen, Adjektiv, etc.) bzw. welcher Symbolart (Zahl, Punktuation) das Token entspricht. Jeder POS-Tagger vergibt Tags aus einer festgelegten Menge, die *Tagset* genannt wird und sich für unterschiedliche Sprachen unterscheidet. Für die englische Sprache ist das *Penn Treebank Tagset* weit verbreitet und wird unter anderem vom *Stanford Log-linear POS-Tagger* verwendet. Die Annotierung des Satzes

And of course it is not a domain-specific source

auf Basis des Penn Treebank Tagsets ist in Tabelle 4.1 zu sehen. POS-Tagging wird im Bereich der Dokumentenklassifikation und der natürlichen Sprachverarbeitung im Kontext des Textmining angewendet (Miner u. a., 2012a, S. 38).

Wort	**POS-Tag**	**Bedeutung**
And	CC	Koordinierende Konjunktion
of	IN	Präposition
course	NN	Nomen
it	PRP	Personalpronomen
is	VBZ	Verb, Gegenwart $3te$ Person Singular
not	RB	Adverb
a	DT	Artikel (engl. allgemein: Determiner)
domain-specific	JJ	Adjektiv
source	NN	Nomen

Tabelle 4.1: Annotierung eines Nebensatzes aus (Senellart und Blondel, 2007, S. 32)

4.2.3 Wortstammreduzierung und Lemmatisierung

In vielen Anwendungsfällen sind bestimmte Wortarten von besonderer Bedeutung, sodass die POS-Tags verwendet werden können, um nur bestimmte Wortarten aus

den Dokumenten herauszufiltern. In der Ontologieextraktion sollen unter anderem die Konzepte einer Domäne ermittelt werden. Es ist naheliegend, dass u.a. Nomen für Konzepte herangezogen werden können. Gleich welche Methode zum Auffinden der Nomen verwendet wird, die letzten Endes als Konzepte einer Domäne aufgefasst werden, treten Nomen in unterschiedlichen Ausprägungen auf. Das Wort *Wortstamm* hat im Zusammenhang mit der Domäne *Textmining* eine besondere Bedeutung, wie dieser Abschnitt darstellen soll. Es ist naheliegend, dass dieses Wort als Konzept betrachtet werden soll. Es kann u.a. jedoch auch in den Formen *Wortstämme*, *Wortstämmen* etc. auftreten, wobei immer dasselbe Konzept gemeint ist. Das Ziel der Konzeptextraktion sollte es sein, zu erkennen, dass hier von nur einem Konzept die Rede ist und nicht von drei Konzepten. Die *Wortstammreduzierung* (engl. Stemming) führt die Worte auf ihren Wortstamm zurück - in diesem Fall also auf das Wort *Wortstamm*. Im Zuge der Wortstammreduzierung werden allerdings nicht immer richtige Wörter erzeugt, was je nach Textmining-Architektur berücksichtigt werden muss. Beispielsweise erhält die Textmining-Architektur GATE (General Architecture for Text Engineering) jedes Dokument im Zuge der Verarbeitungsschritte in seiner Originalform, sodass Annotationen nicht direkt im Dokument vorgenommen werden, sondern in einer, für jeden Verarbeitungsschritt eigenen, Schicht. Dies bedeutet, dass weitere Verarbeitunggschritte immer noch mit den ursprünglichen Wörtern arbeiten können. Das Stemming des Satzes aus Abschnitt 4.2.2 erzeugt die folgenden Wortstämme:

and of cours it is not a domain-specif sourc

Zu erkennen ist, dass die Wortstämme keine korrekten Wörter darstellen. Der *Snowball Stemmer* ist einer der bekanntesten Stemmer (Miner u. a., 2012b, S. 48) und ist in unterschiedlichen Varianten für 14 Sprachen verfügbar, darunter Deutsch, Englisch, Französisch und Russisch.

Eine andere Möglichkeit der Vereinheitlichung unterschiedlicher Worte ist die *Lemmatisierung*. Die vereinheitlichte Form wird als *Lemma* bezeichnet und kann sich vom Wortstamm unterscheiden. Im slowenischen haben die Wörter *pisati, pišem, pišeš* und *pišemo* (schreiben) das gemeinsame Lemma *pisati* aber den Wortstamm *pi* (Juršič u. a., 2007, S. 206). Außerdem bezieht die Lemmatisierung die POS-Tags und andere Wörter in der Umgebung des gerade betrachteten Wortes in das Auffinden des Lemmas

ein: Das Wort *meeting* in der Gebrauchsform des Verbs würde in der Lemmatisierung das Lemma *meet* ergeben. Würde es hingegen als Nomen gebraucht werden, so wäre das Lemma *meeting* (Miner u. a., 2012b, S. 48). Die Lemmatisierung kann im Kontext des Textmining dem Bereich der natürlichen Sprachverarbeitung zugeordnet werden (Miner u. a., 2012a, S. 38), ebenso wie das Stemming.

4.2.4 Named Entity Recognition

Die *Named Entity Recognition* (NER) ermittelt Eigennamen in Texten und weist diesen Kategorien zu (z.B. Person, Organisation, Ort, etc.) (Maynard und Bontcheva, 2014, S. 55). Es können drei Ansätze unterschieden werden: Der regelbasierte Ansatz verwendet Pattern (dt.: Muster) für die NER, der statistische Ansatz basiert auf dem Trainieren eines Klassifizierers (engl.: Classifier) (z.B. Hidden Markov Modelle, Support Vector Machines, etc.) bzw. auf statistischen Methoden und der dritte Ansatz kombiniert die ersten beiden Ansätze (Maynard und Bontcheva, 2014, S. 55f). NER wird in den Bereichen der Dokumentenklassifikation und der Informationsextraktion angewendet (Miner u. a., 2012a, S. 38).

4.2.5 Zusammenfassung

Die Vorverarbeitung von Dokumenten ist für die meisten Anwendungen, die im Bereich des Textmining eingesetzt werden, notwendig. Oft werden die Ergebnisse, die in der Vorverarbeitungsphase erzeugt werden, von anderen Algorithmen benötigt. Fehler, die im Zuge der Vorverarbeitung gemacht werden, wie z.B. falsch segmentierte Sätze, können zu Qualitätseinbußen der später eingesetzten Algorithmen führen. Für die hier vorgestellten Vorverarbeitungsschritte gibt es allerdings Anwendungen/Bibliotheken, die sehr gute Ergebnisse erzielen. Darüber hinaus muss darauf geachtet werden, dass die eingesetzten Komponenten der jeweiligen Vorverarbeitungsschritte aufeinander abgestimmt sind. Dies bedeutet z.B., dass ein NER-Tagger mit den Tags arbeiten muss, die von einem zuvor eingesetzten POS-Tagger für die Tokens vergeben wurden.

4.3 Konzeptextraktion

Wie bereits in Kapitel 2 geschildert wurde, bilden Konzepte und Relationen zwischen diesen Konzepten die Grundlage für Ontologien. In diesem Abschnitt geht es darum, die relevanten Konzepte einer Domäne zu erkennen und zu extrahieren.

Jede nicht zu allgemeine Domäne hat bestimmte Begriffe, die charakteristisch für diese Domäne sind. In der Domäne Fussball wären dies u.a. die Begriffe *Auswechselspieler* und *Mannschaftsaufstellung*, in der Domäne Biologie könnten dies u.a. die Begriffe *Enzyme* und *Proteine* sein. Diese Begriffe entsprechen den Fachbegriffen, die in dieser Domäne verwendet werden. Die Menge der Fachbegriffe einer Domäne wird als *Terminologie*, und ein Element aus dieser Menge - d.h. ein Fachbegriff - als *Term* bezeichnet. Die linguistische Repräsentation eines Konzeptes ist ein Term (Frantzi u. a., 2000, S. 115). In der Konzeptextraktion sollen diese Terme identifiziert werden, sodass sie als Konzepte der Domäne verwendet werden können bzw. als deren Instanzen.

Nicht alle Terme bestehen aus nur einem einzigen Wort. *Multi-Wort-Terme* (engl.: Multi-word terms) bezeichnen solche Fachbegriffe, die aus mehreren Worten bestehen. Dies ist beispielsweise bei dem Multi-Wort-Term *Support Vector Machine* der Fall, welcher in der Domäne des maschinellen Lernens auftritt und im Rahmen der Klassifikation eingesetzt wird (siehe Abschnitt 4.2.4). Algorithmen aus dem Bereich der Termextraktion konzentrieren sich häufig auf eine oder mehrere der nachfolgend genannten Charakteristiken von Termen:

Kollokation. Der Begriff Kollokation betrifft in der Linguistik die lexikalische Ebene der sprachwissenschaftlichen Analyse und umfasst die syntagmatischen Verbindungen zwischen Wörtern (Bahns, 1996, S. 6).

Unithood. Die Unithood ist ein Maß, das die Stärke/Stabilität syntagmatischer Kombinationen oder Kollokationen angibt (Kageura und Umino, 1996, S. 272). Per Definition ist die Unithood für komplexe Terme, komplexe grammatikalische Kollokationen und idiomatische Ausdrücke relevant. Idiomatische Ausdrücke sind Ausdrücke, deren Bedeutung sich nicht allein aus ihnen selbst ableiten lässt, wie beispielsweise viele Redensarten (z.B. „Es ist noch kein Meister vom Himmel gefallen!“).

Termhood. Die Termhood ist ein Maß, das kennzeichnet, inwieweit eine lexikalische Einheit (d.h. ein N-Gramm) mit einem domänenspezifischen Konzept in Beziehung steht (Kageura und Umino, 1996, S. 272). Ein N-Gramm ist ein Wort das aus N Wörtern besteht. Die Termhood ist per Definition sowohl für komplexe linguistische Einheiten ($N \geq 2$) als auch für einfache linguistische Einheiten ($N = 1$) relevant.

Noise. Jeder Algorithmus, der Konzepte, Instanzen und Relationen aus Texten extrahiert, kann falsche Ergebnisse produzieren. Die Menge der falschen Ergebnisse wird als *Noise* bezeichnet.

Eine (mögliche) Vorgehensweise in der Termextraktion ist in Abbildung 4.2 dargestellt. Diese zeigt, dass eine Menge von Dokumenten zunächst die bereits geschilderten Vorverarbeitungsschritte duchläuft. Die Algorithmen zur Termextraktion arbeiten dann auf dem annotierten bzw. gefilterten Textkorpus und extrahieren und bewerten Terme. Der Prozess der Termextraktion erzeugt Ergebnisse, die von einem Domänenexperten überprüft werden sollten, um die Qualität der extrahierten Terme sicherzustellen. Eine Sortierung der Terme nach Termhood/Unithood kann bei einer sehr großen Menge extrahierter Terme dazu beitragen, dass der Domänenexperte mit der Überprüfung jener Terme beginnt, die entsprechend des verwendeten Maßes am wahrscheinlichsten Konzepte darstellen.

4.3.1 Die NC-Value Methode

Die *NC-Value* Methode von Frantzi u. a. (2000) wurde für die Extraktion von Multi-Wort-Termen aus englischen Textkorpora entwickelt. Die Methode arbeitet dabei in drei Phasen: Die *C-Value Methode* extrahiert Multi-Wort-Terme und weist diesen einen C-Wert zu. Multi-Wort-Terme, die einen vorher definierten Schwellwert bezüglich ihres C-Wertes unterschreiten, werden verworfen. Anschließend werden die Kontextinformation der so ermittelten Multi-Wort-Terme ermittelt, um danach diese Kontextinformationen dafür zu verwenden, die Qualität der Multi-Wort-Terme zu erhöhen.

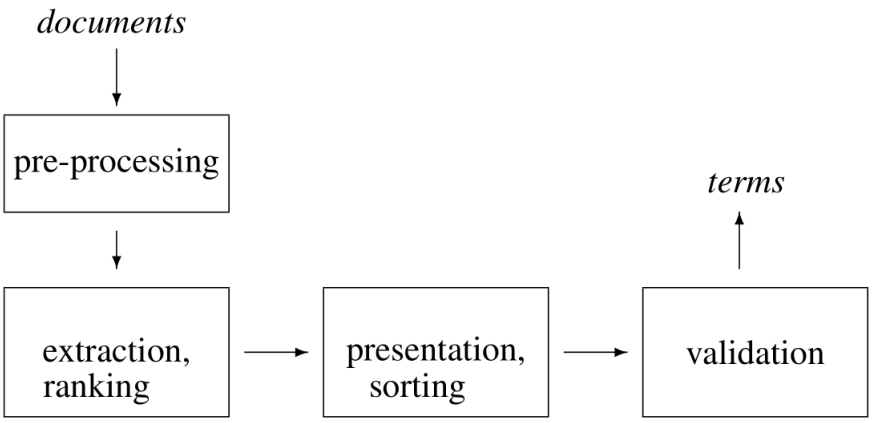

Abbildung 4.2: Die vier Verarbeitungsschritte des Termextraktionsprozesses (Ahrenberg, 2009, S. 3)

Der C-Wert

Vor der Anwendung der C-Value Methode müssen auf dem Textkorpus die *Segmentierung* und das *Part-of-speech Tagging* angewendet werden. Die Anwendung der *Wortstammreduzierung* bzw. der *Lemmatisierung* erhöht die Genaugikeit dieser Methode. Ein zu definierender *linguistischer Filter* sorgt dafür, dass nur solche Wörter betrachtet werden, deren Wortart gemäß ihres POS-Tags als Term der Domäne in Betracht kommen. Die besten Ergebnisse werden erzielt, wenn nur Nomen und Adjektive als mögliche Kandidatenterme betrachtet werden, da die meisten Terme diese Wortarten enthalten (Frantzi u. a., 2000, S. 116ff). Eine Liste mit Stopp-Wörtern sorgt dafür, dass von den verbliebenen Wörtern nur solche betrachtet werden, die nicht in der Stopp-Liste sind. Das Ergebnis ist eine Liste aus *Kandidatentermen*, das heißt N-Grammen, die als Terme in Betracht kommen. Da potentiell sehr viele Kandidatenterme ermittelt werden, spielt der Zeitaufwand der manuellen Auswertung eine wichtige Rolle. Aus diesem Grund werden die Kandidatenterme in dieser Liste absteigend nach ihrer *Termhood* sortiert. Der Begriff Termhood wird in Frantzi u. a. (2000) synonym mit dem Begriff C-Value verwendet. Wie später jedoch noch erklärt werden wird, entspricht der C-Wert allerdings eher der Unithood. Ein Domänenexperte kann die so erzeugte Liste dann beginnend mit den Kandidatentermen, die am wahrscheinlichsten echte

Terme der Domäne sind, überprüfen und aufhören, wenn die Termhood zu niedrig wird. Zur Berechnung des C-Value werden die folgenden 4 Messungen vorgenommen:

1. Es wird für jeden Kandidatenterm im Textkorpus dessen Häufigkeit ermittelt.
2. Es wird die Häufigkeit ermittelt, mit der der Kandidatenterm als Teil anderer, längerer Kandidatenterme auftritt.
3. Es wird für jeden Kandidatenterm aus Messung 2 die Anzahl der längeren Kandidatenterme aus Messung 2 ermittelt. Jeder längere Kandidatenterm wird nur einmal gezählt.
4. Es wird für jeden Kandidatenterm dessen Länge in Form von Wörtern ermittelt.

Messung 1 entspricht der Messung, die bis zum Jahr 1999 als Standard der Termextraktion galt und bereits damals gute Ergebnisse erzielte (Frantzi u. a., 2000, S. 117). Ein Vergleich der einfachen Häufigkeiten der Multi-Wort-Terme zeigt, dass jene Multi-Wort-Terme, die auch von Domänenexperten als solche erkannt werden, häufiger auftreten, als jene, auf die das nicht zutrifft. Aus dieser Beobachtung wird die Schlussfolgerung gezogen, dass geringe Häufigkeiten von echten Multi-Wort-Termen statistischen Ansätzen Probleme bereiten. Deshalb werden die Messungen 2 und 3 verwendet, um nur solche Multi-Wort-Terme zu berücksichtigen, die auch ausreichend oft außerhalb längerer Multi-Wort-Terme, d.h. eigenständig, in der Domäne auftreten. Das folgende Beispiel macht die Problematik deutlich und ist (Frantzi u. a., 2000, S. 117) entnommen: Der Multi-Wort-Term „soft contact lens“ ist ein Term der Domäne „Augenheilkunde“. Würde nur die Messung 1 verwendet werden, so würden auch die Multi-Wort-Terme „soft contact“ und „contact lens“ als Kandidatenterme ermittelt werden, wobei ersterer kein echter Multi-Wort-Term der Domäne ist. Wenn „soft contact“ nicht als eigenständiger Multi-Wort-Term im Textkorpus auftritt (d.h. nicht außerhalb längerer Multi-Wort-Terme), ist es wahrscheinlich kein echter Term.

Die Messung 4 wird verwendet, um die Termhood derjenigen Kandidatenterme zu erhöhen, die häufiger als andere in anderen Kandidatentermen enthalten sind. Die Annahme, die damit vertreten wird, ist die, dass solche Sub-Kandidatenterme (wahrscheinlich) auch außerhalb ihrer einbettenden Kandidatenterme als eben solche

von Domänenexperten wahrgenommen werden würden, selbst dann, wenn sie nicht außerhalb ihrer einbettenden Kandidatenterme im Textkorpus auftreten. Die vierte Messung stellt in diesem Fall also ein Gegengewicht zu den Messungen 2 und 3 dar und ist ein Sonderfall.

Zur Berechnung des C-Value bzw. der Termhood werden noch zwei Fälle unterschieden. Manche Kandidatenterme treten nie als Teil anderer Kandidatenterme auf, andere tun dies jedoch. Der C-Value berechnet sich dann nach Gleichung 4.1

$$\text{C-value(a)} = \begin{cases} log_2(|a|) \cdot f(a) & \text{Fall 1} \\ log_2(|a|) \cdot (f(a) - \frac{1}{P(T_a)} \cdot \sum_{b \in T_a} f(b)) & \text{Fall 2} \end{cases} \tag{4.1}$$

In Gleichung 4.1 ist a ein Kandidatenterm, $f(a)$ dessen Häufigkeit im Textkorpus (Messung 1) und $|a|$ ist die Länge von a gemessen in Wörtern (Messung 4). Der Logarithmus wird verwendet, damit der Längenunterschied zwischen sehr langen und kürzeren Kandidatentermen keine zu große Auswirkung auf die Termhood hat, da die Länge eines Kandidatenterms nicht entscheidend für dessen Relevanz als Term der Domäne sein sollte. T_a ist die Menge der extrahierten Kandidatenterme, die a enthalten und $P(T_a)$ entspricht der Anzahl dieser Kandidatenterme. Der durch die Messungen 2 und 3 geschilderte Effekt wird durch die Subtraktion $f(a) - \frac{1}{P(T_a)} \cdot \sum_{b \in T_a} f(b)$ ausgedrückt.

Jody Foo macht in ihrer Dissertation darauf aufmerksam, dass der C-Wert im Gegensatz zur Aussage von Frantzi u. a. (2000) *nicht* die Termhood angebe, sondern die Unithood (Foo, 2012, S. 33). Die Autorin bezieht sich dabei auf die Definitionen der Termhood und der Unithood, wie sie von Kageura und Umino (1996) vorgenommen wurden. Der C-Wert gibt nicht an, inwieweit die Kandidatenterme mit Konzepten der Domäne in Beziehung stehen (Termhood), da die Konzepte der Domäne noch nicht identifiziert wurden. Er gibt stattdessen an, wie stark die syntagmatischen Kombinationen (d.h. die N-Gramme) in Bezug auf den Textkorpus sind - und entspricht daher eher der Unithood. Foo weist allerdings darauf hin, dass die Summe im NC-Wert einen Bezug zur Termhood herstellt. Da die Terme bereits als Konzepte der Domäne identifiziert wurden, die von einem Domänenexperten validiert werden müssen, bewertet die

Summe im NC-Wert, inwieweit die Kontextwörter mit diesen Termen in Beziehung stehen.

Der Kontextgewichtungsfaktor

Die Unithood eines Multi-Wort-Terms bildet das Maß, nach dem beurteilt wird, ob ein Multi-Wort-Term ein Term einer Domäne ist oder nicht. Nach (Frantzi u. a., 2000, S. 123) können die Wörter, die in Texten zusammen mit den Kandidatentermen auftreten, herangezogen werden, um die Unithood zu verfeinern. In Frantzi u. a. (2000) wird hier statt von der Unithood, von der Termhood gesprochen. Wie bereits im vorherigen Abschnitt erwähnt wurde, ist dies nicht ganz korrekt (vgl. (Foo, 2012, S. 33)).

Nachdem die Kandidatenterme ermittelt wurden, wird für die n Kandidatenterme, deren C-Wert am größten ist, eine Menge von Wörtern ermittelt, die zusammen mit diesen Kandidatentermen auftreten. Diese Wörter werden als *Kontextwörter* bezeichnet. Für diese Kontextwörter wird eine Gewichtung durch Gleichung 4.2 berechnet.

$$\text{weight(a)} = \frac{t(w)}{n} \tag{4.2}$$

In Gleichung 4.2 ist a ein Kontextwort, $t(w)$ die Kardinalität der Menge der betrachteten Kandidatenterme, mit denen a im Textkorpus auftritt, und n die Kardinalität der Menge der betrachteten Kandidatenterme. Der so ermittelte *Kontextgewichtungsfaktor* der Kontextwörter wird in der Berechnung des NC-Wertes verwendet. Die Annahme der Kontextwörter ist, dass die Kontextwörter der *wahrscheinlichsten* Kandidatenterme ein starker Hinweis auf die Güte anderer Kandidatenterme sind. Die Ermittlung der Kontextwörter entspricht in diesem Sinne der Ermittlung der *Kollokation* (siehe Abschnitt4.3).

Der NC-Wert

Die *NC-Value Methode* verwendet den *C-Wert* und den *Kontextgewichtungsfaktor*, um den *NC-Wert* eines Kandidatenterms zu ermitteln. In Gleichung 4.3 ist a ein Kandidatenterm, C_a die Menge der Kontextwörter von a, b ein Kontextwort aus dieser Menge, $f_a(b)$ die Häufigkeit, mit der das Kontextwort b mit a zusammen im Textkorpus auftritt und *weight(b)* der Kontextgewichtungsfaktor von b. Die konstanten Faktoren 0.8 und

0.2 wurden von Frantzi u. a. (2000) im Zuge ihrer Experimente festgelegt und sollen die beste *Precision* erzeugen.

$$\text{NC-value(a)} = 0.8 \cdot \text{C-value(a)} + 0.2 \sum_{b \in C_a} f_a(b) \text{weight(b)} \tag{4.3}$$

4.3.2 Methoden basierend auf der tf-idf Methode

Die *term frequency inverse document frequency* (tf-idf) ist ein Maß dafür, wie relevant ein Dokument bezogen auf eine Suchanfrage ist und wird in diesem Zusammenhang im Information Retrieval verwendet (Manning und Schütze, 2000, S. 539). Die direkte Anwendung dieser Methode zum Zwecke der Termextraktion ist jedoch ungeeignet (Foo, 2012, S. 53), da sie Kandidatenterme, die in vielen Dokumenten auftreten, schlechter bewertet als solche, die in wenigen auftreten. Wie in Abschnitt 4.3.1 jedoch bereits dargestellt wurde, ist die Häufigkeit, mit der ein Wort oder eine Folge von Worten in einem Textkorpus auftritt, einer von mehreren wichtigen Indikatoren, die anzeigen, ob eine solche lexikalische Einheit ein Term ist oder nicht. Verschiedene Autoren nahmen deshalb gewisse Anpassungen an dieser Methode vor, um sie doch auf die Termextraktion anwenden zu können. Die nachfolgenden Abschnitte stellen deshalb die tf-idf Methode im allgemeinen vor und zeigen eine Variation der tf-idf Methode, die sich als geeignet für die Termextraktion herausgestellt hat.

Die tf-idf Methode des Information Retrieval

Um relevante Dokumente, bezogen auf eine Suchanfrage, zu finden, werden die relevanten Terme einer Suchanfrage ermittelt. Dies kann beispielsweise durch einen linguistischen Filter (siehe Abschnitt 4.3.1) bewerkstelligt werden. Wenn m relevante Terme identifiziert wurden und es n zu beurteilende Dokumente gibt, werden $n + 1$ Vektoren der Dimension m erzeugt. Dabei gibt es n Vektoren für die zu beurteilenden Dokumente und einen Vektor für die gestellte Suchanfrage. Jeder Vektor wird durch die Angabe von m Koordinaten in einem m-dimensionalen Vektorraum angegeben. Die Koordinaten der Vektoren werden durch die tf-idf Methode berechnet.

Ein Beispiel ist in Abbildung 4.3 zu sehen. Die Anfrage q beinhaltet die Wörter „Ontologie“ und „Definition“, die von einem linguistischen Filter, der nur die Nomen

aus der Anfrage extrahiert, ermittelt wurden. Der Textkorpus besteht aus den 3 Dokumenten $d1, d2$ und $d3$, sodass in dem 2-dimensionalen Vektorraum 4 Vektoren sind. Ein einzelner Vektor hat die Form (x, y) wobei x der ermittelte Wert des Suchbegriffs „Definition“ und y der Wert des Suchbegriffs „Ontologie“ ist. Der Vektor $d3$ ist am dichtesten an dem Vektor der Anfrage q, sodass das Dokument $d3$ am relevantesten in Bezug auf die Anfrage zu sein scheint.

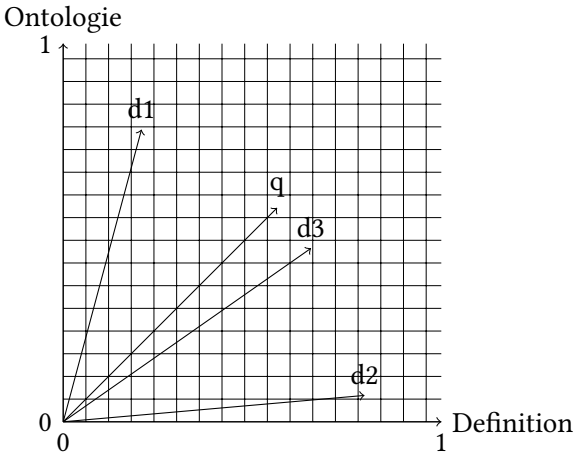

Abbildung 4.3: Anordnung von Vektoren in einem 2-dimensionalen Vektorraum in Anlehnung an (Manning und Schütze, 2000, S. 540)

Eine Variante der *tf-idf* ist in Gleichung 4.4 dargestellt (Manning und Schütze, 2000, S. 543). Für jedes Wort der Suchanfrage wird diese Gleichung angewendet.

$$\text{weight(i,j)} = \begin{cases} (1 + \log(\text{tf}_{i,j})) \cdot \log(\frac{N}{df_i}) & \text{wenn } \text{tf}_{i,j} \geq 1 \\ 0 & \text{wenn } \text{tf}_{i,j} = 0 \end{cases} \tag{4.4}$$

Hierbei ist $tf_{i,j}$ die Häufigkeit des Terms w_i im Dokument d_j (term frequency), df_i die Anzahl der Dokumente, die den Term w_i enthalten (document frequency), N die Gesamtanzahl der Dokumente, die betrachtet werden und $weight(i, j)$ der errechnete tf-idf Wert des Terms w_i im Dokument d_j (d.h. die Koordinate im Vektor d_j).

Der Ausdruck $(1 + (\log(tf_{i,j}))$ ist 1 wenn der Term w_i nur einmal im Dokument d_j beobachtet wird, sonst entsprechend größer. Der Ausdruck $\log(\frac{N}{df_i})$ entspricht der inversen Dokumentenhäufigkeit, d.h. der *idf*. Die Autoren machen darauf aufmerksam,

dass die Termhäufigkeit vollständig berücksichtigt wird, wenn der betrachtete Term in nur einem Dokument beobachtet wird und überhaupt nicht berücksichtigt wird, wenn der Term in allen Dokumenten beobachtet wird. Dies liegt an den Umformungsregeln, die auf logarithmische Ausdrücke angewendet werden können. Diese erlauben die Umformung des Bruchs in eine Subtraktion, die je nach geschildertem Fall zu einer Multiplikation mit $log(N)$ oder einer Multiplikation mit 0 führt. Dies bedeutet, dass ein Term, der in zu vielen Dokumenten auftritt, als nicht relevant erachtet wird und die tf-idf des Terms (d.h. $weight(i, j)$) gleich 0 ist.

Um zu ermitteln, wie ähnlich sich Vektoren in einem Vektorraum sind, kann die Gleichung 4.5 (Manning und Schütze, 2000, S. 299/541) verwendet werden, nachdem die Termgewichtungen mit Gleichung 4.4 berechnet wurden.

$$\cos(\vec{q}, \vec{d}) = \frac{\vec{q} \cdot \vec{d}}{|\vec{q}||\vec{d}|} = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} q_i^2} \sqrt{\sum_{i=1}^{n} d_i^2}} \tag{4.5}$$

In Gleichung 4.5 ist $\vec{q}$ der Vektor der Anfrage, $\vec{d}$ der Vektor eines Dokuments und i der Term i. Im Zähler steht dabei das Skalarprodukt der beiden Vektoren und im Nenner ein Ausdruck, der einer Längennormalisierung der Vektoren entspricht, d.h. die Längen beider Vektoren werden multipliziert und der Cosinus ist das Ergebnis dieser Division. Dabei entspricht ein Cosinus von 1.0 (d.h. $\cos(0°)$), dass die Vektoren in dieselbe Richtung zeigen und gleich sind, ein Kosinus von -1.0 (d.h. $\cos(180°)$) bedeutet hingegen, dass die Vektoren in die entgegengesetzte Richtung zeigen (Manning und Schütze, 2000, S. 541). Durch diese Gleichung kann das Dokument als am relevantesten identifiziert werden, dessen Cosinuswert am dichtesten bei 1.0 ist.

Contrastive Weight

Die *contrastive weight* stellt eine Anpassung der tf-idf zum Zwecke der Termextraktion dar. In Gleichung 4.4 können zwei Komponenten unterschieden werden: Die Komponente der Termhäufigkeit ($1 + log(tf_{i,j})$) und die Komponente der Dokumentenhäufigkeit ($log(\frac{N}{df_i})$). Angepasste Variationen der tf-idf Methode passen deshalb eine oder beide Komponenten der Formel 4.4 an.

Vor diesem Hintergrund muss erwähnt werden, dass es bei diesen Methoden im Vergleich zur NC-Value Methode von Frantzi u. a. (2000) einen grundlegenden Unterschied

bezüglich der Dokumente gibt, auf denen gearbeitet wird. Für domänenspezifische Ontologien gilt, dass die Konzepte, die sie enthalten, domänenspezifisch sind. Deshalb dürften diese Konzepte/Terme in Textkorpora die domänenunspezifisch sind, wenn überhaupt nur in sehr geringem Maße auftreten. Auf Grundlage dieser Annahme kann ein weiterer Textkorpus verwendet werden, der domänenunspezifisch ist. Die Unithood eines Terms wird dann besser, je öfter dieser Term im domänenspezifischen Textkorpus auftritt bei gleichzeitig seltenem Auftreten im domänenunspezifischen Textkorpus. Ein solcher Textkorpus wird in diesem Zusammenhang *contrastive corpus* und die Verwendung solcher Daten allgemein *contrastive data* genannt (Foo, 2012, S. 34).

Die Berechnungsvorschrift der *contrastive weight* ergibt sich nach (Basili u. a., 2001, S. 7) wie folgt: Ein linguistischer Filter ermittelt eine vorläufige Liste von Termkandidaten aus der Zieldomäne, d.h. aus dem domänenspezifischen Textkorpus. Die *contrastive weight* für *1-Gramme* wird durch Gleichung 4.6 berechnet.

$$w_t^i = log(f_t^i) \cdot log(\frac{N}{F_t}) \tag{4.6}$$

Dabei ist f_t^i die Häufigkeit des Kandidatenterms t in der Zieldomäne i. Der Ausdruck $log(\frac{N}{F_t})$ entspricht der inversen Worthäufigkeit (engl. inverse word frequency, IWF). F_t ist die Häufigkeit des Terms t in allen Textkorpora, die betrachtet werden und N die aufsummierten Häufigkeiten aller Kandidatenterme in allen betrachteten Textkorpora. Hier fällt die Ähnlichkeit zur tf-idf Methode aus Gleichung 4.4 auf. Die tf-idf Methode wurde durch ein anderes methodisches Vorgehen (ein kontrast-Korpus) und die Ersetzung der Dokumentenhäufigkeitskomponente durch die Worthäufigkeitskomponente angepasst.

Da die Häufigkeit eines Terms in allen Textkorpora nie größer als die Summe über alle Termhäufigkeiten in allen Textkorpora sein kann, gilt immer $F_t \leq N$. Sobald $F_t = N$ gilt, wird der Term mit 0.0 bewertet. Dies bedeutet, er ist Teil des allgemeinen Sprachgebrauchs und nicht domänenspezifisch. Je größer die Distanz p, mit $p = N - F_t$, desto besser wird ein Term bewertet. Das bedeutet aber auch, dass sich die Termhäufigkeit $log(f_t^i)$ weniger stark auf die Bewertung eines Terms auswirkt. Dies kann beispielsweise zur Folge haben, dass ein Term, der 70 mal in der Zieldomäne

beobachtet wurde und 72 mal ingesamt, schlechter bewertet wird, als ein Term der 10 mal in der Zieldomäne beobachtet wurde und 11 mal insgesamt: $log(70) \cdot log(\frac{83}{72}) = 0.11$ und $log(10) \cdot log(\frac{83}{11}) = 0.88$. Dies liegt an der inversen Worthäufigkeit (IWF). Diese bewertet Terme, die selten in der Zieldomäne vorkommen, gegenüber Termen, die häufig in der Zieldomäne auftreten, besser. Dieser Effekt der IWF tritt aber nur dann auf, wenn die Verteilung dieser Terme in allen Textkorpora merklich geringer ist, als die Summe der Häufigkeiten aller Terme in allen Textkorpora. Dies bedeutet allerdings auch, dass Kandidatenterme, die häufig in anderen Textkorpora gezählt wurden aber keine echten Terme sind (also schlecht bewertet werden sollen), einen positiven Effekt auf seltene echte Terme haben und einen negativen Effekt auf häufige echte Terme.

Für Multi-Wort-Terme, d.h. $N_{\geq 2}$-Gramme, wird die Gleichung 4.7 verwendet.

$$cw_{ct}^{i} = log(w_{h(ct)}^{i}) \cdot f_{ct}^{i} \tag{4.7}$$

Dabei ist $w_{h(ct)}^{i}$ die auf den Kopf des N-Gramms angewandte Gleichung 4.6. Als Kopf eines N-Gramms wird in diesem Zusammenhang von den am weitesten links stehenden Einheiten des N-Gramms gesprochen, wenn sie Nomen oder ein Adjektiv, gefolgt von einem Nomen sind (Basili u. a., 2001, S. 5). ct ist ein *complex term* (= Multi-Wort-Term, $N_{\geq 2}$-Gramm), f_{ct}^{i} ist die Häufigkeit von ct in der Domäne i. Die Liste der Kandidatenterme wird nach Anwendung der Gleichungen sortiert, damit sie schneller von einem Domänenexperten ausgewertet werden kann.

4.3.3 Zusammenfassung

Die Konzeptextraktion ist, nach der Vorverarbeitung, häufig der erste Schritt der Ontologieextraktion. Die linguistische Repräsentation eines Konzeptes ist nach Frantzi u. a. (2000) ein *Term*. Ein Term kann aus einem Wort oder aus mehreren Wörtern bestehen. Um Terme unter allen anderen Wörtern eines Textkorpus identifizieren zu können, werden je nach Methode unterschiedliche Heuristiken eingesetzt. Die NC-Value-Methode wurde konzipiert, um auf domänenspezifischen Textkorpora eingesetzt zu werden. Sie identifiziert Multi-Wort-Terme in einem solchen Textkorpus. Die tf-idf-Methode muss vor der Anwendung im Bereich der Term- bzw. Konzeptextraktion angepasst werden. Die contrastive-weight-Methode stellt eine solche Anpassung dar.

Im Gegensatz zur NC-Value-Methode identifiziert diese Methode Terme auf Grundlage eines domänenspezifischen und eines domänenunspezifischen Textkorpus.

4.4 Relationsextraktion

Das Ziel der Extraktion von Relationen ist die Verknüpfung von Konzepten. Für diese Aufgabe gibt es verschiedene Methoden, jede birgt gegenüber einer anderen gewisse Vorzüge, aber auch Nachteile. Einige der bekanntesten Methoden werden in den Abschnitten 4.4.1 bis 4.4.4 vorgestellt.

4.4.1 Lexico-Syntaktische Pattern

Lexico-syntaktische Pattern entsprechen regulären Ausdrücken, die auf POS-Tags bzw. Phrasen eines Satzes definiert werden. Sehr bekannt sind *Hearst Pattern*, die für das Auffinden von Hyponymen und Hypernymen (siehe Abschnitt 2.2.2) definiert wurden. Diese sind durch die folgenden 5 Regeln definiert (Maynard und Bontcheva, 2014, S. 61f):

1. such NP as (NP,)* (or | and) NP
2. NP (, NP)* , or other NP
3. NP (, NP)* , and other NP
4. NP (,) including (NP,)* (or | and) NP
5. NP (,) especially (NP,)* (or | and) NP

Hierbei steht NP für eine Nominalphrase, * für beliebig viele Wiederholungen (auch keine) und | entspricht dem „entweder oder“. Tabelle 4.2 zeigt, welche Nominalphrasen dann als Hypernyme und Hyponyme aufgefasst werden.

In Tabelle 4.2 stellt die Perspektive dar, aus welcher Sicht die is-a-kind-of Beziehung definiert wurde: Aus Sicht eines Hypernyms oder aus Sicht von Hyponymen. Hearst Pattern werden vor allem dann angewendet, wenn taxonomische Beziehungen identifiziert werden sollen.

Diese Regeln müssen im Rahmen der Ontologieextraktion angepasst werden, da die Pattern nicht kennzeichnen, ob Relationen zwischen Istanzen und Klassen oder

Hearst Pattern	Beziehung	Perspektive
such NP_1 as $(NP_2,)^*$ (or | and) NP_3	NP_2 and NP_3 is-a-kind-of NP_1	NP_1 ist Hypernym von NP_2 und NP_3
NP_1 (, NP_2)* , or other NP_3	NP_1 and NP_2 is-a-kind-of NP_3	NP_1 und NP_2 sind Hyponyme von NP_3
NP_1 (, NP_2)* , and other NP_3	NP_1 and NP_2 is-a-kind-of NP_3	NP_1 und NP_2 sind Hyponyme von NP_3
NP_1 (,) including $(NP_2,)^*$ (or | and) NP_3	NP_2 and NP_3 is-a-kind-of NP_1	NP_1 ist Hypernym von NP_2 und NP_3
NP_1 (,) especially $(NP_2,)^*$ (or | and) NP_3	NP_2 and NP_3 is-a-kind-of NP_1	NP_1 ist Hypernym von NP_2 und NP_3

Tabelle 4.2: Hearst Pattern und die entsprechenden Hyponyme und Hypernyme

zwischen Subklassen und Superklassen identifiziert werden (Maynard und Bontcheva, 2014, S. 62). Eine Lösung, die vorgeschlagen wird, besteht darin, erst einen NER-Tagger auf dem Textkorpus anzuwenden und anschließend die so annotierten Entitäten als Kandidaten für Instanzen aufzufassen und alle anderen Nominalphrasen als Kandidaten für Konzepte. Im Zuge der Ontologieextraktion ist es aber nicht ungewöhnlich, dass vor der Relationsextraktion ein Algorithmus zur Konzeptextraktion angewendet wird. Dann könnten alle Nominalphrasen als Instanzen aufgefasst werden, die nicht bereits als Konzept, von einem vorher angewendeten Algorithmus, erkannt wurden. Darüber hinaus sollten nicht relevante Modifikatoren aus den Nominalphrasen entfernt werden, wie z.B. Artikel und Konjunktionen.

Neben diesen Pattern existieren noch andere Pattern (Maynard und Bontcheva, 2014, S. 62f), die die Hearst Pattern erweitern, um eine größere Ausbeute bei der Relationsextraktion zu erzielen. Hearst erzielte mit den nach ihm benannten Pattern eine Accuracy von 66% (MacCartney, S. 18).

Der Vorteil von Hearst Pattern liegt darin, dass Entwickler die volle Kontrolle über den Relationsextraktionsprozess besitzen, sodass häufig nur solche Relationen aus den Textkorpora extrahiert werden, die extrahiert werden sollen. Da jedoch kein Bezug auf die Abhängigkeiten zwischen Wörtern bzw. Nominalphrasen genommen wird, können auch Hearst Pattern Relationen falsch zuordnen. Ein Beispiel dafür wird in Abschnitt 6.3 gezeigt. Ein weiteres Problem ist, dass die Erzeugung manueller Pattern sehr viel

Zeit in Anspruch nehmen kann und bei großen bis sehr großen Textkorpora nie alle Relationen manuell kodiert werden können. Relationen, die dem Anwender unbekannt sind und für die deshalb keine Pattern erzeugt werden, werden nie gefunden. Ein Ansatz, der dieses Problem löst, ist der des *Bootstrapping* im nächsten Abschnitt.

4.4.2 Bootstrapping

Im Vergleich zu Lexico-Syntaktischen Pattern muss beim *Bootstrapping* nicht für jede Relation ein Pattern erzeugt werden. Eine initiale Menge von Instanzen einer bestimmten Relation, die sogenannte *seed*, dient einem Algorithmus als Basis für das Auffinden weiterer Relationen. Das Bootstrapping kann auch assoziative Relationen erkennen, d.h. nicht-taxonmische-Relationen.

Der Ablauf ist in Abbildung 4.4 dargestellt. Wenn beispielsweise die Relation *is CEO of* mit den initialen Instanzen *Tim Cook* und *Apple* gegeben ist, kann daraus das Pattern NP_1 *is CEO of* NP_2 extrahiert werden. Auch hierbei steht NP für eine Nominalphrase. Dieses Muster ist Teil des *Pattern Set*. Im Textkorpus kann dann nach Stellen gesucht werden, auf die dieses Muster passt. Wird eine Textstelle gefunden, können die Instanzen/Konzepte, die an dieser Stelle über die Relation assoziiert werden, extrahiert werden. Eine anschließende Suche nach den beiden Instanzen/Konzepten kann wiederum zu neuen Relationen führen. Ein möglicher Fund wäre z.B.: „Tim Cook was not the founder of Apple.“. Das so ermittelte neue Pattern wäre dann NP_1 *was not the founder of* NP_2.

Das Bootstrapping hat jedoch nach (Konstantinova, 2014, S. 19) die folgenden Schwächen:

- Frühe Fehler führen zu noch mehr Fehlern in späteren Durchläufen
- Unter gewissen Umständen kann ein *semantischer Drift* auftreten

Der semantische Drift betrifft die mangelnde Beachtung der Semantik. Wenn z.B. die Relation *is CEO of* betrachtet wird, sollten mit dieser Relation als Ausgangsbasis auch solche Relationen gefunden werden, die eine semantische Nähe zu dieser Relation haben, d.h. Relationen, die „wichtige“ Personen mit Unternehmen in Beziehung setzen, z.B. als Gründer oder CEO. Es könnte aber auch passieren, dass der Kontext der

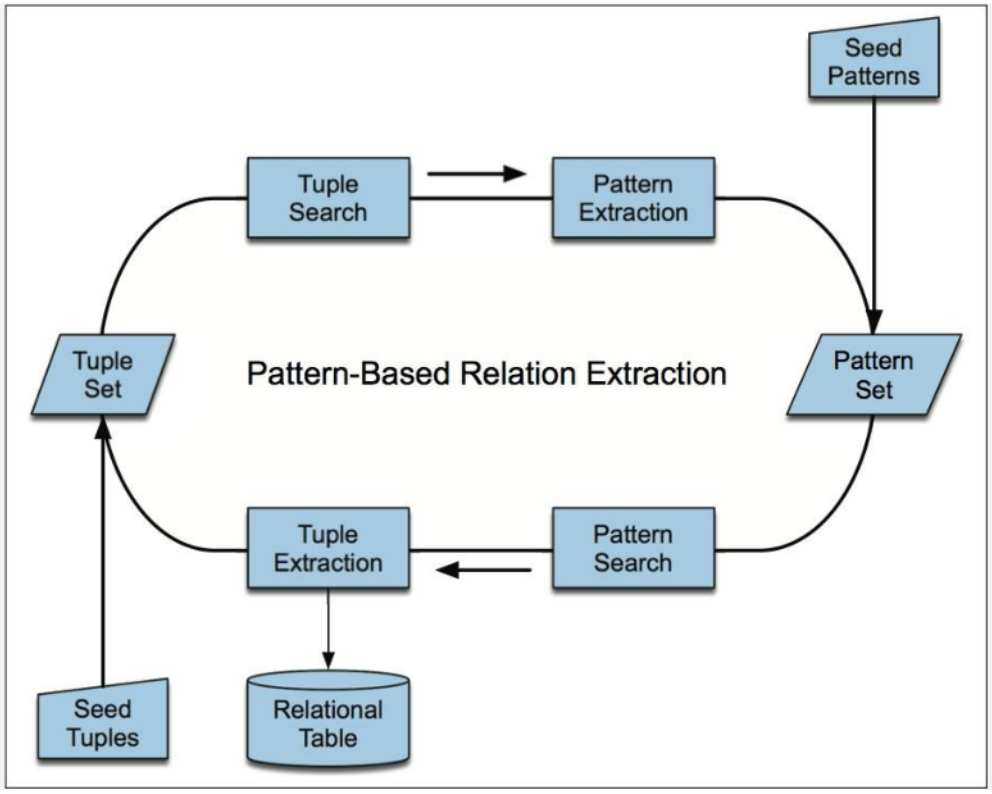

Abbildung 4.4: Ablauf Bootstrapping nach (MacCartney, S. 23)

Suche vollständig verlassen wird. Das Pattern NP_1 *was not the founder of* NP_2 würde beispielsweise auch „Julius Caesar was not the founder of Rome“ als zulässigen Satz erkennen. „Julius Caesar“ und „Rome“ würden wiederum zu Relationen führen, die mit der ursprünglichen Suche nichts zu tun hätten. Je früher diese Fehler gemacht werden, desto größer sind die Auswirkungen auf die späteren Durchläufe.

Eine Arbeit, die diese Probleme adressiert und eine Accuracy von 89% erzielt, stellt Kozareva (2012) dar. Generell spielt der semantische Drift in Korpora eine Rolle, die in der Anzahl der zu betrachtenden Dokumente nicht begrenzt sind, wie z.B. das World Wide Web. Aber auch in Korpora, die eine begrenzte Größe aufweisen, aber mehr als eine Domäne abdecken oder allgemein domänenunspezifisch sind, kann der semantische Drift beim Bootstrapping eintreten. Ob Bootstrapping die Methode der Wahl ist, wird also auch davon beeinflusst, welche Eigenschaften der Korpus aufweist, auf den

diese Methode angewendet wird. Im Falle der Extraktion einer domänenspezifischen Ontologie aus einem Korpus begrenzter Größe, der aus Dokumenten besteht, die eine Vorauswahl hinsichtlich ihrer Tauglichkeit zum Zwecke einer Ontologieextraktion durchlaufen haben, spielt der semantische Drift deshalb eine geringe Rolle.

Das Bootstrapping hat den Vorteil, durch relativ wenig Aufwand eine große Menge Relationen im Textkorpus auffinden zu können. Zusätzlich dazu können auch neue, unbekannte Relationen entdeckt werden. Der große Nachteil ist die relativ große Menge unerwünschter Relationen und Instanzen, sodass das Ergebnis des Bootstrapping sehr verrauscht sein kann. Die Relationsextraktion durch *überwachte Methoden* kann unter gewissen Umständen weniger verrauschte Ergebnisse erzielen, als das Bootsrapping.

4.4.3 Überwachte Methoden

Bootstrapping hat den Vorteil, dass im Gegensatz zu lexico-syntaktischen Pattern nicht jedes Pattern per Hand definiert werden muss. Da aber die Gefahr des semantischen Drifts vorliegt, kann es passieren, dass die so gewonnenen Daten relativ viel Rauschen (engl. Noise) enthalten.

Eine sicherere Methode, Relationen zwischen Konzepten und Instanzen zu identifizieren, ist die des überwachten Lernens. Beim überwachten Lernen geht es darum, dass ein Klassifizierer im Datenbestand Relationen zwischen Konzepten und Instanzen als solche markiert. Dafür müssen zunächst manuell oder (semi-)automatisch Trainingsbeispiele für den Klassifizierer vorbereitet werden. Ein Trainingsbeispiel kann die Form eines Satzes haben, in dem eine Relation zwei Entitäten assoziiert. Diesem Satz wird dann ein *Label* zugewiesen, dass der zu identifizierenden Relation entspricht. Für die Sätze werden eine Menge von Merkmalen, sogenannte *Features*, festgelegt, auf die hin der Klassifizierer die Sätze analysiert. Intern leistet der Klassifizierer eine Abbildung der Trainingsbeispiele auf die Merkmale und vergleicht die Merkmale neuer Sätze mit den gelernten. Ein Klassifizierer weist neuen Sätzen dann das Label zu, das er im Zusammenhang mit den Merkmalen des Satzes gelernt hat. Ein Klassifizierer wird auch bei den nachfolgenden Ansätzen dieses Abschnittes verwendet.

Ein relativ neuer Ansatz für das Auffinden von Relationen in Textkorpora ist der Ansatz der *Open information Extraction* (OIE). Die OIE adressiert zwei Probleme die bei

dem Ansatz des überwachten Lernens bestehen. Erstens wird kein manuell annotierter Textkorpus zum Erlernen von Relationen benötigt. Zweitens ermöglicht dieser Ansatz den Umgang mit heterogenen Inhalten aus dem World Wide Web, d.h. er ist domänenunabhängig. Der in Banko u. a. (2007) beschriebene Ansatz setzt dafür einen Naive Bayes Klassifizierer ein. Die Trainingsbeispiele werden erzeugt, indem ein *Dependency Parser* auf eine Teilmenge des Textkorpus angewendet wird. Einem Dependency Parser liegt eine *Dependency Grammatik* zugrunde. Durch diese Grammatik wird ein Wort als Kopf eines Satzes identifiziert und alle anderen Wörter werden als davon abhängig markiert, oder von einem anderen Wort, welches wiederum über eine Reihe von Abhängigkeiten vom Kopf des Satzes abhängig ist (Manning und Schütze, 2000, S. 428). Abbildung 4.5 zeigt das Beispiel erzeugter Abhängigkeiten eines Satzes. Die Pfeilrichtung zeigt die Abhängigkeit an, sodass beispielsweise ersichtlich ist, dass *The* und *old* auf *man* bezogen sind und nicht auf *rice*.

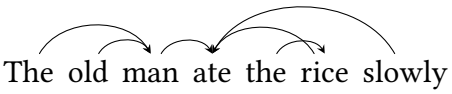

Abbildung 4.5: Dependency Struktur nach (Manning und Schütze, 2000, S. 428)

In diesem Dependency Graphen werden potentielle Entitäten und die Relation zwischen diesen identifiziert. Dafür werden die Nominalphrasen des Dependency Graphen ermittelt. Abbildung 4.6 zeigt eine mögliche Repräsentation eines Syntaxbaumes für den Satz aus Abbildung 4.5. Dabei werden *S, NP, DET, ADJ, N* und *V* als *Nichtterminalsymbole* bezeichnet, die durch *Terminalsymbole* (d.h. Wörter) beim Parsen ersetzt werden, wenn ein Satz der verwendeten Grammatik entspricht.

In Abbildung 4.6 würden die Entitäten $(e_i = man, e_j = rice)$ erkannt werden, da optionale Modifizierer in der OIE ignoriert werden.

Die Relation wird ermittelt, indem eine Sequenz von Wörtern auf Grundlage der geparsten Struktur, die zwischen den Entitäten liegt, ermittelt wird. In Abbildung 4.6 liegt nur das Verb *ate* zwischen den Entitäten. Das OIE-Tripel *(man, ate, rice)*, das so erzeugt wurde, stellt entweder ein positives Trainingsbeispiel dar, wenn *ate* tatsächlich eine Relation ist, sonst ein negatives Trainingsbeispiel. Wie das Trainingsbeispiel gelabelt wird, wird von einigen Heuristiken festgelegt. Auf diese Weise werden automatisch,

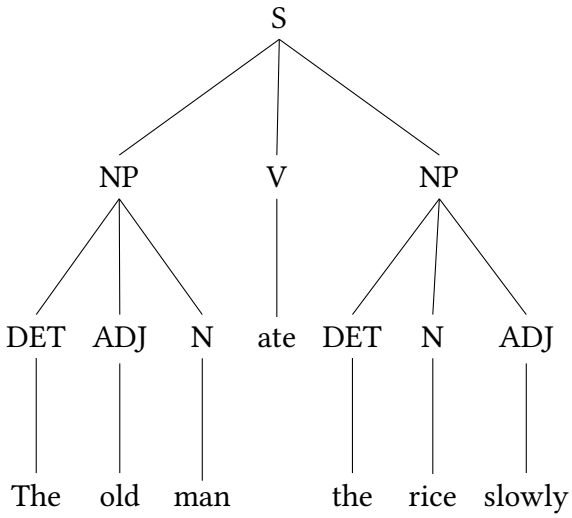

Abbildung 4.6: Beispiel eins Syntaxbaums. S = Sentence, NP = Nominalphrase, DET = Determiner, ADJ = Adjective, N = Noun, V = Verb.

ohne manuellen Aufwand, Trainingsbeispiele erzeugt, auf denen der Klassifizierer trainiert wird.

Nach dem Training extrahiert eine zweite Komponente, der *Single-Pass Extractor* (SPE), aus einem Textkorpus in derselben Art wie bereits dargestellt, OIE-Tripel. Diese werden dem trainierten Naive Bayes Klassifizierer als Eingabeparameter übergeben, welcher die OIE-Tripel als glaubwürdig oder unglaubwürdig klassifiziert, je nachdem ob das interne Modell des Klassifiziers in dem OIE-Tripel eine Relation erkennt oder nicht. Der SPE behält dann nur die OIE-Tripel, die als glaubwürdig vom Klassifizierer eingestuft wurden. Da Tripel mehrfach erzeugt werden können, merkt sich eine dritte Komponente, der *Redundancy-Based Assessor*, aus wie vielen verschiedenen Sätzen ein OIE Tripel erzeugt wurde. Ein probabilistisches Modell wird anschließend auf die verbleibenden OIE-Tripel angewendet, um diesen einen Wahrscheinlichkeitswert zuzuweisen. Ein zuvor festgelegter Schwellwert kann dann verwendet werden, um nur solche Relationen in der Ergebnismenge zu behalten, deren Wahrscheinlichkeitswert diesen Schwellwert nicht unterschreiten.

Das in Banko u. a. (2007) beschriebene System wurde auf einem Korpus von 9 Millionen Websites angewendet und erzeugte 7.8 Millionen Tripel von denen 80.4%

von menschlichen Bewertern als korrekt eingestuft wurden. Das System, das TextRunner genannt wurde, wurde in einigen Nachfolgesystemen optimiert. ReVerb enthält zusätzliche Heuristiken, die die Precision von TextRunner verbessern. Während TextRunner und ReVerb das *dependency parsing* nur in der Trainingsphase einsetzen, verwendet Ollie, das Nachfolgesystem von ReVerb, dieses auch außerhalb des Trainings (MacCartney, S. 77) um die Korrektheit der erzeugten Ergebnisse zu verbessern.

Während Ollie Bootstrapping einsetzt und die Effekte des semantischen Drifts erzeugt, arbeitet das in Angeli u. a. (2015) beschriebene System ohne Bootstrapping. Auf einem Teil des Textkorpus wird dafür ein Klassifizierer trainiert, der längere Sätze in kürzere Sätze zerlegt. Um den ursprünglichen Sinn der Sätze nicht zu verlieren, wird *Natural Logic* eingesetzt. Natural Logic ermöglicht es, aus Sätzen wie *Heinz Fischer of Austria visits China* den Satz *Heinz Fischer visits China* abzuleiten (Angeli u. a., 2015, S. 345). Diese kleineren Ausdrücke werden dann durch 14 manuell erzeugte Pattern in OIE-Tripel zerlegt. Das so entwickelte System wurde mit Ollie und OpenIE 4.0, dem aktuellen Nachfolger von Ollie, verglichen und erzielte eine F_1 Score von 22.7%, wodurch das System um 3.1% über der F_1 Score von Ollie und OpenIE 4.0 liegt.

Die Weiterentwicklung von TextRunner über ReVerb zu Ollie zeigt, dass von der ursprünglichen Idee, eine tiefere syntaktische Analyse der Sätze nur in der Trainingsphase vorzunehmen, abgewichen wurde. Dieses Vorgehen hat jedoch den Vorteil, dass die erzeugten Ergebnisse mit größerer Wahrscheinlichkeit korrekt sind. Die automatische Erzeugung einer Trainingsmenge wird durch die verwendeten Heuristiken gesteuert, sodass die Qualität der Trainingsdaten von der Qualität der Heuristiken abhängig ist. Für den Einsatz der Erzeugung domänenspezifischer Ontologien eignet sich der Ansatz der OIE nur bedingt, da die ursprüngliche Intention dahinter war, einen domänenunabhängigen Ansatz für das World Wide Web zu entwickeln. Dies schließt eine Anwendung der OIE auf domänenspezifische Textkorpora jedoch keineswegs aus. So könnten beispielsweise domänenspezifische Heuristiken steuern, ob ein Trainingsbeispiel als positiv oder negativ eingestuft wird.

Bei dem Vorgehen überwachter Methoden gilt, je mehr Trainingsdaten vorhanden sind und desto besser die Qualität dieser Trainingsdaten ist, desto bessere Ergebnisse

kann der Klassifizierer erzielen. Die potentiell hohe Genaugikeit bei der Extraktion von Relationen ist ein Vorteil dieses Ansatzes. Im Gegensatz zu lexico-syntaktischen Pattern wird auf *Features* einer Eingabe gearbeitet, statt auf konkreten Ausprägungen einer Eingabe (d.h. auf konkreten Wörtern/Phrasen). Insofern weist dieser Ansatz eine höhere Abstraktion auf, als der lexico-syntaktische Ansatz. Auf der anderen Seite ist die große Menge der Trainingsdaten, die für ein Erreichen einer hohen Genauigkeit erforderlich ist, ein wesentliches Problem überwachter Ansätze. Wie schon bei der manuellen Definition von Pattern kann der damit verbundene Aufwand sehr hoch sein. Wenn die Trainingsdaten automatisiert erzeugt werden, kann nie mit Sicherheit davon ausgegangen werden, dass die Trainingsbeispiele positiv oder negativ im Sinne einer Relation sind. Ebenso hat dieser Ansatz den Nachteil, dass er eine *perfect understanding assumption* umsetzt. Dies bedeutet, ein Klassifizierer kann einer Eingabe eine richtige Klasse zuweisen, wenn alle Merkmale oder die wesentlichsten Merkmale der zu identifizierenden Klasse, bekannt sind. Sind diese jedoch unbekannt, fällt es schwer dem Klassifizierer beizubringen, einer Eingabe die richtige Klasse zuzuweisen.

4.4.4 Semi-überwachte Methoden

Der Term *Distant Supervision* wurde in Mintz u. a. (2009) eingeführt und greift die Idee auf, eine externe Datenbank in den Prozess der Relationsextraktion einzubeziehen. Die Annahme, die diesem Ansatz zugrunde liegt, ist die, dass zwei Entitäten, die in dieser Datenbank über eine Relation miteinander verbunden sind, diese Relation auch in jedem Satz des Textkorpus ausdrücken. Deswegen wird ein Satz, der zwei Entitäten enthält, die in der Datenbank in Relation zueinander stehen, aus dem Textkorpus extrahiert. Jeder so identifizierte Satz wird dann mit der Relation gelabelt, die die Entitäten des Satzes in der Datenbank zueinander in Beziehung setzt. Auf der so erzeugten Menge von Sätzen wird dann ein Klassifizierer trainiert. In diesem Sinne dient die Datenbank der *entfernten Überwachung* des Lernprozesses.

Die Vorteile dieses Ansatzes liegen darin, dass die Trainingsmenge nicht manuell erzeugt werden muss und gleichzeitig die Qualität der Trainingsmenge durch die verwendete Datenbank sichergestellt wird. Ein grundsätzliches Problem ist, dass jedes Paar von Entitäten nur eine Relation besitzen kann. Dieses widerspricht aber der

natürlichen Logik, denn zwei Entitäten können in der Realität über mehrere Relationen miteinander verbunden sein. Ein genereller Ansatz, der dieses Problem angeht, ist der des *Multi-Instance Multi-Label* (MIML).

In (Xiang u. a., 2015, S. 250) werden die Probleme behandelt, dass die generierten Daten viel Rauschen enthalten und zwischen jedem Paar Entitäten nur eine Beziehung existieren kann. Dieses Problem wird durch Bewertungsmethoden angegangen, um das Rauschen in den Daten zu reduzieren und MIML zu ermöglichen.

4.4.5 Zusammenfassung

Es wurden vier Methoden der Relationsextraktion vorgestellt: *Lexico-syntaktische Pattern, Bootstrapping, Open Information Extratction* und *Distant Supervision*. Die *Lexico-syntaktischen* Pattern werden vor allem zur Identifikation taxonomischer Beziehungen zwischen Konzepten und zur Identifikation der Instanzen von Konzepten eingesetzt. Da die Relationen explizit in den verwendeten Mustern ausgedrückt werden, wird auf einer konkreten Eingabe gearbeitet. Dieser Umstand führt in der Regel zu einer geringen Menge extrahierter Relationen. Wie Abschnitt 6.3 verdeutlicht, können auch Lexico-Syntaktische Pattern falsche Ergebnisse produzieren. Ein weiterer Nachteil ist, dass der Aufwand der Erzeugung solcher Pattern hoch ist, wenn diese keine Standard-Pattern sind (wie z.B. Hearst-Pattern). Das *Bootstrapping* löst dieses Problem dadurch, dass nur eine initiale Menge von Entitäten, die über Relationen assoziiert sind, benötigt wird. Ausgehend davon werden auch neue Relationen im Textkorpus identifiziert. Dies führt zu einer größeren Menge von identifizierten Relationen, als dies durch lexico-syntaktische Pattern alleine möglich wäre. Die Genauigkeit der Relationsextraktion, die lexico-syntaktische aufweisen, leidet jedoch beim Bootstrapping aufgrund des *semantischen Drifts*. Um eine potentiell höhere Genauigkeit der Relationsextraktion, ohne die Einschränkungen der lexico-syntaktischen Pattern, zu erreichen, werden Ansätze des überwachten Lernens eingesetzt. Einer dieser Ansätze ist die *Open Information Extraction* (OIE). Der allgemeine Nachteil überwachter Ansätze, eine Trainingsmenge manuell zu annotieren, wird in diesem Ansatz dadurch umgangen, dass die Trainingsmenge automatisch erzeugt wird. Die Genaugikeit eines Klassifizierers hängt dann von den verwendeten Heurisitiken ab, die für die automatische Annotierung der Trai-

ningsbeispiele verwendet werden, sowie von den Merkmalen, auf Grundlage derer der Klassifizierer lernt. Die *Distant Supervision* geht davon aus, dass zwischen zwei Entitäten eines Satzes eine Relation besteht, wenn diese in einer Datenbank in Relation zueinander stehen. Wenn Trainingsdaten aufgrund dieser Annahme in positiv und negativ eingeteilt werden, kann dies die Qualität der Trainingsdaten erhöhen. Der Erfolg der Methode hängt jedoch von den Daten ab, die in der Datenbank vorhanden sind. Gleichzeitig kann wird nur das Erkennen von einer Relation zwischen einem Paar von Entitäten trainiert. Domänenspezifische Textkorpora enthalten jedoch Entitäten, die mit großer Wahrscheinlichkeit nicht in einer solchen Datenbank enthalten sind. Dies kann zur Folge haben, dass Relationen zwischen diesen Entitäten nicht erkannt werden.

4.5 Instanzextraktion

Die ABox einer Ontologie enthält Faktenwissen (siehe Abschnitt 2.4). Ein Teil davon wurde bereits durch die Konzeptextraktion (Abschnitt 4.3) und die Relationsextraktion (Abschnitt 4.4) ermittelt, indem die Konzepte und Relationen identifiziert wurden. In diesem Abschnitt geht es darum, die Erweiterungen der Konzepte (d.h. die Instanzen) im Text zu identifizieren.

4.5.1 Lexico-syntaktische Pattern

Wenn Konzepte bereits identifiziert wurden, aber die Instanzen noch nicht ermittelt wurden, können lexico-syntaktische Pattern, wie sie in Abschnitt 4.4.1 vorgestellt wurden, zur Ermittlung der Instanzen verwendet werden. In Fudholi u. a. (2016) wurde ein Standard NER-Tagger (siehe Abschnitt 4.2.4) verwendet, um Entitäten wie Personen, Lokationen und Organisationen im Text zu identifizieren. Zusätzlich wurde eine Gazetteer Liste verwendet, um den Identifikationsprozess zu verbessern. Eine Gazetteer Liste entspricht einer Liste von Instanzen, die zur selben konzeptuellen Klasse gehören (Chen u. a., 2016, S. 625). Das Lexico-syntaktische Pattern

(NPc) (as|including|include|especially) (a|an)? (NPi) ((,)(NPi))* ((and|or|to) (NPi))?

erkennt beispielsweise die Nominalphrasen *NPi*, welche die Instanzen enthalten, wenn die Nominalphrase *NPc* ein gegebenes Konzept darstellt (Fudholi u. a., 2016, S. 1120). Auf die Ergebnisse, die mit den Pattern ermittelt wurden, wurden verschiedene Filter angewendet, die darauf abzielten, die Ergebnisse zu verbessern. Beispielsweise wurden Phrasen mit speziellen Zeichen, Konjunktionen und Abkürzungen aus den Nominalphrasen entfernt. Fudholi u. a. (2016) erzielten mit diesem Ansatz eine Precision von 75.95% auf einer Testmenge von 54 Dokumenten, die zu identifizierende Instanzen enthielten.

4.5.2 Hybride Methoden

Hybride Methoden kombinieren mehrere Ansätze miteinander, um die ABox einer Ontologie zu erzeugen. In Wang und Cohen (2009) wurden lexico-syntaktische Pattern und Bootstrapping (siehe Abschnitt 4.4.2) verwendet. Das System, das die Autoren entwickelten, wurde ASIA (Automatic Set Instance Acquirer) genannt. In einem ersten Schritt wurden *Hearst-Pattern* verwendet, wie sie in Abschnitt 4.4.1 vorgestellt wurden. Der Textkorpus wurde jedoch keiner Vorverarbeitung unterzogen, weshalb nicht auf lexikalischen Features wie POS-Tags gearbeitet werden konnte. Die Vorverarbeitung wurde bewusst weggelassen, um den Ansatz sprachunabhängig zu gestalten. Dies hatte jedoch zur Folge, dass das Ergebnis des ersten Schrittes viel Rauschen enthielt.

Die Ergebnisse wurden entsprechend einer Metrik hinsichtlich ihrer Qualität gewichtet. Die aussichtsreichsten Ergebnisse (d.h. die Kandidaten-Instanzen) wurden einem externen System, Seal (Set Expander for Any Language), als Eingabeparameter übergeben. Seal ermittelt aus einer initialen Menge von Instanzen (engl. Seed) weitere Instanzen aus dem World Wide Web, die im Zusammenhang mit den Instanzen der Seed auftreten (Kollokation, siehe Abschnitt 4.3). Die Menge der Instanzen wurde in einem weiteren Schritt mit einem Nachfolgesystem von Seal anhand verschiedener Kriterien verfeinert. In einem vierten Schritt wurde *Bootstrapping* in Kombination mit einer iterativen Version von Seal verwendet. In jeder Iteration ermittelte der Bootstrapper 25 Webpages, die 3 konkatenierte Instanzen aus der Seed enthielten. In der ersten Iteration wählte der Bootstrapper zufällige Instanzen aus der Seed. In jeder darauf folgenden Iteration wählte der Bootstrapper zufällige Instanzen aus der

Ergebnissmenge der vorherigen Iteration aus, wenn diese auch in der initialen Menge enthalten waren. Das Ergebnis dieses letzten Schrittes war eine finale Menge von Instanzen.

Um ihre Ergebnisse zu vergleichen, verwendeten Wang und Cohen (2009) eine angepasste Version der *mean average precision* (MAP). Diese Metrik entspricht einer durchschnittlichen Precision über mehrere Anfragen bzw. Testmengen. Die MAP lag bei diesem Ansatz bei 82% über alle Testläufe.

4.5.3 Zusammenfassung

In diesem Abschnitt wurden zwei Methoden vorgestellt, Instanzen zu extrahieren. Die Schwierigkeit der Instanzextraktion ist nicht das Auffinden von Instanzen im Text, sondern von der Assoziation dieser Instanzen mit bereits identifizierten Konzepten. *Lexico-syntaktische Pattern* erzielen, wie bereits erwähnt, eine hohe Genauigkeit aber eine geringe Ausbeute. Der Ansatz von Wang und Cohen (2009) setzt solche Pattern in Kombination mit *Bootstrapping* ein, um die Menge der identifizierten Instanzen, durch zyklisches Hinzufügen von Instanzen, zu erhöhen. Dabei besteht jedoch die bereits erwähnte Gefahr des *semantischen Drifts*.

4.6 Axiomextraktion

Die Extraktion von Axiomen aus natürlichsprachlichen Texten ist die herausfordernste Tätigkeit auf dem Gebiet der Ontologieextraktion. Auch wenn Axiome eine Ontologie mit der Eigenschaft ausstatten, ausdrucksstarke Aussagen über Domänenwissen treffen zu können, wurde auf dem Gebiet der automatischen Extraktion von Axiomen aus natürlichsprachlichen Texten bisher wenig geforscht. Zum Zeitpunkt der Arbeit lagen keine Forschungsergebnisse im Hinblick auf die (semi-)automatische Erzeugung der *RBox* einer Ontologie vor, sodass die folgenden Abschnitte lediglich aktuelle Forschungsergebnisse der automatischen *TBox* Erzeugung vorstellen.

4.6.1 Erzeugung der TBox

Die TBox einer Ontologie enthält terminologisches Wissen (siehe Abschnitt 2.4). In diesem Abschnitt werden Ansätze aus den Bereichen des unüberwachten- und überwachten Lernens vorgestellt.

Unüberwachte Ansätze

In Völker u. a. (2015) wird *Association Rule Mining* (ARM) auf *Linked Data* angewendet um Disjointness-Axiome zu ermitteln. Association Rules sind in diesem Kontext Implikationen zwischen Konzepten ($K_i \rightarrow K_j$), im Allgemeinen können aber auch andere Elemente als Prämissen und Konklusionen solcher Implikationen dienen. Der Ansatz besteht aus fünf Schritten:

1. Terminologie-Extraktion
2. Instanz-Extraktion
3. Erzeugung einer Transaktionstabelle
4. Association Rule Mining
5. Axiom Generation

In dem vorgestellten Ansatz wird angemerkt, dass Methoden, die ARM umsetzen, auf sehr große Datenbestände ausgelegt sind. Für einen effizienten Zugriff auf diese Datenbestände wird die Anfragesprache SPARQL verwendet, welche die de-facto Standard-Anfragesprache für RDF-Stores ist. Die ersten beiden Schritte betreffen die Extraktion der Konzepte (1) und Instanzen (2) aus einem solchen RDF-Store, indem zwei einfache Anfragen (engl. query) gestellt werden. Anschließend wird eine Tabelle erzeugt, deren Zeileneinträge den Instanzen entsprechen und die als Spalteneinträge die Konzepte aufweist. Tabelle 4.3 zeigt dies beispielhaft.

Für jede Instanz wird in dieser Tabelle vermerkt, welche Konzepte sie aufweist (bzw. für jedes Konzept, welche Instanzen es als Erweiterung hat). Im Schritt 4 wird dann eine Spalte $\neg K_j$ hinzugefügt, sobald eine Instanz nicht dem Konzept K_j entspricht. Ist dies der Fall, wird für die Instanz in der Tabelle vermerkt, dass sie dem Konzept $\neg K_j$ entspricht. Tabelle 4.4 zeigt dies beispielhaft.

	K_1	K_2	...	K_j
I_1				
I_2				
...				
I_i				

Tabelle 4.3: Schema einer Transaktionstabelle. $I_i :=$ Instanz i, $K_j :=$ Konzept j.

Von den Autoren wird angemerkt, dass dies der *closed world assumption* entspricht, da ein fehlender Eintrag in der Spalte K_j nicht bedeutet, dass die Instanz in keinem Fall dem Konzept K_j entspricht, sondern dass die Daten dieses Wissen nicht beinhalten. Die Annahme, die diesem Vorgehen zugrunde liegt, ist jedoch, dass der große Datenbestand, der ARM-Verfahren zugrunde liegt, diesen Effekt relativiert.

	Mieter	**Eigentümer**	**¬Mieter**	**¬Eigentümer**
Tom		x	x	
Ben	x			x
Joe		x	x	

Tabelle 4.4: Beispiel einer erweiterten Transaktionstabelle. Der Eintrag x zeigt an, dass eine Instanz i einem Konzept j entspricht.

Die Definition der Konzeptinklusion ($K_i \sqsubseteq K_j$) legt fest, dass eine Teilmengenbeziehung zwischen K_i und K_j bestehen muss, damit eine solche Konzeptinklusion vorliegt. Wenn eine Transaktionstabelle mindestens eine Instanz aufweist, die zwei Konzepten K_i und K_j zugehörig ist, könnte also daraus die Assoziationsregel $K_i \rightarrow K_j$ abgeleitet werden. Die theoretische Möglichkeit besteht allerdings bereits, wenn die Transaktionstabelle noch die Gestalt der Tabelle 4.3 aufweist (Völker u. a., 2015, S. 128).

Die Defintion der Disjunktheit zweier Konzepte ($K_i \sqcap K_j \sqsubseteq \bot$) schreibt vor, dass zwei Konzepte nur dann disjunkt sind, wenn sie keine gemeinsamen Instanzen besitzen. Die Disjunktheit zweier Konzepte kann deshalb nur dann aus Assoziationsregeln abgeleitet werden, wenn beide Richtungen der Implikationen aus der Transaktionstabelle abgeleitet werden können, wie die Autoren anmerken. Dies bedeutet für zwei Konzepte K_i und K_j, dass diese disjunkt sind, wenn gilt:

$$(K_i \sqcap K_j \sqsubseteq \bot) \Leftrightarrow (K_i \rightarrow \neg K_j) \wedge (K_j \rightarrow \neg K_i)$$

Die Extraktion von Assoziationsregeln, die Disjunktheits-Axiomen entsprechen, wird in Völker u. a. (2015) durch ein von Borgelt und Kruse (2002) entwickeltes Programm durchgeführt. Dieses Programm erzeugt für jede Assoziationsregel einen *confidence value*. Der *confidence value* entspricht der „Sicherheit“, mit der eine erkannte Association Rule auch tatsächlich eine Association Rule ist. Dieser ist notwendig, da Assoziationsregeln, auch wenn sie im Sinne logischer Implikationen notiert werden, nicht gleichbedeutend mit diesen sind (Völker u. a., 2015, S. 127). Assoziationsregeln entsprechen tatsächlich nur bisherigen Gesetzmäßigkeiten eines Datenbestands. Zukünftige Transaktionen bzw. das Hinzufügen neuer Konzepte und Instanzen, können eine zuvor gültige Assoziationsregel ungültig werden lassen. Aus diesem Grund wird der *confidence value* benötigt.

Im fünften Schritt werden Association Rules paarweise gruppiert, wenn sie dieselben Klassen aufweisen, d.h. es werden Tupel der Form $(K_i \rightarrow \neg K_j, K_j \rightarrow \neg K_i)$ erzeugt. Ein zuvor festgelegter minimaler *confidence value* muss von je zwei Association Rules eines Tupels mindestens erreicht werden, damit ein Disjunktheits-Axiom $K_i \sqcap K_j \sqsubseteq \bot$ aus diesem Tupel extrahiert wird.

Dieser Ansatz ist im System GoldMiner umgesetzt, welches im Jahr 2015 dem aktuellen Stand der Forschung im Bereich der Axiom-Extraktion entsprach. Im Rahmen der Evaluation wurde in (Völker u. a., 2015, S.12) eine F_1-Score von bis zu $\approx 97\%$ mit GoldMiner erzielt. Im Gegensatz zu den Ansätzen, die in Abschnitt 4.6.1 vorgestellt werden, ist dieser Ansatz mit wenig Aufwand umsetzbar. Dies bedeutet, die Trainingsphase fällt weg, wodurch dieser Ansatz schneller umgesetzt werden kann. Die große Menge an Daten, die für eine erfolgreiche Umsetzung von *Association Rule Mining* Verfahren benötigt wird, relativiert sich bei einem Vergleich mit *machine learning* Verfahren, die für ein erfolgreiches Training ebenfalls viele Daten in Form von Trainingsbeispielen und Testdaten erfordern. Trotzdem dieser Ansatz unüberwacht ist, wurde damit eine sehr gute F_1-Score erzielt, wie sie sonst nur von Verfahren aus der Vorverarbeitung (siehe Abschnitt 4.2) bekannt sind.

Überwachte Ansätze

Überwachte Ansätze verwenden Machine Learning Verfahren, die Trainingsbeispiele voraussetzen, um eingesetzt werden zu können. Probabilistische Ansätze verwenden

Wahrscheinlichkeitswerte, um TBox Axiome für Ontologien zu erlernen. Der Vorteil dieser Ansätze liegt in der Fähigkeit, mit unsicherem Wissen umgehen zu können. Ergebnisse müssen nicht *wahr* oder *falsch* sein, was einer Wahrscheinlichkeit von einhundert Prozent oder null Prozent entspräche, sondern können mit einer gewissen Wahrscheinlichkeit wahr oder falsch sein. Der Ansatz von Zhu u. a. (2015) verwendet dafür Bayessche Netze.

Bayessche Netze sind gerichtete, azyklische Graphen. Die Knoten in einem solchen Graphen entsprechen Variablen der Wahrscheinlichkeitsrechnung. Gerichtete Kanten zwischen diesen Knoten entsprechen Abhängigkeiten, d.h. wenn eine Kante von A nach B führt, dann ist der Wert der Variablen A vom Wert der Variablen B abhängig. Ein Bayessches Netz im Sinne von Zhu u. a. (2015) wird BelNet$^+$ genannt, da es in der Syntax der Beschreibungslogik $\mathcal{ALC}$ Subsumptions- und Disjointness-Axiome darstellen kann und gleichzeitig die Semantik Bayesscher Netze aufweist. Ein Knoten in einem BelNet$^+$ entspricht einem Konzept bzw. einer Rolle. Ein BelNet$^+$ wird automatisch aus der ABox einer vorliegenden Ontologie erzeugt, indem ein *structural learning* Verfahren angewendet wird. Bei diesem Verfahren werden zunächst die Knoten eines Bayesschen Netzes aus der ABox einer Ontologie erzeugt, d.h. es werden Knoten für Relationen und Konzepte erzeugt. Zusätzlich können bereits an dieser Stelle *conditional links* (CL) zwischen Konzepten existieren. Ein CL vom Konzept A zum Konzept B bedeutet, dass der Wert von A vollständig den Wert von B festlegt (Zhu u. a., 2015, S. 32). Das heißt, wird der Knoten A instantiiert, steht sofort fest, welchen Wert der Knoten B annimmt. Abbildung 4.7 zeigt beispielhaft, dass der Wert des Knotens *Father* $\sqcup$ *Mother* vollständig von den Werten der Knoten *Father* und *Mother* abhängt. Dies gilt allerdings nur dann, wenn zwischen den Instanzen von *Father* und *Mother* eine bijektive Abbildung besteht.

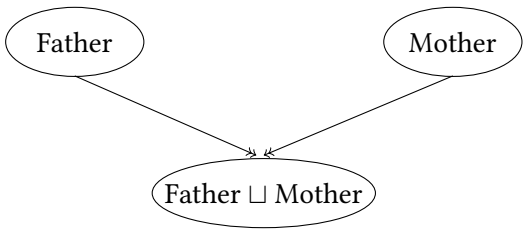

Abbildung 4.7: Ausschnitt eines initialen Belnet$^+$ nach (Zhu u. a., 2013, S. 762)

Die initiale Struktur aus Knoten und CLs wird iterativ um *dependency links* (DPL) erweitert, indem für je zwei Knoten überprüft wird, ob DPLs hinzugefügt/entfernt oder umgedreht werden sollten, falls diese bereits vorhanden sind. Ein DPL vom Knoten A zum Knoten B zeigt an, dass eine Instantiierung des Knotens A die noch möglichen Instantiierungen von B einschränkt. Für das Hinzufügen der DPLs wird eine Bewertungsfunktion verwendet, die für je zwei Knoten den Grad der Abhängigkeit ermittelt und überprüft, welche Operation diesen Grad erhöhen würde. Wenn eine der Operationen diesen Grad erhöhen würde, würde diese ausgewählt werden. Ein anschließendes Selektionskriterium überprüft, ob die von der Bewertungsfunktion vorgeschlagene Operation zwei Knoten in eine Subsumptions- oder Disjointness-Relation setzen würde. Ist dies nicht der Fall, wird die Operation verworfen.

Aus dem resultierenden BelNet$^+$ können im Anschluss TBox Axiome (Subsumptions- und Disjointness-Axiome) extrahiert werden, indem die Extraktionsregeln aus Zhu u. a. (2013) angewendet werden. Während Ansätze des maschinellen Lernens die *closed world assumption* (CWA) verwenden, liegt diesem Ansatz die *open world assumption* (OWA) zugrunde, da Wahrscheinlichkeitswerte verwendet werden, weshalb dieser Ansatz auf die Eigenschaften des (Semantic) Web ausgerichtet ist (Zhu u. a., 2015, S. 30). In einem direkten Vergleich von BelNet$^+$ mit anderen Systemen schnitt BelNet$^+$ bei Precision und F_1-Score deutlich besser ab. Bei diesem Ansatz muss trotz guter Ergebnisse darauf geachtet werden, dass ein *overfitting* nicht eintritt. Dahinter verbirgt sich die Gefahr, dass ein Lernverfahren zu stark auf vorhandene Daten angepasst ist, sodass auf den Testdaten sehr gute Ergebnisse erzielt werden können, auf Daten die nicht während des Lernens bzw. Testens verwendet wurden, jedoch deutlich schlechtere Werte erzielt werden. Diese Gefahr besteht jedoch auch bei den bereits vorgestellten Methoden des überwachten Lernens.

Ein klassischer Ansatz des maschinellen Lernens wird in Völker u. a. (2015) vorgestellt. In diesem Ansatz wird ein Klassifizierer trainiert, der Disjointness- und Subsumptions-Axiome in Texten erkennen soll. Dafür werden drei Arten von Features verwendet:

- Logische Features

- Lexikalische Features
- Korpus basierte Features

Logische Features wurden auf den Strukturen der Ontologie definiert. Hierzu zählt u.a. die taxonomische Überlappung, die die gemeinsamen Subklassen zweier Konzepte zählt und den Quotienten daraus bildet. Neben anderen Messungen können die so ermittelten Werte ein Indiz dafür sein, dass zwei Klassen disjunkt sind oder nicht. *Lexikalische Features* werden aus den Annotationen im Text und anderen lexikalischen Eigenschaften gewonnen. Hierzu zählen u.a. QGramme, die Teilzeichenketten der Länge Q von längeren Zeichenketten darstellen. Neben anderen Messungen wird durch QGramme die Gleichheit zweier Zeichenketten ermittelt. Die Annahme ist, dass Konzepte, die einen ähnlichen Post- oder Präfix in ihrer Bezeichnung aufweisen, in einem gewissen Maße ähnlich sind. Die Ähnlichkeit zweier Konzepte kann dann ein Indiz für Subsumption oder Disjointness dieser Konzepte sein. Als *Korpus basiertes Feature* wurde Hintergrundwissen in Form von Wikipedia-Artikeln über Konzepte, die im Textkorpus auftraten, verwendet. Mit diesem Hintergrundwissen wurden dann einige Ähnlichkeitsmetriken verwendet, die schon bei den *logischen Features* Anwendung fanden, um neue Subsumptions-Axiome zu erkennen.

Im Ergebnis wurde mit den lexikalischen Features eine F_1-Score von $\approx 95\%$ erzielt. Die Verwendung der logischen Features führte zu einer F_1-Score von $\approx 97\%$ und die Korpus basierten Features resultierten in einer F_1-Score von $\approx 96\%$.

Wird nur die F_1-Score betrachtet, dann gibt es keinen nennenswerten Vorteil dieses Ansatzes gegenüber dem in (Völker u. a., 2015, S. 124ff) beschriebenen und in GoldMiner verwendeten Ansatz. Der unüberwachte Ansatz des *Association Rule Mining* (ARM, siehe Abschnitt 4.6.1) erzielt wie der überwachte Ansatz des *Machine Learning* gute Ergebnisse, sofern genügend Daten vorhanden sind. Im Gegensatz zum Machine Learning liegt der Vorteil des ARM im geringeren Aufwand, da keine Trainingsdaten erzeugt werden müssen.

4.6.2 Zusammenfassung

Dieser Abschnitt betrachtete die Ermittlung von Subsumptions- und Disjointness-Axiomen. Es wurden drei Ansätze vorgestellt: *Association Rule Mining*, *Maschine Lear-*

ning und *Bayessche Netze*. *Association Rule Mining* liegt die *closed world assumption* zugrunde. Dies bedeutet, es wird die Disjunktheit zwischen zwei Konzepten angenommen, wenn diese keine gemeinsamen Instanzen aufweisen. Diese Annahme kann vertreten werden, wenn die Menge der Verfügbaren Daten sehr groß ist. Ist diese jedoch nicht ausreichend groß, kann es sein, dass ein Disjointness-Axiom fälschlicher Weise zwischen zwei Konzepten angenommen wird. Im *Maschine Learning* wurde ein Klassifizierer trainiert. Dieses Vorgehen erfordert jedoch eine große Menge manuell annotierter Trainingsbeispiele, wenn kein Verfahren verwendet wird, dass diese automatisiert erzeugt. Dem in Zhu u. a. (2015) vorgestellten Ansatz liegt die *open world assumption* zugrunde. Dies ist ein Vorteil gegenüber den Ansätzen des *Machine Learning* und des *Association Rule Mining*.

5 Evaluierung

Für Endanwender/innen, Entwickler/innen und Forscher/innen ist die Antwort auf die Frage entscheidend, ob eine bestimmte Ontologie ihren Erwartungen entspricht. Die Evaluierung von Ontologien soll diese Frage beantworten. Die folgenden Abschnitte geben eine Übersicht über das Gebiet der Evaluation von Ontologien. Abschließend wird eine mögliche Vorgehensweise hinsichtlich der Evaluation von Ontologien vorgestellt.

5.1 Begriffsbestimmung

Gómez-Pérez (2001) beschreibt die Evaluierung von Ontologien als das Sicherstellen, dass die Definitionen der Ontologie die ontologischen Anforderungen und Kompetenzfragen (Anfragen an die Ontologie) richtig umsetzen bzw. dass die Ontologie in der echten Welt korrekt arbeitet (Gómez-Pérez, 2001, S. 393f). Die Definitionen einer Ontologie sind nach dieser Definition in einer natürlichen und einer formalen Sprache geschrieben. Entlang dieser Definitionen wird im Zuge der Evaluierung überprüft, welche Dinge von einer Ontologie korrekt definiert werden, welche nicht definiert werden und welche Dinge falsch definiert werden. Dabei werden Definitionen und Axiome überprüft, die von einer Ontologie explizit definiert werden, doch auch solche, die von anderen Ontologien importiert wurden und solche, die aus anderen Definitionen und Axiomen abgeleitet werden können. Das Ziel der Evaluierung ist nach dieser Definition der Beweis, dass das formale Modell der Ontologie in Bezug auf den Gegenstand, der modelliert wurde, korrekt ist. In (Gómez-Pérez, 2004, S. 255f) wird dieses Ziel als die *Validierung* der Ontologie bezeichnet und die Überprüfung der Definitionen und Axiome als die *Verifikation* der Ontologie. Demnach entspricht die Evaluierung einer Ontologie der Validierung und Verifikation einer Ontologie. Vrandečić (2010) zählt zur Verifikation die Überprüfung der gesamten Kodierung einer Ontologie, von zirkulären

Abhängigkeiten in der Klassenhierarchie bis zu inkonsistenten Namensgebungen von Konzepten, Instanzen und Relationen (Vrandečić, 2010, S. 49).

5.2 Kategorisierung von Evaluierungsansätzen

Evaluierungsansätze können nach (Brewster u. a., 2004, S. 3) in *quantitative* und *qualitative* Ansätze unterschieden werden. Ein qualitativer Ansatz bindet Anwenderinnen und Anwender in den Prozess der Evaluierung einer Ontologie ein. Diese Anwender und Anwenderinnen beurteilen die Qualität einer Ontologie in Bezug auf festgelegte Kriterien. Die Schwierigkeit dieser Ansätze besteht darin, die Personengruppen auszuwählen, die für eine solche Beurteilung geeignet sind. In Betracht kommen beispielsweise Endanwender/innen und Entwickler/innen aber auch Domänenexpertinnen und Domänenexperten. Diese Personengruppen haben jede für sich andere Ansprüche, sodass die Auswahl der richtigen Kriterien, entlang derer die Qualität einer Ontologie beurteilt werden soll, schwierig ist. Eine weitere Schwierigkeit besteht darin, dass manche Personengruppen nicht geeignet sind, die formalen Eigenschaften einer Ontologie zu beurteilen. Auf der anderen Seite beurteilt ein quantitativer Ansatz die Qualität einer Ontologie in Bezug auf messbare Kriterien, wie beispielsweise die Anzahl der Konzepte und die Anzahl der Instanzen eines Konzeptes. Mit diesen Messungen sind dann bestimmte Annahmen verbunden. Eine Annahme könnte beispielsweise sein, dass ein Konzept, dass nur eine Instanz im Nachbereich aufweist, als Konzept ungeeignet ist.

5.3 Dimensionen der Evaluierung

Die Evaluierung einer Ontologie kann aus verschiedenen Blickwinkeln erfolgen, die im Folgenden als *Dimension* bezeichnet werden. Innerhalb einer Dimension können dann *qualitative* und *quantitative* Kriterien identifiziert werden, nach denen die Qualität einer Ontologie beurteilt wird. In (Gangemi u. a., 2005, S. 8) werden die folgenden drei Dimensionen identifiziert:

Strukturell. Diese Dimension entspricht der Betrachtung einer Ontologie als Graph. Da die Ontologie als Graph betrachtet wird, können quantitative Metriken definiert werden, die die Struktur des Graphen messen. Diese Metriken können z.B. die Topologie des Graphen und/oder die Logik innerhalb des Graphen messen. Eine Metrik, die die Logik des Graphen betrachtet, kann beispielsweise überprüfen, ob es Zyklen im Graphen gibt. Mit einer Metrik, die die Gestalt des Graphen betrachtet, könnte ermittelt werden, ob der Graph ungünstige Strukturen aufweist, wie z.B. zu lange Pfade, und ob es möglich wäre, diese Pfade zu verkürzen.

Funktional. In dieser Dimension können Kriterien definiert werden, anhand derer festgestellt werden kann, ob die Ontologie ihren Verwendungszweck erfüllt. Bezugnehmend auf (Gómez-Pérez, 2004, S. 255f) wird in dieser Dimension die Validierung durchgeführt.

Benutzbarkeit. Kriterien, die in dieser Dimension festgelegt werden, zielen darauf ab, festzustellen wie benutzerfreundlich der Umgang mit einer Ontologie ist. Beispielsweise kann ein Kriterium ermitteln, wie einfach oder schwierig Anwender bestimmte Eigenschaften der Ontologie finden können.

In (Poveda-Villalón u. a., 2012, S. 263f) wird die Validierung von Ontologien mit einem Online-Tool betrachtet. Im Zuge dieser Betrachtung wurden verschiedene Tools miteinander verglichen und es wurden sechs Dimensionen identifiziert, anhand derer die Qualität einer Ontologie ermittelt werden kann:

Verständlichkeit. Diese Dimension entspricht der Dimension *Benutzbarkeit* in (Gangemi u. a., 2005, S. 8).

Logische Konsistenz. In dieser Dimension wird betrachtet, ob eine Ontologie logische Inkonsistenzen aufweist oder potentiell aufweisen könnte. Dies wäre z.B. der Fall, wenn zwei Konzepte die gleichen Instanzen besitzen, aber ein Ungleichheits-Axiom zwischen diesen definiert wurde.

Modellierungsaspekt. In dieser Dimension wird betrachtet, ob die Ontologie unter Verwendung von Primitiven (z.B. *subclassOf*) einer Implementierungssprache

(z.B. OWL2) entwickelt wurde und ob dies korrekt erfolgte. Außerdem wird in dieser Dimension überprüft, ob es Modellierungsentscheidungen gab, die verbessert werden könnten.

Sprachspezifikation. In dieser Dimension wird betrachtet, ob die Implementierung einer Ontologie in Bezug auf die Syntax einer Implementierungssprache korrekt erfolgte.

Modell-Wirklichkeit. In dieser Dimension wird überprüft, inwieweit die Ontologie die Domäne repräsentiert, für die sie entwickelt wurde.

Semantische Anwendungen. In dieser Dimension wird überprüft, ob die Ontologie für die Anwendungen, von denen sie verwendet werden soll, geeignet ist, indem z.B. die Kompatibilität in Bezug auf verwendete Formate überprüft wird.

Die von Poveda-Villalón u. a. (2012) aufgeführten Dimensionen zeigen teilweise Überschneidungen mit den in Gangemi u. a. (2005) aufgeführten Dimensionen. Die Dimension *Logische Konsistenz* lässt sich auch in der Dimension *Strukturell* wiederfinden und die Dimension *Modell-Wirklichkeit* deckt sich mit der Dimension *Funktional*. Solche Überschneidungen lassen sich auch bei anderen Autoren entdecken (z.B. in (Neuhaus u. a., 2013, S. 181)), sodass ein gewisser (impliziter) Konsens in Bezug auf solche Dimensionen vorhanden ist. Im Zuge der Evaluierung muss festgelegt werden, welche Dimensionen evaluiert werden sollen. Beispielsweise ist eine Evaluierung einer Ontologie in der Dimension *Semantische Anwendungen* nur dann nötig, wenn die Ontologie in der Dimension *Funktional* ein ausreichendes Maß an Qualität aufweist.

5.4 Evaluierungsansätze

Die folgenden vier Evaluierungsansätze können unterschieden werden:

Benutzerzentrierte Ansätze. Bei diesen Ansätzen steht der Mensch im Mittelpunkt, d.h. die Qualität der Ontologie wird aus der (subjektiven) Perspektive von einer oder mehreren Personen beurteilt. In Supekar (2005) wird der Aufbau einer Metadaten-Ontologie beschrieben, die zusätzliche Informationen über eine Ontologie in Form von Metadaten bereithält. Beispiele solcher Metadaten sind

Informationen über die Domäne einer Ontologie und Erfahrungsberichte von Anwendern der Ontologie. Auf Grundlage dieser Informationen können mehrere Anwender die Qualität einer Ontologie beurteilen. In Smith (2008) werden Peer Reviews einer Ontologie von drei Rollen durchgeführt. Jede Rolle kann von mehreren Personen zur selben Zeit eingenommen werden. *Coordinate Editors* haben die Funktion, die Projektinteraktion von verschiedenen ontologiebezogenen Entwicklungsprojekten zu harmonisieren. *Associate Editors* sind die Entwickler einer Ontologie und *Ad-hoc Reviewer* sind Personen mit besonderen Kenntnissen. Die ersten beiden Personengruppen führen die Reviews durch, die Ad-hoc Reviewer werden nur angefordert, wenn sie benötigt werden.

Benutzerzentrierte Ansätze weisen den Vorteil auf, Qualitätsmerkmale beurteilen zu können, die nicht in Zahlen gemessen werden können. Dazu zählt u.a. die *Verständlichkeit*, die besser von Menschen als von Maschinen beurteilt werden kann. Doch dieser Ansatz weist Schwächen auf, wenn es darum geht, Beurteilungen der Personengruppen zu vergleichen bzw. diese durch vorgegebene Bewertungskriterien vergleichbar zu machen. Außerdem ist es schwierig, geeignete Personengruppen auszuwählen.

Anwendungszentrierte Ansätze. Ansätze, die anwendungszentriert sind, evaluieren die Qualität einer Ontologie im Kontext einer Anwendung. Ansätze dieser Art werden teilweise mit aufgabenzentrierten Ansätzen kombiniert, sodass die Qualität einer Ontologie daran gemessen wird, wie gut eine Anwendung eine Aufgabe, unter Verwendung der Ontologie, erfüllen kann. In (Porzel und Malaka, 2004, S. 2f) wird eine Anwendung als ein Algorithmus betrachtet, der eine bestimmte Aufgabe erfüllen muss. Es wurde die Aufgabe gewählt, Relationen zwischen Konzepten zu identifizieren. Dafür wurde ein Korpus erzeugt, der bereits annotierte Konzepte enthielt, zwischen denen Relationen mit Hilfe einer Ontologie identifiziert werden sollten. Die Auswertung erfolgte durch die Messung von Fehlern bei der Erkennung von Relationen.

Viele Ontologien werden nach ihrer Entwicklung von Anwendungen zur Erfüllung bestimmter Aufgaben verwendet. Aus diesem Grund ist diese Evaluationsumgebung für solche Ontologien sehr gut geeignet. Auf der anderen Seite

gibt es Ontologien, die nicht von Anwendungen verwendet werden sondern von Menschen. Ein denkbares Szenario sind E-Learning-Plattformen, aber auch andere Medien kommen in Betracht, durch Ontologien Wissen zu vermitteln bzw. zu veranschauchlichen. In diesen Anwendungsfällen sind benutzerzentrierte Ansätze deutlich besser für die Evaluation geeignet.

Aufgabenzentrierte Ansätze. Ein aufgabenzentrierter Ansatz evaluiert eine Ontologie in Bezug auf eine konkrete Aufgabe. In (Porzel und Malaka, 2004, S. 2) wird gefordert, dass eine solche Aufgabe ausreichend komplex sein muss, um eine geeignete Bewertung einer Ontologie vornehmen zu können. Die Autoren betonen, dass es in Bezug auf das Messen der Performance einer Ontologie, gerade bei assoziativen Relationen (siehe Abschnitt 2.2.2) wichtig ist, Aufgaben zu definieren, durch die unterschiedliche Modellierungen dieser Assoziationen Auswirkungen auf die Performance haben.

Wie die anwendungszentrierten Ansätze auch, bieten die aufgabenzentrierten Ansätze den Vorteil, dass sie eine Ontologie bereits in dem Kontext evaluieren, in dem die Ontologie eingesetzt werden soll. Da Ontologien in der Informatik meistens von Maschinen verwendet werden, erfolgt die Durchführung der Aufgabe innerhalb einer Anwendung. Bei diesem Ansatz muss deshalb darauf geachtet werden, dass die Modellierung der Aufgabe so erfolgt, dass die Performanz der Anwendung keine Auswirkung auf das Ergebnis.

Datenzentrierte Ansätze. Datenzentrierte Ansätze evaluieren Ontologien in Bezug auf Daten bzw. Korpora. In Brewster u. a. (2004) wird dies in drei Schritten getan. Im ersten Schritt werden aus dem Korpus automatisch Terme extrahiert. In einem zweiten Schritt wurde WordNet verwendet, um Hypernyme der Terme zu identifizieren. Die Hypernyme dienten dem Zweck, als Stellvertreter der Konzepte der Ontologie zu dienen, sodass ein Term als in der Ontologie vertreten angesehen wurde, wenn eines seiner Hypernyme in ihr vertreten war. In einem dritten Schritt wurde überprüft, wie viele der Terme und Hypernyme in der Ontologie vertreten waren.

Datenzentrierte Ansätze bieten für Ontologien, die in der Informationsextraktion eingesetzt werden, den Vorteil, dass sie diese Ontologien im Kontext ihrer späte-

ren Anwendung evaluieren. Allerdings ist dieser Ansatz auch durch die Menge der Daten beschränkt, auf denen eine Ontologie getestet wird. Dies bedeutet, für diese Ansätze müssen relativ viele Daten vorliegen, damit beurteilt werden kann, ob die Konzepte und Relationen einer Ontologie die Terme und Relationen einer Domäne abdecken oder nicht. Gleichzeitig kann nicht sicher behauptet werden, ob es weitere Daten gibt, auf denen die Ontologie schlechtere Ergebnisse erzielen würde.

Neben den hier vorgestellten Ansätzen existieren auch andere Kategorisierungen von Ansätzen der Evaluation von Ontologien, die allerdings ungeeignet erscheinen. So gibt es die Möglichkeit, eine Ontologie mit einer anderen zu vergleichen und auf diese Weise zu evaluieren. Da sich selbst Ontologien derselben Domäne stark voneinander unterscheiden können, sind Ontologien im allgemeinen unvergleichbar (Witte und Mülle, 2006, S. 127). In (Yu u. a., 2007, S. 224f) werden auch kriterienbasierte Ansätze als mögliche Kategorie aufgeführt. Gemeint ist damit, dass eine Ontologie in Bezug auf bestimmte Qualitätskriterien evaluiert wird. Qualitätskriterien, auf deren Grundlage eine Ontologie evaluiert wird, werden allerdings in jedem Ansatz verwendet, sodass dieser Evaluierungsansatz in jedem anderen Ansatz enthalten ist und deshalb keinen eigenständigen Evaluierungsansatz darstellt.

5.5 Evaluierungsgegenstände

Wenn eine Ontologie evaluiert werden soll, dann muss eine Festlegung auf den Evaluierungsgegenstand innerhalb einer Ontologie erfolgen, d.h. auf den *Aspekt* einer Ontologie, der evaluiert werden soll. In Bezug auf die Verifikation einer Ontologie werden in (Vrandečić, 2010, S. 61f) sechs Aspekte identifiziert, die für eine automatische und domänen- sowie taskunabhängige Verifikation einer Ontologie geeignet sind:

Vokabular. Das Vokabular einer Ontologie besteht aus den Konzepten und Relationen einer Ontologie. Diese können in URIs oder Literalen vorliegen. In Abhängigkeit der Kodierung können unterschiedliche Methoden verwendet werden, um das Vokabular zu verifizieren. Im Kontext dieser Arbeit werden nur Methoden betrachtet, die Literale verifizieren. Ein Literal wird in einem bestimmten For-

mat kodiert, z.B. als Zeichenkette, Datum oder Zahl. Eine Ontologiesprache wie OWL bietet neben solchen Kodierungen auch die Möglichkeit zusätzlicher Kennzeichnungen von Literalen. Diese Kennzeichnungen umfassen u.a. die *Typisierung* und die *Kommentierung* von Literalen (Vrandečić, 2010, S. 75f). Die Typisierung entspricht dem Datentyp eines Literals (z.B. String oder Integer). In der Evaluierung sollte eine Menge erlaubter Datentypen festgelegt werden, sodass anschließend überprüft werden kann, ob diese Datentypen eingehalten wurden. In Bezug auf die Ontologiesprache sollten nur solche Datentypen für Literale zugelassen werden, die von dieser Ontologiesprache unterstützt werden. Kommentare können für eine spätere Fehlersuche in einer Ontologie zusätzliche Informationen bereitstellen, wie z.B. Verweise auf die Stellen, an denen eine Relation zwischen zwei Konzepten identifiziert wurde.

Syntax. Die Beschreibung einer Ontologie erfolgt entlang einer verwendeten Sprache, welche eine Syntax aufweist. Nach (Vrandečić, 2010, S. 83) können solche Syntaxen eine Ontologie als Graph (z.B. RDF/XML) oder direkt (z.B. OWL/XML) beschreiben. Die Evalulierung der Syntax einer Ontologie muss sich an den jeweiligen Standards der verwendeten Sprache orientieren.

Struktur. Wird von der Struktur einer Ontologie gesprochen, dann ist damit die Gestalt der Ontologie als Graph gemeint (siehe Abschnitt 5.3). Diese Struktur kann z.B. in Form von RDF-Tripeln vorliegen, auf der gewisse Metriken definiert werden können, die beispielsweise auf zirkuläre Abhängigkeiten prüfen (Vrandečić, 2010, S. 99).

Semantik. Die Semantik einer Ontologie betrifft die Ebene der Logik.

Repräsentation. Die Repräsentation einer Ontologie betrifft die Art, wie die *Semantik* der Ontologie *strukturell* repräsentiert ist (Vrandečić, 2010, S. 143ff). Die Repräsentation kann von Menschen, wie z.B. Domänenexperten, evaluiert werden oder automatisch durch Metriken.

Kontext. In Bezug auf die Evaluation einer Ontologie umfasst der Kontext einer Ontologie zusätzliche Quellen, die eine Ontologie oder ihre Eigenschaften beschreiben

(Vrandečić, 2010, S. 62). Bestandteil des Kontext können z.B. formulierte Fragen sein, die die Ontologie beantworten soll. In der Evaluation kann dann überprüft werden, ob eine Ontologie die zuvor spezifizierten Eigenschaften aufweist und zuvor festgelegte Fragen beantworten kann.

Die verschiedenen Aspekte, unter denen eine Ontologie evaluiert werden kann, können in Bezug auf Qualitätskriterien untersucht werden. Diese werden im nächsten Abschnitt vorgestellt.

5.6 Qualitätskriterien

Bisher erfolgte in der Fachwelt noch keine Festlegung auf eine Menge von Kriterien, anhand derer die Qualität einer Ontologie beurteilt werden kann. Wie Abschnitt 5.5 gezeigt hat, gibt es verschiedene Aspekte, unter denen eine Ontologie evauliert werden kann, sodass eine einheitliche Menge von Qualitätsmerkmalen nur schwierig für alle Aspekte gemeinsam definiert werden kann. In (Vrandečić, 2010, S. 53ff) wird eine Übersicht über die in der Literatur bisher verwendeten Qualitätskriterien gegeben und zugleich eine Zusammenfassung dieser Kriterien in acht Qualitätsmerkmale vorgeschlagen:

Accuracy. Die Accuracy (dt. Genauigkeit) ist ein Qualitätskriterium, das angibt, ob die Axiome der Ontologie mit dem Wissen der Personengruppen über die jeweilige Domäne übereinstimmen (Vrandečić, 2010, S. 56). Dies bedeutet, dass die Ontologie keine falschen Aussagen darstellen sollte.

Adaptability. Die Adaptability (dt. Anwendbarkeit) einer Ontologie gibt an, inwieweit eine Ontologie auf ihren Verwendungszweck anpassbar ist (Vrandečić, 2010, S. 56f). Dies bedeutet, kleine Änderungen auf der Ebene der Axiome sollten zu vorhersagbarem Verhalten der Ontologie führen und die Erweiterung einer Ontologie kann durchgeführt werden, ohne Axiome zu entfernen.

Clarity. Die Clarity (dt. Deutlichkeit/Klarheit) einer Ontologie misst, inwiefern die Ontologie die beabsichtigte Bedeutung kommunizieren kann (Vrandečić, 2010, S. 57). Dazu zählt u.a., dass die in der Ontologie verwendeten Namen verständlich

und nicht mehrdeutig sein sollten und dass das Ausmaß der Dokumentation von Entitäten, Klassen und Relationen in einem geeigneten Maße erfolgen sollte.

Completeness. Die Completeness (dt. Vollständigkeit) einer Ontologie misst, in wie weit die Zieldomäne von der Ontologie abgedeckt wird (Vrandečić, 2010, S. 57f). Die Vollständigkeit kann in 3 Ebenen gemessen werden: (1) Auf der Ebene der Sprache bezeichnet sie den Grad, mit dem die Fähigkeiten der Sprache ausgereizt wurden, (2) auf der Ebene der Domäne den Grad, mit dem die Individuen und Konzepte der Domäne abgedeckt wurden und (3) auf der Ebene der Anforderungen einer Anwendung den Grad, mit dem alle nötigen Informationen vorhanden sind.

Computational Efficiency. Die Computational Efficiency misst, wie gut Werkzeuge, die auf der Ontologie arbeiten, bestimmte Aufgaben mit der Ontologie ausführen können (Vrandečić, 2010, S. 58). Zu diesen Werkzeugen zählen z.B. Schlussfolgerungssysteme, die die Ontologie als Wissensbasis verwenden, um daraus Antworten abzuleiten.

Conciseness. Die Conciseness (dt. Redundanz) misst, inwieweit eine Ontologie überflüssige Elemente in Bezug auf die Domäne enthält (Vrandečić, 2010, S. 58).

Consistency. Die Consistency (dt. Konsistenz) beschreibt, ob eine Ontologie Aussagen ermöglicht, die sich gegenseitig ausschließen (Vrandečić, 2010, S. 59). Die Konsistenz betrifft nicht nur die logische Ebene, sondern auch die Ebene der Kommentare von Klassen und Relationen, die mit den Axiomen übereinstimmen sollten, sowie die Übereinstimmung der Kommentare mit einer Gestaltungsrichtlinie. Wenn es möglich ist, in der Ontologie Label für Konzepte, Relationen und/oder Instanzen zu vergeben, dann zählt auch die Überprüfung der Label auf gewisse Gestaltungsrichtlinien zur Konsistenzüberprüfung.

Organization Fitness. Die Organization Fitness (dt. Organisationale Brauchbarkeit) umfasst Kriterien, die Rückschlüsse darauf zulassen, inwiefern eine Ontologie von einer Organisation verwendet werden kann (Vrandečić, 2010, S. 59f). Dazu zählen u.a. für die Ontologie verwendbare Werkzeuge, Bibliotheken und andere

Datenquellen, die eine Organisation im Zusammenhang mit einer Ontologie verwenden kann.

Diese Qualitätskriterien können durch geeignete Metriken und Methoden ermittelt werden. Ein kleiner Ausschnitt einer sehr großen Menge von Metriken und Methoden wird im folgenden Abschnitt vorgestellt. Für einen umfassenderen Blick auf eine sehr große Menge von Methoden und Metriken wird im selben Kapitel auf weiterführende Literatur verwiesen.

5.7 Qualitätserhebung

Die Methoden der Qualitätserhebung können in *Metriken* und *manuelle Methoden* unterschieden werden. In (Tartir u. a., 2010, S. 22f) werden in Bezug auf die *Struktur* einer Ontologie die folgenden drei Metriken definiert:

Clarity/Efficency. Die **relationship richness (RR)** gibt den Grad an, mit dem die Ontologie Relationen aufweist, die nicht nur hierarchisch sind. Die Annahme ist, dass eine Ontologie, die nur Vererbungsrelationen (is-a Relationen) beinhaltet, weniger Informationen enthält als eine Ontologie, die assoziative Relationen enthält. Die RR ist definiert als $RR = \frac{|P|}{|H| + |P|}$ mit $|P|$ als die Anzahl der Relationen, die keine Vererbungsbeziehung ausdrücken und $|H|$ als die Anzahl der Relationen, die Vererbungsbeziehungen ausdrücken.

Die **inheritance richness (IR)** misst die Verteilung von Informationen innerhalb einer Ontologie und ist definiert als $IR = \frac{|H|}{|C|}$ mit $|C|$ als die Anzahl der Konzepte und $|H|$ als die Anzahl direkter Subkonzepte. Je mehr direkte Subkonzepte pro Konzept existieren, desto größer wird die IR und umgekehrt. Die Annahme ist, dass Ontologien, die eine spezifische Domäne detailliert beschreiben, eine geringe IR haben, da sie weniger direkte Subkonzepte pro Konzept aufweisen. Wenn ein Konzept weniger direkte Subkonzepte aufweist, dann deutet dies auf eine größere Strukturierung des Domänenwissens hin. Wenn viele Konzepte eine hohe Zahl direkter Subkonzepte aufweisen, dann deutet dies auf einen geringen Detaillierungsgrad des Domänenwissens hin.

Die **attribute richness (AR)** misst die Verteilung von Attributen auf die Konzepte der Ontologie. Die Annahme ist, dass mehr Attribute pro Konzept auf einen höheren Informationsgehalt hinweisen. Die AR ist definiert als $AR = \frac{|att|}{|C|}$ mit $|attr|$ als die Anzahl der Attribute aller Konzepte und $|C|$ als die Anzahl der Konzepte.

Diese Metriken ermöglichen eine Beurteilung der Qualitätskriterien *Computational Efficiency* und *Clarity*. Eine geringe *IR* weist darauf hin, dass eine Ontologie detailliert dargestellt und domänenspezifisch ist. In Kombination mit einer hohen *AR* ist auch der Grad der Clarity besser, da mehr Informationen im Idealfall auch zu mehr Verständlichkeit führen. Eine hohe *RR* kann zu einer geringeren Ausführungsgeschwindigkeit eines Schlussfolgerungssystems führen, da die Relationen zwischen Konzepten komplexer sind als in einer reinen Taxonomie.

In (Vrandečić, 2010, S. 56ff) werden zahlreiche manuelle Methoden und auch Metriken für die Ermittlung der Qualitätsmerkmale *Accuracy, Adaptability, Completeness, Conciseness, Consistency, Computational Efficiency, Organization Fitness* und *Clarity* vorgestellt. Zu beachten ist bei diesen Methoden und Metriken, dass viele davon nur nach einer vorherigen Normalisierung der Ontologie, die ebenfalls von Vrandečić (2010) beschrieben wird, anwendbar sind. Eine Methode, die auch auf nicht normalisierten Ontologien zur (teilweisen) Erhebung des Qualitätskriteriums *Consistency* angewendet werden kann, ist die Methode *Check labels and comments* (Vrandečić, 2010, S. 81):

Check label and comments. Die Konzepte und Relationen einer Ontologie können in mehreren Sprachen vorliegen. Jedes Konzept, jede Relation und jede Instanz kann ein Label tragen, das in einer Ontologie einen Bezeichner zum Zwecke besserer Lesbarkeit darstellt. Die Methode fordert, dass eine Menge relevanter Sprachen für eine Ontologie definiert wird. Anschließend muss überprüft werden, ob alle Label und Kommentare dahingehend gekennzeichnet wurden, welcher Sprache sie zugehörig sind. Danach muss überprüft werden, ob jedes Konzept, jede Instanz und jede Relation einen Kommentar hat, sofern sie einen benötigen. Außerdem muss überprüft werden, ob die Label und Kommentare einer zuvor festgelegten Gestaltungsrichtlinie folgen.

Der Vorteil der Normalisierung ist, dass bei der Entwicklung weiterer Methoden und Metriken Annahmen über die Struktur und vorhandenen Axiome einer Ontologie getroffen werden können. Metriken, die auf einer solchen normalisierten Ontologie entwickelt wurden, besitzen das Potential, auf sehr vielen Ontologien anwendbar sein zu können. Wenn jedoch nicht die Entwicklung weiterer Metriken und Methoden angestrebt wird, sondern nur die Evaluierung als solches, dann stellt die Erfordernis der Normalisierung zum Zwecke der Evaluierung einen zusätzlichen Aufwand dar. Diejenigen, die eine Ontologie evaluieren, müssen dann entscheiden, ob sich dieser Aufwand lohnt.

5.8 Zusammenfassung

Die Ermittlung der Qualität einer Ontologie kann in den folgenden Schritten erfolgen:

1. Auswahl der Evaluierungsdimension (Abschnitt 5.3)
2. Auswahl des Evaluierungsgegenstandes (Abschnitt 5.5)
3. Festlegung auf eine quantitative oder qualitative Qualitätserhebung (Abschnitt 5.2)
4. Auswahl der Qualitätskriterien (Abschnitt 5.6)
5. Auswahl eines Evaluierungsansatzes (Abschnitt 5.4)

Wenn in einem Schritt eine Auswahl getroffen wird, dann schränkt dies auch die weiteren Wahlmöglichkeiten ein. Im ersten Schritt sollte eine Festlegung auf die Dimension der Evaluierung erfolgen. Wenn eine Evaluierung in der Dimension *Benutzbarkeit* erfolgen soll, ist klar, dass nicht der Evaluierungsgegenstand *Semantik* im Schritt 2 ausgewählt werden sollte. Stattdessen sollte eine Auswahl zwischen den Evaluierungsgegenständen *Vokabular*, *Struktur* und *Repräsentation* erfolgen. Da Beurteilungen im Hinblick auf die Benutzbarkeit aber stark von der Zielgruppe abhängig sind, können die zur Auswahl stehenden Evaluierungsgegenstände variieren.

Wenn im zweiten Schritt ein Evaluierungsgegenstand ausgewählt wurde, sollte eine Entscheidung zwischen einer quantativen oder einer qualitativen Qualitätserhebung erfolgen. Wenn im zweiten Schritt mehr als ein Evaluierungsgegenstand ausgewählt

werden kann, dann werden die Schritte 3 bis 5 für jeden Evaluierungsgegenstand durchlaufen. Wenn ein Evaluierungsgegenstand sowohl quantitativ als auch qualitativ bewertet werden soll, dann werden die Schritte 4 bis 5 separat für beide Auswahlen durchlaufen. Wenn beispielsweise der Evaluierungsgegenstand *Vokabular* ausgewählt wurde, dann kann dieser sowohl quantitativ als auch qualitativ bewertet werden. Eine quantitative Bewertung könnte ermitteln, wie viel Prozent der Konzepte Kommentare besitzen. Eine qualitative Bewertung könnte Benutzer danach Fragen, ob die Bezeichnung der Konzepte und Relationen verständlich ist oder nicht.

Im vierten Schritt stehen die Qualitätskriterien zur Auswahl. Bei Wahl einer quantitativen Bewertung des Evaluierungsgegenstandes könnten u.a. die Qualitätskriterien *Computational Efficiency* und *Consistency* zur Auswahl stehen. Bei Wahl einer qualitativen Bewertung könnten u.a. die Qualitätskriterien *Adaptability* und *Clarity* zur Auswahl stehen.

Der vierte und fünfte Schritt sind nicht klar voneinander getrennt. Es ist denkbar, dass in einem konkreten Evaluierungsansatz mehrere Qualitätskriterien erhoben werden können. Es kann jedoch auch sein, dass nur einige Qualitätskriterien mit einem konkreten Ansatz erhoben werden können. Ist dies der Fall, muss zwischen dem vierten und fünften Schritt gewechselt werden, bis alle Qualitätskriterien erhoben wurden.

6 Anwendungsfall

In diesem Kapitel werden einige der vorgestellten Methoden auf das deutsche Grundgesetz angewendet. Aus den Phasen der *Konzept-, Relations-, Instanz-* und *Axiomextraktion* wurde jeweils eine Methode ausgewählt, um den gesamten Ontologieextraktionsprozess zu veranschaulichen. Als Textmining-Architektur wurde die Entwicklerversion von GATE verwendet (General Architectur for Text Engineering). Mit GATE können verschiedene (bereits implementierte) Algorithmen der Textverarbeitung auf einen Textkorpus angewendet, sowie eigene Algorithmen implementiert werden. Die Vorverarbeitung wurde vollständig durch bereits implementierte Algorithmen von GATE bewerkstelligt. Die Reihenfolge, in der die Phasen der Ontologieextraktion ausgeführt wurden, entspricht der Reihenfolge der nachfolgenden Abschnitte.

6.1 Konzeptextraktion

Für die Konzeptextraktion wurde die NC-Value-Methode aus Abschnitt 4.3 gewählt. Damit auch *1-Gramme* als Konzepte erkannt werden können, wurde eine Anpassung der C-Value-Gleichung nach Gelbukh (2009) vorgenommen. Wenn ein kleiner Wert zum Logarithmus hinzuaddiert wird, können dadurch auch Uni-Gramme berücksichtigt werden (Gelbukh, 2009, S. 132f). Tabelle 6.1 zeigt eine Strukturierung der Artikel des Grundgesetzes nach Themenkomplex. Grundsätzlich können verschiedene Ontologien den selben Kontext modellieren. Da die größe der Ontologie eine vollständige Darstellung an dieser Stelle verhindert, wird im Folgenden ein kleiner Ausschnitt der Ontologie betrachtet. Die Themenkomplexe aus Tabelle 6.1 werden deshalb im weiteren Verlauf als Konzepte aufgefasst. Insgesamt wurden mit der NC-Value Methode 2838 Konzepte extrahiert. Um die Genauigkeit der Methode zu erhöhen, wurde der *Snowball-Stemmer*

verwendet. Tabelle 6.2 setzt die Konzepte, die von der NC-Value-Methode erkannten wurden, mit der Auflistung aus Tabelle 6.1 in Beziehung.

Artikel	
	Präambel
$1-19$	Basic Rights
$20-37$	The Federation and the Länder
$38-49$	The Bundestag
$50-53$	The Bundesrat
$53a$	The Joint Committee
$54-61$	The Federal President
$62-69$	The Federal Government
$70-82$	Federal Legislation and Legislative Procedures
$83-91$	The Execution of Federal Laws and the Federal Administration
$91a-91e$	Joint Tasks
$92-104$	The Judiciary
$104a-115$	Finance
$115a-155l$	State of Defence
$116-146$	Transitional and Concluding Provisions

Tabelle 6.1: Themenkomplexe des deutschen Grundgesetzes nach Tumuschat und Kommers (2012)

Bei der Konzeptextraktion zeigt sich, dass Fehler, die zuvor angewandte Algorithmen produzieren, die Ergebnisse der darauf aufbauenden Algorithmen negativ beeinflussen können. Im Zuge der Vorverarbeitung wurde der *DefaultTokeniser* von GATE verwendet. Dieser Tokeniser erzeugt aus einem Wort, das durch einen Bindestrich am Zeilen- oder Seitenende getrennt wurde, kein einzelnes Token, sondern mehrere Tokens. Das hat z.B. zur Folge, dass das Konzept *Joint Committee* in den Varianten *Joint Com-mittee* und *Joint Commit-tee* vorliegt. Im Zuge der weiteren Verarbeitung müssen verschiedene Ausprägungsformen desselben Konzeptes vereint werden, damit Instanzen den selben Konzepten korrekt zugeordnet und Relationen korrekt erkannt werden können.

Ein weiteres Problem stellen falsche POS-Tags dar. Diese können zur Folge haben, dass ein Wort als Konzept angenommen wird, obwohl es, was die Wortart angeht, nicht als solches erkannt werden sollte.

Konzept	Erkannt	Grundform
Präambel	✓	preambl
Basic Rights	✓	basic right
(1) The Federation and (2) the Länder	✓	(1) feder, (2) länder
The Bundestag	✓	bundestag
The Bundesrat	✓	bundesrat
The Joint Committee	✓	joint committe
The Federal President	✓	feder presid
The Federal Government	✓	feder govern
(1) Federal Legislation and (2) Legislative Procedures	✓	(1) feder legisl, (2) legisl procedur
(1) The Execution of Federal Laws and (2) the Federal Administration	✓	(1) feder law, (2) feder administr
Joint Tasks	✓	joint task
The Judiciary	✓	judiciari
Finance	✓	financ
State of Defence	×	
(1) Transitional and (2) Concluding Provisions	(1) ✓ (2) ×	(1) transit provis

Tabelle 6.2: Konzepte, die von der NC-Value-Methode erkannt wurden.

6.2 Relationsextraktion

Für die Relationsextraktion wurde die *Open Information Extraction* nach Banko u. a. (2007), ohne Implementierung der Komponente *Redundancy Based Assessor*, angewendet. Die Aufgabe dieser Komponente wäre es, optionale Modifizierer in Relationen zu entfernen und einen Wert zu ermitteln, der Auskunft darüber gibt, ob eine vom Klassifizierer erkannte Relation als Relation aufgefasst werden sollte oder nicht. Diese Komponente würde die Menge der Relationen reduzieren. In diesem Abschnitt soll jedoch betrachtet werden, was potentiell möglich ist.

Die Anzahl der erkannten assoziativen Relationen beläuft sich auf 5518. Von Interesse sind die Relationen der Konzepte, die in Tabelle 6.2 aufgeführt sind. Die Tabellen 6.3 und 6.4 zeigen eine Teilmenge dieser Relationen.

Konzept (Grundform)	Relation (Grundform)	Konzept (Grundform)
basic right	shall also apply to	person
basic right	shall bind	legislatur
feder	shall requir	consent
feder	shall pay for	grant
feder	shall particip in	discharg
feder	shall have	right
feder	shall have	power
feder	shall grant	compens
feder	shall financ	expenditur
feder	shall establish	forc
feder	shall establish	net-work
feder	shall establish	court
feder	shall discharg	administr
feder	perform	duti
länder	to be financ from	share
länder	shall particip through	bundesrat
länder	shall have	right
länder	shall have	power
länder	shall financ	expenditur
länder	shall bear	cost
länder	shall be entitl to receiv	payment
bundestag	shall requir	major
bundestag	shall have	right
bundestag	shall elect	presid
bundestag	shall determin when	session
bundestag	shall design	member
bundestag	may elect	chancellor

Tabelle 6.3: Relationen der Konzepte Länder, Federation und Bundestag. Die Konzepte und Relationen sind in ihrer Grundform angegeben.

In Tabelle 6.4 ist zu sehen, dass mitunter viele Relationen zwischen Konzepten identifiziert werden, dass aber keineswegs für alle Konzepte Relationen erkannt werden. Es ist auch zu sehen, dass kaum Relationen zwischen den Konzepten aus Tabelle 6.2 bestehen. Darüber hinaus ist die Menge der Relationen, die eine geringe oder keine Aussagekraft besitzen, sehr groß. Dies ist darauf zurück zu führen, dass im Verfahren der *Open Information Extraction* ein Naive Bayes Klassifizierer die Relationen

Konzept (Grundform)	Relation (Grundform)	Konzept (Grundform)
bundesrat	shall specif	task
bundesrat	shall particip in	process
bundesrat	shall have	right
bundesrat	shall select	presid
bundesrat	shall debat	bill
bundesrat	shall be kept inform by	govern
joint committe	shall be inform without	delay
feder presid	-	-
feder govern	-	-
feder legisl	-	-
legisl procedur	-	-
feder law	-	-
feder administr	-	-
joint task	of	construct
judiciari	-	-
financ	within	territori
financ	may introduc	rule
financ	entail or will bring about decreas in	revenu

Tabelle 6.4: Weitere Relationen der Konzepte aus Tabelle 6.2. Die Konzepte und Relationen sind in ihrer Grundform angegeben.

erkennt. Dieser erkennt Relationen auf Grundlage von Wahrscheinlichkeitswerten. Im Konkreten Fall erzielte dieser eine Accuracy $\approx$ 77%, d.h. die Menge „falscher“ Relationen ist relativ hoch. Letzten Endes hätte es am *Redundancy Based Assessor* gelegen, solche Relationen aus der Ergebnismenge zu entfernen.

6.3 Instanzextraktion

Zum Zwecke der Instanzextraktion wurden *Hearst-Pattern* verwendet, wie sie in Abschnitt 4.5 gezeigt wurden. Hier zeigt sich, dass Hearst-Pattern, hinsichtlich der Ergebnismenge, nicht sehr ergiebig sind. Insgesamt wurden für 2838 Konzepte nur 7 Instanzen gefunden. Je nach dem, welcher Algorithmus für die Relationsextraktion eingesetzt wird, kann die Menge der extrahierten Instanzen vergrößert werden. Da im

konkreten Fall die *Open Information Extraction* eingesetzt wurde, konnten bestimmte Nominalphrasen aus OIE-Tripeln der Form (e_1, r, e_2) als Instanzen aufgefasst werden, wenn entsprechende Nominalphrasen von einem zuvor eingesetzten Verfahren der Konzeptextraktion nicht bereits als Konzepte identifiziert wurden. Insgesamt wurden durch dieses Vorgehen zusätzlich 226 Instanzen extrahiert. Die Qualität der gewonnenen Daten ist abhängig vom jeweils verwendeten Verfahren. Da sowohl die *Open Information Extraction* als auch *Hearst-Pattern* auf Nominalphrasen arbeiten, ist die Genauigkeit eines verwendeten *Noun Phrase Chunker* (NP-Chunker) entscheidend. Ein NP-Chunker identifiziert Nominalphrasen in Sätzen. Verwendet wurde der *GATEWrapper*. Unter anderem wurden, in Bezug auf die Konzepte aus Tabelle 6.2, die folgenden Instanzen identifiziert:

Konzept (Grundform)	**Instanz**
basic right	-
feder feder feder	federal chancellor government burdens office
länder länder länder	federal territory territory real property
bundestag bundesrat	petitions committee territory
joint committe	-
feder presid	-
feder govern	-
feder legisl	-
legisl procedur	-
feder law	-
feder administr	-
joint task	-
judiciari	-
financ	-

Tabelle 6.5: Zuordnung von Instanzen zu Konzepten, nach Anwendung der Hearst-Pattern.

Tabelle 6.5 zeigt, dass der größte Teil der Instanzen Konzepten zugehörig ist, die in der Tabelle nicht aufgeführt werden. Auch in der Instanzextraktion gilt, dass die gewonnenen Daten nicht einfach als korrekt angenommen werden können. Sie müssen weiteren Phasen der Überprüfung standhalten, damit Wortarten wie z.B. Pronomen, die als einzelne Nominalphrase auftreten können, nicht als Instanz betrachtet werden. In der Instanzextraktion wird deutlich, dass es Konzepte gibt, die keine Instanzen besitzen. Dies bedeutet, die Ontologie ist *inkohärent* (Zhu u. a., 2013, S. 31).

6.4 Taxonomieerkennung

Taxonomische Beziehungen zwischen Konzepten wurden durch *Hearst Pattern* erkannt, wie sie in Abschnitt 4.4 vorgestellt wurden. Auch hier erzielten die verwendeten Pattern wenig Ergebnisse. Insgesamt wurden 18 taxonomische Beziehungen identifiziert, wie in Abbildung 6.1 zu sehen ist.

Einige dieser Beziehung sind leicht nachvollziehbar, wie z.B., dass *refugee* ein Subkonzept von *person* ist. Andere Beziehungen sind falsch, wie z.B. die Beziehung zwischen *customs* und *foregin country*. Letztere wurde aus Art. 73 Abs. 1 Nr. 5 GG extrahiert. Darin heißt es im ersten Satz:

> *„The Federation shall have exclusive legislative power with respect to: . . . „*

sowie in Absatz 5:

> *„the unity of the customs and trading area, treaties regarding commerce and navigation, the free movement of goods, and the exchange of goods and payments with* **foreign countries, including customs** *and border protection; „*

Der Fettgedruckte Teil des Satzes wurde als taxonomische Relation erkannt. Korrekt wäre jedoch, dass der Zollschutz zum Zahlungs- und Wahrenverkehr mit dem Ausland gehört. Diese Relation wurde jedoch nicht erkannt, da der NP-Chunker *exchange of goods and payments* und *foreign countries* als zwei unterschiedliche Nominalphrasen identifiziert hat.

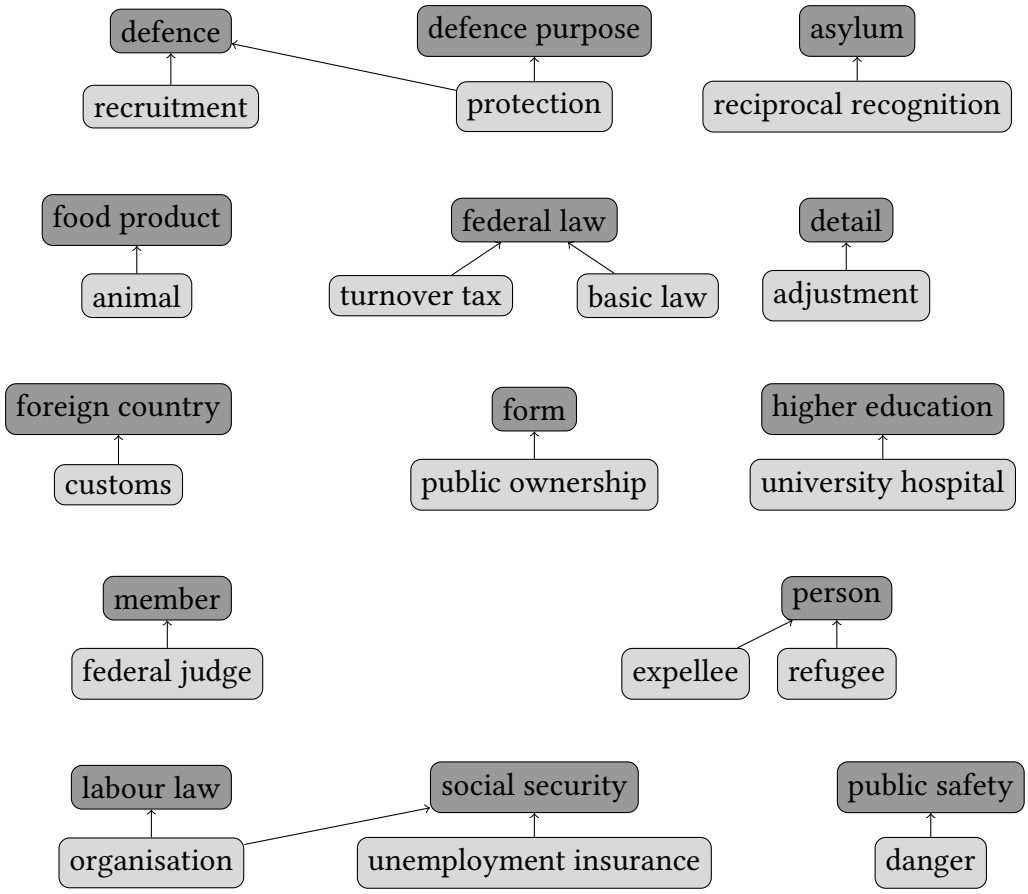

Abbildung 6.1: Extrahierte taxonomische Beziehungen nach Anwendung der *Hearst-Pattern*.

Bei der Taxonomie-Erzeugung muss berücksichtigt werden, dass die Instanzen eines Konzeptes entlang der Hierarchie auch an die oberen Konzepte weitergereicht werden, wenn ein Pfad zu diesen besteht.

6.5 Axiomextraktion

Für die Axiomextraktion wurde das Verfahren des *Association Rule Mining* (ARM) nach Völker u. a. (2015), ohne die Verwendung eines *confidence values*, umgesetzt. Der *confidence value* würde dazu führen, dass evtl. weniger Axiome extrahiert werden, diese jedoch mit größerer Wahrscheinlichkeit gültig sind.

Insgesamt wurden 33239 Axiome extrahiert. Der Großteil davon waren Disjunktheits-Axiome, ein kleinerer Teil Inklusions-Axiome. Zu beachten ist, dass ARM ein Verfahren ist, dass normalerweise auf sehr große Datenbestände angewendet wird. Entscheidend für die Korrektheit der extrahierten Axiome ist die Anzahl der Instanzen eines Konzeptes. Je mehr Instanzen die Konzepte besitzen, desto geringer ist die Wahrscheinlichkeit, dass das Hinzufügen einer weiteren Instanz zu einem Konzept ein Disjunktheits-Axiom ungültig werden lässt. In Bezug auf die Anzahl der extrahierten Instanzen im konkreten Anwendungsfall ist deshalb davon auszugehen, dass das angewendete Verfahren ungeeignet ist, um Disjunktheits-Axiome zu extrahieren. Dies kann sich jedoch ändern, wenn weitere Gesetzestexte in den Textkorpus aufgenommen werden, da dadurch die Anzahl der extrahierten Instanzen evtl. erhöht werden würde.

7 Zusammenfassung & Fazit

Die Extraktion von Ontologien aus natürlichsprachlichen Texten besteht im Wesentlichen aus sechs Phasen. In der Phase der *Vorverarbeitung* wird der Textkorpus für die Anwendung von Algorithmen, die in den anderen Phasen zum Einsatz kommen, vorbereitet. Fehler, die in dieser Phase hinsichtlich der Verarbeitung geschehen, können zu schlechteren Ergebnissen der anderen Algorithmen führen.

Die oftmals erste Phase der Ontologieextraktion, ist die Konzeptextraktion. Es gibt zahlreiche Verfahren, durch die Konzepte extrahiert werden können. Diese unterscheiden sich zum Teil in erheblichem Maße. Die *NC-Value Methode* extrahiert Multi-Wort-Terme aus Textkorpora. Ein Multi-Wort-Term ist ein Term, der aus mehr als einem Wort besteht. Ein Term ist nach Frantzi u. a. (2000) die linguistische Repräsentation eines Konzeptes. Methoden, die auf der *tf-idf* (term frequence inverse document frequency) basieren, gewichten Worte schlechter, wenn diese in vielen Dokumenten auftreten. Wenn ein Wort in wenigen Dokumenten auftritt, wird es dagegen höher gewichtet. Diese *inverse* Gewichtung der Wörter soll die Termerkennung verbessern und ist wegen dieser Eigenschaft nur dann geeignet, wenn, zusätzlich zu domänenspezifischen Texten auch ausreichend Dokumente vorhanden sind, die nicht domänenspezifisch sind. Eine Methode, die darauf aufsetzt, ist die *contrastive weight*.

Die Menge der Verfahren, die für die Relationsextraktion entwickelt wurden, ist sehr hoch. Diese Verfahren unterscheiden sich zum Teil stark voneinander. Um taxonomische Beziehungen zwischen Konzepten zu erkennen, werden häufig *lexico-syntaktische Pattern* eingesetzt. Auch wenn häufig auf die hohe Genauigkeit dieser Pattern verwiesen wird, können diese falsche Ergebnisse erzielen. Dies ist häufig dann der Fall, wenn die Nominalphrasen eines lexico-syntaktischen Pattern einen Bezugspunkt weit außerhalb des Kontext dieses Pattern haben. Dies wurde in Abschnitt 6.4 ersichtlich.

Das Verfahren des *Bootstrapping* kann eingesetzt werden, wenn eine initiale Menge von Relationen zwischen Instanzen/Konzepten vorhanden ist. Durch Bootstrapping kann die Menge der Relationen sehr schnell vergrößert werden. Es ist allerdings nur in sehr spezialisierten Domänen sinnvoll, da ein *semantischer Drift* eintreten kann. *Überwachte Methoden* beruhen auf Methoden des *maschinellen Lernens*. Hier geht es darum, dass von einem Klassifzierer Relationen erkannt werden. Dieser muss zunächst auf einer Menge von Trainingsbeispielen trainiert werden. Trainingsbeispiele entsprechen positiven und negativen Beispielen von Relationen zwischen Konzepten/Instanzen. In der *Open Information Extraction* (OIE) werden diese Beispiele automatisch erzeugt. Verfahren, die Trainingsbeispiele automatisch erzeugen, verwenden häufig Heuristiken, um Trainingsbeispiele in positiv und negativ einzuteilen. Im Verfahren der *Distant Supervision* werden Trainingsbeispiele erzeugt, indem eine externe Datenbank verwendet wird. Wenn zwei Entitäten (Konzepte oder Instanzen) in dieser Datenbank in einer Relation zueinander stehen, wird jeder Satz, in dem die Entitäten auftreten, als ein positives Trainingsbeispiel für diese Relation angenommen.

Instanzen werden typischerweise durch *lexico-syntaktische Pattern* extrahiert. Doch auch eine Kombination von lexico-syntaktischen Pattern mit anderen Methoden, wie z.B. *Bootstrapping*, wurde bereits erfolgreich eingesetzt. Die Verfahren der Taxonomie- und Konzeptextraktion ergänzen sich gegenseitig, wenn für eines der Verfahren lexico-syntaktische Pattern eingesetzt werden. Wenn zunächst die Konzeptextraktion angewendet wird, können in der Instanzextraktion die Nominalphrasen, die kein bereits bekanntes Konzept darstellen, als Instanzen aufgefasst werden. Wenn zunächst die Instanzextraktion stattfindet, können in der Konzeptextraktion die Nominalphrasen, die nicht bereits als Instanz erkannt wurden, als Konzept aufgefasst werden. Wenn Instanzen extrahiert und Konzepten zugeordnet wurden, und Relationen zwischen Konzepten und Instanzen extrahiert wurden, ist die *ABox* der Ontologie erzeugt (siehe Abschnitt 2.4).

Auf dem Gebiet der Axiomextraktion konnten nur wenige Arbeiten ermittelt werden. Diese Arbeiten konzentrierten sich auf die Extraktion von Disjunktheits- und Subsumptions-Axiomen (d.h. Inklusions-Axiomen). Ein sehr schnell zu implementierender Ansatz zur Extraktion von Disjunktheits-Axiomen, ist der Ansatz des *Association*

Rule Mining (ARM) nach Völker u. a. (2015). Dieser Ansatz sollte jedoch nur auf sehr große Datenbestände angewendet werden, da er die *closed world assumption* umsetzt. Dieser Ansatz ermöglicht auch die Extraktion von Inklusions-Axiomen. Der Ansatz von Zhu u. a. (2015) verwendet zur Extraktion von Subsumptions- und Disjunktheits-Axiomen Bayessche Netze. So wie bei ARM muss auch hier die ABox einer Ontologie bereits vorliegen. Im Gegensatz zu ARM liegt diesem Ansatz die *open world assumption* zugrunde, weshalb dieser Ansatz zur Anwendung auf das World Wide Web geeignet ist. Ein weiterer Ansatz zur Extraktion von Disjunktheits- und Subsumptions-Axiomen ist im Bereich des maschinellen Lernens zu finden. In Völker u. a. (2015) wird neben ARM auch dargestellt, wie ein Klassifizierer zum Erlernen dieser Axiome trainiert werden kann.

Für die Phasen der Konzept-, Relations-, Instanz- und Axiomextraktion ist die Menge der zur Verfügung stehenden Dokumente und der Stil der Sprache entscheidend. Mit letzterem ist gemeint, dass sich der oft zeitaufwändige Entwurf von lexico-syntaktische Pattern erst dann lohnt, wenn bestimmte Formulierungen in den Dokumenten sehr häufig auftreten. Dies ist beispielsweise häufig in Gesetzestexten der Fall. Für solche Domänen lohnt sich der Entwurf solcher Pattern wahrscheinlich eher, als für Domänen, in denen die Formulierung bestimmter Sachverhalte vom Schreibstil der Autoren abhängig ist. Die Menge der Dokumente ist insbesondere für Verfahren wichtig, die auf ARM beruhen, sowie auf Verfahren, die auf überwachtem Lernen beruhen.

Nach der Ontologieextraktion muss diese *validiert* und *verifiziert* werden. Nach Gómez-Pérez (2004) ist die Validierung die Überprüfung, ob die erzeugte Ontologie ein korrektes Modell des modellierten Sachverhalts darstellt und die Verfikation betrifft die Überprüfung, ob die Axiome und andere Definitionen in der Ontologie korrekt sind. Es gibt verschiedene Evaluierungsansätze. In qualitativen Ansätzen wird die Validierung bzw. Verifizierung einer Ontologie manuell durch Menschen durchgeführt. Quantitative Ansätze beurteilen die Qualität einer Ontologie in Bezug auf messbare Kriterien. Die *Dimension* der Evaluierung betrifft die Evaluierung bestimmter Eigenschaften der Ontologie. Neben der formalen Korrektheit kann z.B. auch die Benutzbarkeit evaluiert werden. Innerhalb von einer Dimension kann ein konkreter *Evaluierungsgegenstand* evaluiert werden. Zu den konkreten Evaluierungsgegenständen zählen z.B. das Voka-

bular (Konzepte und Relationen) und die Syntax einer Ontologie. In Bezug auf diese Evaluierungsgegenstände kann dann die *Qualität* ermittelt werden. Zu den Qualitätskriterien zählen z.B. die Genauigkeit und die Anwendbarkeit. Nach Vrandečić (2010) gibt die Genauigkeit an, in wie weit Domänenexperten den Axiomen einer Ontologie zustimmen. Die Anwendbarkeit ist ein Qualitätskriterium, das angibt, in wie weit eine Ontologie auf einen Verwendungszweck anpassbar ist. Konkrete Evaluierungsansätze können in *benutzer-, anwendungs-, aufgaben-* und *datenzentrierte Ansätze* unterschieden werden. Die Auswahl eines Evaluierungsansatzes kann beispielsweise entlang der beschriebenen Vorgehensweise in Kapitel 5.8 erfolgen.

Zum Abschluss kann gesagt werden, dass sich der Gesamtprozess der Ontologieextraktion aus natürlichsprachlichen Texten aus der Konzept-, Relations-, Instanz- und Axiom-Extraktion zusammensetzt. Die meisten Arbeiten auf diesem Gebiet behandeln ein einziges dieser Gebiete. Nur wenige Arbeiten behandelten bisher den gesamten Prozess. Das Ergebnis einer dieser Arbeiten ist der *Ontology Learning Layercake* (OLC). Dieser trennt die einzelnen Phasen jedoch zu stark voneinander. Im OLC wird nicht abgebildet, wie die Ergebnisse einer Phase die Ergebnisse einer anderen Phase positiv beeinflussen können. Wie in Kapitel 6 gezeigt wurde, kann ein Verfahren der Instanzextraktion ein Verfahren der Konzeptextraktion ergänzen - und umgekehrt. Außerdem wird die Instanzextraktion im OLC nicht abgebildet.

Die Ontologieextraktion aus natürlichsprachlichen Texten auf vollautomatischem Wege ist noch nicht realisierbar, wie der geschilderte Anwendungsfall aus Kapitel 6 verdeutlicht. Aus diesem Grund sollten Ontologieextraktionssysteme den Benutzer in den Extraktionsprozess einbinden, sobald eine verwendete Methode dies zulässt.

Literaturverzeichnis

[Ahrenberg 2009] Ahrenberg, Lars: *Term extraction: A Review Draft Version 091221.* 2009. – Eingesehen am 14.05.2016 um 12:19 Uhr

[Angeli u. a. 2015] Angeli, Gabor ; Johnson Premkumar, Melvin J. ; Manning, Christopher D.: Leveraging Linguistic Structure For Open Domain Information Extraction. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers).* Beijing, China : Association for Computational Linguistics, July 2015, S. 344–354. – URL http://www.aclweb.org/anthology/P15-1034

[Bahns 1996] Bahns, J.: *Kollokationen als lexikographisches Problem: Eine Analyse allgemeiner und spezieller Lernerwörterbücher des Englischen.* De Gruyter, 1996 (Lexicographica. Series Maior). – URL https://books.google.de/books?id=-Uj1V1vpnKEC. – ISBN 9783110937930

[Banko u. a. 2007] Banko, Michele ; Cafarella, Michael J. ; Soderland, Stephen ; Broadhead, Matthew ; Etzioni, Oren: Open Information Extraction from the Web. In: *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI-07)*, 2007, S. 2670–2676

[Basili u. a. 2001] Basili, Roberto ; Moschitti, Alessandro ; Pazienza, Maria T. ; Zanzotto, Fabio M.: A constrastive approach to term extraction. In: *TTA* (2001). – URL http://disi.unitn.it/moschitti/articles/TIA2000.pdf. – Eingesehen am 19.05.2016, 17:08 Uhr

[Bergmann 2011] Bergmann, Isumo: *Patentverletzungen in der Biotechnologie: Einsatz semantischer Patentanalysen.* Kap. Patentverletzungen, S. 6–53.

Wiesbaden : Gabler, 2011. – URL http://dx.doi.org/10.1007/978-3-8349-6681-0_2. – ISBN 978-3-8349-6681-0

[Bodendorf 2013] Bodendorf, F.: *Daten- und Wissensmanagement.* Springer Berlin Heidelberg, 2013 (Springer-Lehrbuch). – URL https://books.google.de/books?id=VVgmBgAAQBAJ. – ISBN 9783662064948

[Borgelt und Kruse 2002] Borgelt, Christian ; Kruse, Rudolf: *Induction of Association Rules: Apriori Implementation.* S. 395–400. In: Härdle, Wolfgang (Hrsg.) ; Rönz, Bernd (Hrsg.): *Compstat: Proceedings in Computational Statistics.* Heidelberg : Physica-Verlag HD, 2002

[Brewster u. a. 2004] Brewster, Christopher ; Alani, Harith ; Dasmahapatra, Srinandan ; Wilks, Yorick: *Data Driven Ontology Evaluation.* 2004. – URL http://www.cbrewster.com/papers/BrewsterLREC.pdf. – Zugriffszeit: 02.07.2016, 12:23 Uhr

[Chen u. a. 2016] Chen, Zhe ; Cafarella, Michael ; Jagadish, H. V.: Long-tail Vocabulary Dictionary Extraction from the Web. In: *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, ACM, 2016 (WSDM '16), S. 625–634. – URL http://doi.acm.org/10.1145/2835776.2835778. – ISBN 978-1-4503-3716-8

[Cimiano 2014] Cimiano, Philipp: *Perspectives On Ontology Learning.* Kap. Foreword, IOS Press, 2014

[Cimiano u. a. 2009] Cimiano, Philipp ; Mädche, Alexander ; Staab, Steffen ; Völker, Johanna: *Handbook on Ontologies.* Kap. Ontology Learning, S. 245–267. Berlin, Heidelberg : Springer Berlin Heidelberg, 2009. – URL http://dx.doi.org/10.1007/978-3-540-92673-3_11. – ISBN 978-3-540-92673-3

[Clarke 2001] Clarke, Stella G. D.: *Relationships in the Organization of Knowledge.* Kap. Thesaural Relationships, S. 37–52. Dordrecht : Springer Netherlands, 2001. – URL http://dx.doi.org/10.1007/978-94-015-9696-1_3. – ISBN 978-94-015-9696-1

[Dengel 2012] Dengel, Andreas: *Semantische Technologien: Grundlagen – Konzepte – Anwendungen*. Kap. Semantische Netze, Thesauri und Topic Maps, S. 73–107. Heidelberg : Spektrum Akademischer Verlag, 2012. – URL http://dx.doi.org/10.1007/978-3-8274-2664-2_3. – ISBN 978-3-8274-2664-2

[Foo 2012] Foo, Jody: *Computational Terminology: Exploring Bilingual and Monolingual Term Extraction*. SE-58183 Linköping, Schweden, Linköping University, Dissertation, 2012

[Frantzi u. a. 2000] Frantzi, Katerina ; Ananiadou, Sophia ; Mima, Hideki: Automatic recognition of multi-word terms: the C-value/NC-value method. In: *International Journal on Digital Libraries* 3 (2000), Nr. 2, S. 115–130. – URL http://dx.doi.org/10.1007/s007999900023. – ISSN 1432-5012

[Fudholi u. a. 2016] Fudholi, D. H. ; Rahayu, W. ; Pardede, E.: Ontology-Based Information Extraction for Knowledge Enrichment and Validation. In: *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, March 2016, S. 1116–1123. – ISSN 1550-445X

[Gangemi u. a. 2005] Gangemi, Aldo ; Catenacci, Carola ; Ciaramita, Massimiliano ; Lehmann, Jos: A theoretical framework for ontology evaluation and validation. In: *SWAP* Bd. 166 Citeseer (Veranst.), 2005

[Gelbukh 2009] Gelbukh, A.: *Computational Linguistics and Intelligent Text Processing: 10th International Conference, CICLing 2009, Mexico City, Mexico, March 1-7, 2009, Proceedings*. Springer Berlin Heidelberg, 2009 (Lecture Notes in Computer Science). – URL https://books.google.de/books?id=N8FqCQAAQBAJ. – ISBN 9783642003820

[Gómez-Pérez 2001] Gómez-Pérez, Asunción: Evaluation of ontologies. In: *International Journal of Intelligent Systems* 16 (2001), Nr. 3, S. 391–409. – URL http://dx.doi.org/10.1002/1098-111X(200103)16:3<391::AID-INT1014>3.0.CO;2-2

[Gómez-Pérez 2004] Gómez-Pérez, Asunción: *Ontology Evaluation.* S. 251–273. In: Staab, Steffen (Hrsg.) ; Studer, Rudi (Hrsg.): *Handbook on Ontologies.* Berlin, Heidelberg : Springer Berlin Heidelberg, 2004

[Hitzler u. a. 2012] Hitzler, Pascal ; Krötzsch, Markus ; Parsia, Bijan ; Patel-Schneider, Peter F. ; Rudolph, Sebastian: OWL 2 Web Ontology Language Primer (Second Edition) / W3C. URL http://www.w3.org/TR/owl2-primer/, 2012. – Forschungsbericht

[Hitzler u. a. 2008] Hitzler, Pascal ; Krötzsch, Markus ; Rudolph, Sebastian ; Sure, York: *Semantic Web.* Springer Berlin Heidelberg, 2008

[Hoffmann 2013] Hoffmann, Dirk W.: *Grenzen der Mathematik - Eine Reise durch die Kerngebiete der mathematischen Logik.* Springer Spektrum, 2013

[Jakus u. a. 2013] Jakus, Grega ; Milutinović, Veljko ; Omerović, Sanida ; Tomažič, Sašo: *Concepts, Ontologies, and Knowledge Representation.* Springer, 2013

[Juršič u. a. 2007] Juršič, Matjaž ; Mozetič, Igor ; Lavrač, Nada: Learning Ripple Down Rules for Efficient Lemmatization. In: Mladenić, Dunja (Hrsg.) ; Grobelnik, Marko (Hrsg.): *Proceedings of the 10th International Multiconference Information Society.* Ljubljana, Slovenia : IJS, Oktober 2007, S. 206–209

[Kageura und Umino 1996] Kageura, Kyo ; Umino, Bin: *Methods of Automatic Term Recognition - A Review.* Bd. 3. S. 259 – 289. In: *Terminology* Bd. 3, Academic Press, 1996

[Konstantinova 2014] Konstantinova, Natalia: *Analysis of Images, Social Networks and Texts: Third International Conference, AIST 2014, Yekaterinburg, Russia, April 10-12, 2014, Revised Selected Papers.* Kap. Review of Relation Extraction Methods: What Is New Out There?, S. 15–28. Cham : Springer International Publishing, 2014. – URL http://dx.doi.org/10.1007/978-3-319-12580-0_2. – ISBN 978-3-319-12580-0

[Kozareva 2012] Kozareva, Zornitsa: Cause-Effect Relation Learning. In: *Workshop Proceedings of TextGraphs-7: Graph-based Methods for Natural Language Processing.*

Jeju, Republic of Korea : Association for Computational Linguistics, July 2012, S. 39–43. – URL http://www.aclweb.org/anthology/W12-4107

[Krötzsch u. a. 2014] Krötzsch, Markus ; Simančík, František ; Horrocks, Ian: *Perspectives On Ontology Learning*. Kap. A Description Logic Primer, IOS Press, 2014

[Lehmann und Völker 2014] Lehmann, Jens ; Völker, Johanna: *Perspectives On Ontology Learning*. Kap. An Introduction to Ontology Learning, IOS Press, 2014

[Lim u. a. 2013] Lim, Edward H. ; Liu, James N. ; Lee, Raymond S.: *Knowledge Seeker - Ontology Modelling for Information Search and Management*. Springer, 2013

[MacCartney] MacCartney, Bill: *Relation Extraction*. – URL https://web.stanford.edu/class/cs224u/materials/cs224u-2015-relation-extraction.pdf. – Präsentationsfolien. Eingesehen am 25.05.2016, 19:40 Uhr

[Manning und Schütze 2000] Manning, Christoph D. ; Schütze, Hinrich: *Foundations of Statistical Natural Language Processing*. The MIT Press, 2000

[Maynard und Bontcheva 2014] Maynard, Diana ; Bontcheva, Kalina: *Perspectives On Ontology Learning*. Kap. Natural Language Processing, IOS Press, 2014

[Miner u. a. 2012a] Miner, Gary ; Delen, Dursun ; Elder, John ; Fast, Andrew ; Hill, Thomas ; Nisbet, Robert A.: Chapter 2 - The Seven Practice Areas of Text Analytics. In: Miner, Gary (Hrsg.) ; Delen, Dursun (Hrsg.) ; Elder, John (Hrsg.) ; Fast, Andrew (Hrsg.) ; Hill, Thomas (Hrsg.) ; Nisbet, Robert A. (Hrsg.): *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. Boston : Academic Press, 2012, S. 29 – 41. – URL http://www.sciencedirect.com/science/article/pii/B9780123869791000025. – ISBN 978-0-12-386979-1

[Miner u. a. 2012b] Miner, Gary ; Delen, Dursun ; Elder, John ; Fast, Andrew ; Hill, Thomas ; Nisbet, Robert A.: Chapter 3 - Conceptual Foundations of Text Mining and Preprocessing Steps. In: Miner, Gary (Hrsg.) ; Delen, Dursun (Hrsg.) ; Elder, John (Hrsg.) ; Fast, Andrew (Hrsg.) ; Hill, Thomas

(Hrsg.) ; Nisbet, Robert A. (Hrsg.): *Practical Text Mining and Statistical Analysis for Non-structured Text Data Applications*. Boston : Academic Press, 2012, S. 43 – 51. – URL http://www.sciencedirect.com/science/article/pii/B9780123869791000037. – ISBN 978-0-12-386979-1

[Mintz u. a. 2009] Mintz, Mike ; Bills, Steven ; Snow, Rion ; Jurafsky, Dan: Distant Supervision for Relation Extraction Without Labeled Data. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, Association for Computational Linguistics, 2009 (ACL '09), S. 1003–1011. – URL http://dl.acm.org/citation.cfm?id=1690219.1690287

[Neuhaus u. a. 2013] Neuhaus, Fabian ; Vizedom, Amanda ; Baclawski, Ken ; Bennett, Mike ; Dean, Mike ; Denny, Michael ; Grüninger, Michael ; Hashemi, Ali ; Longstreth, Terry ; Obrst, Leo ; Ray, Steve ; Sriram, Ram ; Schneider, Todd ; Vegetti, Marcela ; West, Matthew ; Yim, Peter: Towards Ontology Evaluation Across the Life Cycle: The Ontology Summit 2013. In: *Appl. Ontology* 8 (2013), Juli, Nr. 3, S. 179–194. – URL http://dl.acm.org/citation.cfm?id=2594763.2594765

[Peters und Weller 2008] Peters, Isabella ; Weller, Katrin: Paradigmatic and Syntagmatic Relations in Knowledge Organization Systems. In: *Information - Wissenschaft & Praxis* 59 (2008), Nr. 2, S. 100–107

[Polovina 2007] Polovina, Simon: *Conceptual Structures: Knowledge Architectures for Smart Applications: 15th International Conference on Conceptual Structures, ICCS 2007, Sheffield, UK, July 22-27, 2007. Proceedings*. Kap. An Introduction to Conceptual Graphs, S. 1–14. Berlin, Heidelberg : Springer Berlin Heidelberg, 2007. – URL http://dx.doi.org/10.1007/978-3-540-73681-3_1. – ISBN 978-3-540-73681-3

[Porzel und Malaka 2004] Porzel, R. ; Malaka, R.: A task-based approach for ontology evaluation. In: *Proc. of ECAI 2004 Workshop on Ontology Learning and Population*, URL http://http://olp.dfki.de/ecai04/final-porzel.pdf, 2004. – Zugriffszeit 02.07.2016, 15:31 Uhr

[Poveda-Villalón u. a. 2012] Poveda-Villalón, María ; Suárez-Figueroa, Mari C. ; Gómez-Pérez, Asunción: *Validating Ontologies with OOPS!* S. 267–281. In: Teije, Annette ten (Hrsg.) ; Völker, Johanna (Hrsg.) ; Handschuh, Siegfried (Hrsg.) ; Stuckenschmidt, Heiner (Hrsg.) ; d'Acquin, Mathieu (Hrsg.) ; Nikolov, Andriy (Hrsg.) ; Aussenac-Gilles, Nathalie (Hrsg.) ; Hernandez, Nathalie (Hrsg.): *Knowledge Engineering and Knowledge Management: 18th International Conference, EKAW 2012, Galway City, Ireland, October 8-12, 2012. Proceedings.* Berlin, Heidelberg : Springer Berlin Heidelberg, 2012

[Powers 2015] Powers, David M. W.: What the F-measure doesn't measure: Features, Flaws, Fallacies and Fixes. In: *CoRR* abs/1503.06410 (2015). – URL http://arxiv.org/abs/1503.06410

[Sasaki 2007] Sasaki, Yutaka: The truth of the F-measure / University of Manchester. URL http://www.toyota-ti.ac.jp/Lab/Denshi/COIN/people/yutaka.sasaki/F-measure-YS-26Oct07.pdf, 2007. – Forschungsbericht. Eingesehen am 13.05.2016 um 18:20 Uhr

[Schoening 2015] Schoening, James R.: *SUMO Ontologie.* 2015. – URL http://ontolog.cim3.net/file/resource/historic-archives/IEEE-SUO-WG/IEEE-SUO-WG_acct-from-JmSchoening_20151013b.txt. – IEEE SUO WG, archivierte E-Mail. Eingesehen am 13.04.2016, 15:50 Uhr

[Schubert 2009] Schubert, Matthias: *Mathematik für Informatiker - Ausführlich erklärt mit vielen Programmbeispielen und Aufgaben.* Vieweg + Teubner, 2009

[Senellart und Blondel 2007] Senellart, Pierre ; Blondel, Vincent D.: *Survey of Text Mining: Clustering, Classification, and Retrieval, Second Edition.* Kap. Automatic Discovery of Similar Words, S. 25–44, Springer, 2007

[Smith 2008] Smith, Barry: The evaluation of ontologies: editorial review vs. democratic ranking. In: *Proceedings of InterOntology* (2008), S. 127–128

[Sowa 2015] Sowa, John F.: *Klassifikation semantischer Netze.* 2015. – URL https://www.jfsowa.com/pubs/semnet.htm. – Eingesehen am 20.04.2016 um 16:03 Uhr

[Stock 2009] Stock, Wolfgang G.: Begriffe und semantische Relationen in der Wissensrepräsentation. In: *Information - Wissenschaft & Praxis* 60 (2009), Nr. 8, S. 403–420

[Supekar 2005] Supekar, K.: A peer-review approach for ontology evaluation. In: *8th Int. Protégé Conference, Madrid, Spain*, 2005

[Tartir u. a. 2010] Tartir, Samir ; Arpinar, I. B. ; Sheth, Amit P.: *Ontological Evaluation and Validation.* S. 115–130. In: Poli, Roberto (Hrsg.) ; Healy, Michael (Hrsg.) ; Kameas, Achilles (Hrsg.): *Theory and Applications of Ontology: Computer Applications.* Dordrecht : Springer Netherlands, 2010

[Tumuschat und Kommers 2012] Tumuschat, Christian ; Kommers, Donald P.: *Gesetze im Internet: Das Grundgesetz.* 2012. – URL https://www.gesetze-im-internet.de/englisch_gg/. – Zugriffszeit: 20.08.2016, 20:50 Uhr

[Völker u. a. 2015] Völker, Johanna ; Fleischhacker, Daniel ; Stuckenschmidt, Heiner: Automatic Acquisition of Class Disjointness. In: *Journal of Web Semantics* 35 (2015), Dezember, Nr. P2, S. 124–139

[Vrandečić 2010] Vrandečić, Zdenko: *Ontology Evaluation*, Karlsruher Institut für Technologie, Dissertation, 2010

[Wang und Cohen 2009] Wang, Richard C. ; Cohen, William W.: Automatic Set Instance Extraction Using the Web. In: *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1 - Volume 1.* Stroudsburg, PA, USA : Association for Computational Linguistics, 2009 (ACL '09), S. 441–449. – URL http://dl.acm.org/citation.cfm?id=1687878.1687941. – ISBN 978-1-932432-45-9

[Witte und Mülle 2006] Witte, Rene ; Mülle, Jutta: *Text Mining: Wissensgewinnung aus natürlichsprachlichen Dokumenten.* Universität Karlsruhe (TH), 2006

[Xiang u. a. 2015] Xiang, Yang ; Wang, Xiaolong ; Zhang, Yaoyun ; Qin, Yang ; Fan, Shixi: *Neural Information Processing: 22nd International Conference, ICONIP 2015, Istanbul, Turkey, November 9-12, 2015, Proceedings, Part II.* Kap. Distant Supervision for Relation Extraction via Group Selection, S. 250–258. Cham : Springer International Publishing, 2015. – URL http://dx.doi.org/10.1007/978-3-319-26535-3_29. – ISBN 978-3-319-26535-3

[Yu u. a. 2007] Yu, Jonathan ; Thom, James A. ; Tam, Audrey: Ontology evaluation: using Wikipedia categories for browsing, in. In: *Proceedings of the 16th Conference on Information and Knowledge Management, ACM*, Press, 2007, S. 223–232

[Zhu u. a. 2013] Zhu, Man ; Gao, Zhiqiang ; Pan, Jeff Z. ; Zhao, Yuting ; Xu, Ying ; Quan, Zhibin: Ontology Learning from Incomplete Semantic Web Data by BelNet. In: *Proceedings of the 2013 IEEE 25th International Conference on Tools with Artificial Intelligence.* Washington, DC, USA : IEEE Computer Society, 2013 (ICTAI '13), S. 761–768. – URL http://dx.doi.org/10.1109/ICTAI.2013.117. – ISBN 978-1-4799-2971-9

[Zhu u. a. 2015] Zhu, Man ; Gao, Zhiqiang ; Pan, Jeff Z. ; Zhao, Yuting ; Xu, Ying ; Quan, Zhibin: TBox Learning from Incomplete Data by Inference in BelNet+. In: *Know.-Based Syst.* 75 (2015), Februar, Nr. C, S. 30–40. – URL http://dx.doi.org/10.1016/j.knosys.2014.11.004. – ISSN 0950-7051

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe.

Hamburg, 31. August 2016 Francis Opoku