



Studienarbeit

Methoden zur automatischen Erstellung von Term-Hierarchien aus Phänotypendaten

Christoph Böhm

21. Dezember 2007

Betreuer: M.Sc. Philip Groth und Prof. Dr. Ulf Leser
Lehrstuhl für Wissensmanagement in der Bioinformatik

Institut für Informatik
Mathematisch-Naturwissenschaftliche Fakultät II

Inhaltsverzeichnis

1	Einleitung	3
1.1	Phänotypen	3
1.2	Ontologien	3
1.3	Methoden zur Hierarchisierung von Termen	4
1.4	Zielsetzung	5
1.5	Aufbau der Arbeit	6
1.6	Notationen	6
2	Term-Hierarchisierung mittels Term-Cluster-Verfahren	7
2.1	Gemeinsames Auftreten (cooccurrence) als Grundlage der Cluster	7
2.1.1	<i>DF</i> -Vektoren nach Salton	7
2.1.2	Plain <i>DF</i> nach Forsyth und Rada	8
2.1.3	Subsumption nach Sanderson und Croft	8
2.1.4	Context-based subsumption (CBS) nach Lawrie et al.	10
2.1.5	CAARD nach Krishna und Krishnapuram	12
2.1.6	Asymmetrisches gemeinsames Auftreten nach Chen et al.	13
2.2	Kontext von Termen als Grundlage der Cluster	14
2.2.1	Kontext-Vektoren nach Gauch und Wang	14
2.2.2	Reduktion der Term-Term-Matrix nach Schütze und Pedersen	14
3	Term-Hierarchisierung mittels Dokument-Cluster-Verfahren	16
4	Term-Hierarchisierung mit Hilfe linguistischer Merkmale	19
4.1	Syntaktische Muster	19
4.2	<i>head</i> und <i>modifier</i>	20
4.3	Logik und linguistische Merkmale	21
5	Weitere Ansätze zur Term-Hierarchisierung	22
6	Evaluation von Term-Hierarchien	23
7	Ergebnisse der Subsumption nach Sanderson und Croft	24
7.1	Technische Details	24
7.2	Ergebnisse	25
7.3	Bewertung und Ausblick	27
8	Zusammenfassung	28
	Literatur	29

1 Einleitung

In dieser Studienarbeit werden Methoden des *Information Retrieval* beschrieben, mit denen Konzepte auf automatischem Wege hierarchisch angeordnet werden können. Die Konzepte werden durch Terme oder Phrasen aus biomedizinischen Daten über Phänotypen repräsentiert. Es handelt sich demzufolge um Phänotyp-spezifische Konzepte, die zu hierarchisieren sind.

Experimente werden zeigen, dass sich mit einfachen Methoden bereits eine gute *Precision* erreichen lässt.

1.1 Phänotypen

Unter einem Phänotyp versteht man das Erscheinungsbild oder die Manifestation einer Menge von Eigenschaften eines Individuums als Resultat eines Genotyps sowie dessen Interaktion mit der Umwelt [Smith et al., 2005]. Der Genotyp ist die individuelle Zusammensetzung der Erbinformationen eines Organismus - die spezifische Zusammenstellung der DNA. Biologen gewinnen laufend neue Informationen über Phänotypen, u.a. aus Hochdurchsatz-Experimenten, wie Microarrays [Bier et al., 2007] oder Gen-Knockout Studien [Walinski, 2004]. Diese Informationen sind meist als Abweichung vom 'Wildtyp' und für ein bestimmtes Merkmal oder einen beschränkten Merkmalsbereich beschrieben. Solche Ergebnisse werden in der Regel in Journal-Artikeln oder Konferenz-Beiträgen veröffentlicht und sind häufig zusätzlich in verschiedenen spezialisierten Datenbanken wiederzufinden.

*PhenomicDB*¹ [Groth et al., 2007] ist eine integrierte organismenübergreifende Datenbank, die Phänotypen zzgl. deren assoziierter Gene, Genindizes und Orthologiebeziehungen enthält. Weiterhin sind Genannotationen mit Konzepten aus der *Gene Ontology (GO)*² [GO-Consortium, 2006] enthalten. Die in *PhenomicDB* enthaltenen Phänotypen wurden aus primär-Datenbanken extrahiert und sind größten Teils in Publikationen beschrieben. Diese natürlichsprachlichen Darstellungen haben die Nachteile, dass sie dem Leser Interpretationsspielraum lassen sowie wegen ihrer Unstrukturiertheit rechentechnisch schwer zu verarbeiten sind.

Jegliche textuelle Beschreibung kann präzise formuliert werden, wenn ein kontrolliertes Vokabular zur Verfügung steht. Das automatische Prozessieren von Texten kann damit vereinfacht werden. Eine solche "präzisere" Publikation ist zwar lediglich semistrukturiert, jedoch geben Glossare oder Thesauri dann Auskunft über die genaue Bedeutung von Termen und stellen Verwandtschaften, z.B. synonyme Bedeutung, klar. Insbesondere ist die Position eines Terms in einer Konzept-Hierarchie von Interesse. Dadurch lassen sich Aussagen über die Generalität von Termen und Texten treffen. Diese sind z.B. für das Clustern von Dokumenten von Interesse.

1.2 Ontologien

Eine Struktur, in der Konzepte hierarchisch entsprechend ihrer Generalität angeordnet sind, kann als Ontologie verstanden werden. Laut *WordNet*³ ist ein Konzept eine abstrakte Idee, abgeleitet aus konkreten Instanzen der realen Welt - in unserem Fall Phänotypen.

Ontologien werden in den verschiedensten Bereichen verwendet: als gemeinsames Vokabular einer Domäne [GO-Consortium, 2006]; zur Erweiterung von Anfragen [Necib und Freytag, 2005] oder zur Anreicherung des WWW mit Semantik [McGuinness und van Harmelen, 2004].

¹PhenomicDB ist öffentlich verfügbar: www.phenomicdb.de

²Die *Gene Ontology* ist ein gemeinschaftlich erstelltes strukturiertes Vokabular zur Annotation von Genen und deren Produkten. Sie ist ebenfalls öffentlich verfügbar: www.geneontology.org

³*WordNet* ist eine Ontologie der englischen Sprache. Siehe: wordnet.princeton.edu

Aufgrund dieser unterschiedlichen Anwendungsgebiete in diversen Disziplinen gibt es keine einheitliche Definition.

In der Informatik hat sich folgende Definition durchgesetzt: Eine Ontologie ist eine explizite, formale Spezifikation von gemeinsamen Konzepten einer Domäne [Staab und Studer, 2004]. Diese Definition ist für unseren Anwendungsfall zu streng. Wir verstehen im Folgenden eine Ontologie als eine Menge von Konzepten (d.h. Terme oder Phrasen), die ggf. durch *ist-ein*-Beziehungen (engl. *is-a*) in Verbindung stehen. Die *ist-ein*-Beziehungen ist eine für zwei Konzepte A und B definierte Relation $ist-ein(A,B)$, wobei A ein spezielleres Konzept als B ist. Weitere Arten von Beziehungen werden hier nicht behandelt. Diese Einschränkung kann getroffen werden, da unser Ergebnis einfach strukturiert und somit gut verständlich sein soll.

Die *Mammalian Phenotype Ontology (MPO)* [Smith et al., 2005] ist ein Beispiel einer Ontologie für Phänotypen. Sie beinhaltet ein begrenztes Vokabular für Phänotypen von Säugetieren und wurde ausschließlich von Hand erstellt. Dies hat eine hohe Qualität zur Folge, erfordert jedoch einen enormen Arbeitsaufwand. Manuelles Vorgehen ist daher ab einer gewissen Anzahl an Konzepten nicht mehr zu vertreten. Die Konzepte der MPO mit ihren Beziehungen werden in Kapitel 7 zur Evaluation von Experimenten herangezogen.

Die Phänotypen in *PhenomicDB* enthalten mehrere Tausend Phänotyp-spezifische Konzepte. Die dieser Arbeit übergeordnete Vision ist es, aus der speziesübergreifenden Datenbank *PhenomicDB* auf automatischem Wege eine Hierarchie von Phänotyp-Konzepten zu erstellen.

1.3 Methoden zur Hierarchisierung von Termen

Diese Arbeit stellt Verfahren zusammen, mit denen gegebene Termlisten strukturiert werden können. Das Erstellen der Termlisten wird hier nicht Thema sein. Methoden zur Hierarchisierung von Termen sind in der Literatur auf unterschiedliche Weise benannt: *ontology mining*, *thesaurus construction*, *taxonomy exploration* oder *hierarchy learning*. Die vorzufindenden Ansätze haben meist eines von zwei Zielen: sie sollen entweder dem Nutzer ein Anfrageergebnis strukturiert präsentieren oder bei der Berechnung des Ergebnisses selbst von Nutzen sein. Nur selten werden Term-Hierarchien mit dem Hintergrund einer Ontologie konstruiert. Dies liegt vermutlich an der Aktualität von Ontologien in der Informatik - d.h. die automatische Erzeugung von Ontologien ist ein recht junges Forschungsgebiet. Ansätze, die Termlisten mit einem Anfrage-Hintergrund strukturieren, werden in dieser Arbeit dennoch aufgeführt, da diese bei Verwendung passender Termlisten für unsere Anwendung in Frage kommen.

Die erarbeiteten Verfahren zur Hierarchisierung von Termen wurden entsprechend ihrer Arbeitsweise in Kategorien eingeteilt. Tabelle 1 zeigt die fünf wichtigsten Kategorien: Verfahren die (1.) Terme oder (2.) Dokumente clustern, Ansätze die (3.) linguistische oder (4.) Lernverfahren verwenden sowie (5.) Ansätze, die auf bestehenden Wissensbasen beruhen.

Der Fokus dieser Arbeit liegt auf den Methoden, die direkt auf Termen operieren (Kategorie 1). Die Trennung von Term-Extraktion und Term-Hierarchisierung hat den Vorteil, dass der erste Schritt für Phänotyp-Konzepte optimiert werden kann. Weiterhin ist deren Arbeitsweise sehr transparent, so dass das Ableiten von Beziehungen gut nachvollzogen werden kann. Deshalb sind Verfahren der Kategorie 1 für die automatische Erstellung einer Hierarchie von Phänotyp-Konzepten geeignet. Tabelle 2 zeigt Term-Cluster-Verfahren unterschieden nach der Definition für Ähnlichkeit von Termen.

Die Kombination bestehender Ontologien ist nicht das Ziel der Erstellung einer speziesübergreifenden Ontologie. Aus diesem Grund werden Sekundärverfahren vernachlässigt. Methoden, die Lernverfahren verwenden, werden ebenfalls nur am Rande erwähnt.

An dieser Stelle sei noch auf [Krishnapuram und Kummamuru, 2003] sowie [Buitelaar et al., 2003a] hingewiesen. Dies sind Arbeiten, die einen Überblick über das Thema *ontology mining* geben.

Tabelle 1: Überblick über Methoden zur Term-Hierarchisierung

Kategorie	Referenz
Term-Cluster-Verfahren (siehe Kapitel 2)	siehe Tabelle 2
Dokument-Cluster-Verfahren (siehe Kapitel 3)	[Crouch, 1988] [Cutting et al., 1992] [Blaschke und Valencia, 2002] [Kummamuru et al., 2004] [Mandhani et al., 2003]
Linguistische Verfahren (siehe Kapitel 4)	[Hearst, 1992] [Caraballo, 1999] [Caraballo und Charniak, 1999] [Grefenstette, 1992] [Ryu und Choi, 2006] [Mani et al., 2004] [Maedche und Staab, 2000] [Cimiano et al., 2005]
Lernverfahren	[Snow et al., 2005] [Maedche und Staab, 2004]
Sekundärverfahren	[Cimiano et al., 2003]

Tabelle 2: Überblick der Methoden zur Term-Hierarchisierung mittels Term-Clustering

Kategorie	Grundlage	Methode	Referenz
Term-Cluster-Verfahren	gemeinsame Vorkommen von Termen (siehe: Kapitel 2.1)	<i>DF</i> -Vektoren	[Salton, 1971]
		Plain <i>DF</i>	[Forsyth und Rada, 1986]
Term-Cluster-Verfahren	Kontext von Termen (siehe: Kapitel 2.2)	Subsumption	[Sanderson und Croft, 1999]
		CBS	[Lawrie et al., 2001]
		CAARD	[Krishnapuram und Kummamuru, 2003]
		Kontext-Vektoren	[Gauch und Wang, 1996]
		Modellreduktion	[Schutze und Pedersen, 1997]

1.4 Zielsetzung

Die in dieser Arbeit zusammengetragenen Methoden zur Hierarchisierung von Konzepten auf der Basis eines Korpus von Dokumenten bilden die Grundlage für die Erzeugung einer speziessübergreifenden Phänotyp-Ontologie. Die Konzepte sollen aus *PhenomicDB* extrahiert werden. Als Besonderheit der Domäne stehen die in *PhenomicDB* enthaltenen Links zu den Genotypen sowie die Annotationen der Gene mit Konzepten der GO zur Verfügung. Diese sollen neben unabhängigen Eigenschaften, wie dem gemeinsamen Vorkommen von Termen, ausgenutzt werden.

Gelingt es eine organismenübergreifende Ontologie Phänotyp-spezifischer Konzepte von hoher Qualität zu erzeugen, kann neben der eindeutigen Beschreibung von Phänotypen auch deren maschinelle Verarbeitung unterstützt werden.

1.5 Aufbau der Arbeit

In Kapitel 2 werden Verfahren detailliert beschrieben, die auf einer gegebenen Liste von Termen operieren um Relationen zu berechnen. Kapitel 3 beschreibt Methoden die als Eingabe Dokumente verlangen. Die in Kapitel 4 erläuterten Verfahren verwenden u.U. einen bereits genannten Mechanismus, basieren aber auf linguistische Merkmalen. In Kapitel 5 werden zwei weitere Gruppen von Ansätzen genannt. Kapitel 6 gibt einen kurzen Überblick über Möglichkeiten zur Evaluation von Term-Hierarchien. Das letzte Kapitel 7 beschreibt eine Implementation der Term-Hierarchisierung mittels Subsumption (siehe Kapitel 2.1.3) sowie Ergebnisse einer Evaluation.

1.6 Notationen

In dieser Arbeit wird häufig vom Auftreten eines Terms in Dokumenten die Rede sein. Aus diesem Grund seien folgende Notationen definiert:

N bezeichnet die Anzahl der Dokumente im Korpus. $TF_k(t_i)$ (*term frequency*) ist die Anzahl der Vorkommen von Term t_i in Dokument k geteilt durch die Anzahl aller Terme. $DF(t_i)$ (*document frequency*) ist die Anzahl der Dokumente im Korpus in denen der Term t_i vorkommt geteilt durch N . Analog ist $DF(t_i, t_j)$ die Frequenz der Dokumente in denen die Terme t_i sowie t_j gemeinsam auftreten. DF steht allgemein für *document frequency*.

$TFiDF$ bezeichnet die in der Literatur beschriebene *text-frequency-inverse-document-frequency*: $TF_k(t_i) * \log \frac{1}{DF(t_i)}$. Sie ist ein Maß für das Gewichten von Termen, wobei Terme die im gesamten Korpus häufig auftreten geringer bewertet werden als andere.

2 Term-Hierarchisierung mittels Term-Cluster-Verfahren

In diesem Kapitel werden Methoden zur Term-Hierarchisierung beschrieben, die direkt auf einer gegebenen Liste von Termen operieren. Diese unterscheiden sich in den zugrunde liegenden Annahmen: die Verwandtschaft von Termen kann zum einen auf der Basis gemeinsamer Auftreten und zum anderen auf der Basis des Kontextes definiert werden. Je nach Semantik der berechneten Verwandtschaften, können diese für die Hierarchisierung genutzt werden. Die Erkenntnis, dass zwei Terme ähnliche oder sogar synonyme Bedeutung haben, wobei ein Term der generellere ist, ist offensichtlich von Bedeutung.

2.1 Gemeinsames Auftreten (cooccurrence) als Grundlage der Cluster

Die Grundannahme der folgenden Ansätze ist, dass die Ähnlichkeit von Termen über ein gemeinsames Auftreten in Dokumenten eines Korpus definiert werden kann. Zwei Terme sind demnach ähnlich, wenn sie häufig genug zusammen in einem Bereich definierter Größe eines Dokumentes vorkommen.

2.1.1 DF-Vektoren nach Salton

In frühen Arbeiten zur automatischen Generierung von Thesaurus-Term-Klassen geht es im Wesentlichen um das Clustern synonymier oder ähnlicher Terme⁴. Somit liegt der Fokus dieser Veröffentlichungen auf verschiedenen Cluster-Verfahren sowie dem Ermitteln von dessen Parametern (vergl. [Grossman und Frieder, 2004]).

Eine Möglichkeit, Gruppen von Termen zu ermitteln, wird in [Salton, 1971] diskutiert. Jeder Term wird als N-dimensionaler Vektor dargestellt. Im ersten Schritt wird eine Term-Dokument-Matrix aufgebaut, wobei jede Zeile einem Term und jede Spalte einem Dokument entspricht. Das Element (i, j) ist die Anzahl der Vorkommen von Term t_i in Dokument d_j . Eine (symmetrische) Term-Term-Matrix ergibt sich aus der Berechnung der Ähnlichkeiten zwischen den Termen. Diese ist z.B. gegeben durch den Euklidischen Abstand⁵ oder den Kosinus des Winkels⁶ der durch die Terme gegebenen Vektoren im N-dimensionalen Raum. Auch das Produkt aus der Term-Dokument-Matrix und dessen Transponierter repräsentiert Ähnlichkeiten von Termen.

Die Term-Term-Matrix bildet die Grundlage für diverse Cluster-Algorithmen. Diese Matrix basiert auf zwei Definitionen: (1.) die Repräsentation der Terme und (2.) die Definition der Ähnlichkeiten der Terme. Die Terme werden oft durch Vektoren repräsentiert, die deren Vorkommen in den Dokumenten erfassen. Diese Definition ist anschaulich. Für die Definition der Ähnlichkeit von Termen verwenden Salton und andere Autoren den Winkel oder den Abstand der Vektoren. In diesen Fällen kann die Art der Beziehung nicht gefolgert werden: handelt es sich um Synonyme oder zufällig oft gemeinsam verwendete Terme? Aus diesem Grund genügen diese Definitionen nicht für die hierarchische Anordnung von Termen im Sinne der *ist-ein*-Beziehung.

Die Laufzeit dieser Verfahren ist im Wesentlichen durch die Definition der Ähnlichkeit zweier Terme bestimmt, wobei in jedem Fall $O(n^2)$ (n ist die Anzahl der Terme) Berechnungen durchgeführt werden müssen.

⁴Das Ziel war hier die Expansion von Anfragen. Aus diesem Grund genügte Cluster von Termen.

⁵Der Euklidische Abstand zwischen zwei Vektoren $\bar{A} = (a_1 \dots a_n)$ und $\bar{B} = (b_1 \dots b_n)$ ist definiert als $d(\bar{A}, \bar{B}) = \sqrt{(a_1 - b_1)^2 + \dots + (a_n - b_n)^2}$.

⁶Der Kosinus-Abstand für zwei Vektoren \bar{A} und \bar{B} ist definiert als $\cos \varphi = \bar{A} * \bar{B} / |\bar{A}| * |\bar{B}|$

Listing 1: Algorithmus zur Term-Hierarchisierung [Forsyth und Rada, 1986] (modifiziert)

```

1 for level  $l = 1$  to  $k$  do
2   for term  $i = 1$  to  $n_l$  do
3     compute cooccurrence of  $term_i^l$  with any  $term_j^{l-1}$ 
4     insert edge from all  $term_j^{l-1}$  that appear most with  $term_i^l$ 
5   if  $term_i^{l-1}$  has no descendant then
6     insert  $term_{t'}^l$  // dummy-term
7     insert edge from  $term_i^{l-1}$  to  $term_{t'}^l$ 

```

2.1.2 Plain DF nach Forsyth und Rada

[Forsyth und Rada, 1986] erweitern die Grundannahme von Salton (gemeinsam auftretende Terme haben synonyme Bedeutung) um die Vermutung, dass ein Term t_1 mit einer geringeren DF spezifischer ist als ein Term t_2 mit einer höheren DF . Der Ansatz erzeugt eine Term-Hierarchie vordefinierter Tiefe k . Zu diesem Zweck werden k Schwellwerte $F_1 > F_2 > \dots > F_k$ festgelegt. Tabelle 3 zeigt die Einteilung der Terme anhand der Schwellwerte der DF in Gruppen. Terme der Gruppe i ($i = 1..k$) werden Terme der Ebene i , d.h. Kinder von Termen der Ebene $i - 1$ in der Hierarchie. Der in Listing 1 angegebene Algorithmus iteriert über jede Ebene der Hierarchie und erzeugt sukzessive Eltern-Kind-Beziehungen zwischen den Termen zweier Ebenen auf der Basis gemeinsamen Auftretens. Für den Fall, dass ein Term t keine Kinder hat, wird ein 'dummy'-Term t' eingefügt, damit dieser Term t als Vorfahre für die übernächste Ebene fungieren kann (Zeile 5-7).

Tabelle 3: DF -Bereiche von Termen [Forsyth und Rada, 1986]

Gruppe / Ebene	Bereich	Terme
1	$DF > F_1 - 1$	$term_1^1 \dots term_{n_1}^1$
2	$F_1 > DF > F_2 - 1$	$term_1^2 \dots term_{n_2}^2$
...
k	$F_k > DF > F_{k-1} - 1$	$term_1^k \dots term_{n_k}^k$

Diese Idee ist ein initialer Ansatz, um Terme bzgl. Generalität in Gruppen zu unterteilen. Generalität basiert hier allerdings lediglich auf der DF . Gemeinsames Auftreten von Termen wird erst für direkte Beziehungen herangezogen. Aus diesem Grund muss zu Beginn des Algorithmus die Anzahl der gewünschten Ebenen sowie deren jeweilige Generalität vordefiniert werden. Dies kann für Term-Hierarchien unbekannter Größe von Nachteil sein.

Die Laufzeit des Algorithmus ist linear in der gewünschten Größe der Hierarchie. Dabei ist aber zu beachten, dass sich hinter *compute cooccurrence* der Zugriff auf eine Term-Term-Matrix verbirgt.

Neben [Doyle, 1961] ist die Arbeit von Forsyth und Rada einer der ersten Ansätze zur Ableitung von hierarchischen Beziehungen mittels Statistik über gemeinsames Auftreten von Termen.

2.1.3 Subsumption nach Sanderson und Croft

Sanderson und Croft nutzen ebenfalls Statistiken über gemeinsame Vorkommen von Termen. Sie definieren die Subsumption von Termen [Sanderson und Croft, 1999]. Ein Term t_1 subsumiert

einen Term t_2 , falls die Menge der Bereiche (engl. window), in denen t_2 auftaucht, eine Teilmenge der Bereiche ist, in denen t_1 auftritt. Mit anderen Worten: jeder Bereich der t_2 beinhaltet, enthält ebenfalls t_1 .

$$P(t_1|t_2) = 1, P(t_2|t_1) < 1$$

Ein Bereich kann ein vollständiges Dokument, ein Absatz oder ein Abschnitt vordefinierter Größe sein. Da der Term t_1 den Term t_2 subsumiert und t_1 häufiger auftritt als t_2 , wird angenommen, dass t_1 Vorfahre von t_2 ist. Mit dieser Bedingung werden allerdings Beziehungen, bei denen Vorkommen von t_2 nicht von t_1 begleitet werden, "übersehen". Deshalb kann die erste Subsumptionsbedingung abgeschwächt werden: $P(t_1|t_2) \geq 0,8$. Der Wert 0,8 wurde von den Autoren empirisch ermittelt. Zur Erzeugung der Hierarchie wird eine Bereich-Term-Matrix erstellt, aus der die bedingten Wahrscheinlichkeiten, bzw. die Subsumptionsbeziehungen, extrahiert werden können. Die Hierarchie wird bottom-up aufgebaut, wobei aufgrund von Transitivität redundante Beziehungen⁷ eliminiert werden. In ihrem Paper formulieren die Autoren folgende Anforderungen an die auf automatischem Wege erstellte Konzept-Hierarchie:

1. Terme der Hierarchie sollen aus Texten extrahiert werden und deren Themen reflektieren.
2. Terme sollen derart hierarchisch angeordnet werden, dass Eltern generellere Konzepte darstellen als deren Kinder - d.h. Eltern-Konzepte beinhalten bzw. überdecken deren Kinder-Konzepte.
3. Kinder-Konzepte können mehr als ein Eltern-Konzept besitzen - d.h. die angestrebte Struktur ist ein Gerichteter Azyklischer Graph (DAG).
4. Mehrdeutige Terme sollen mehrere Repräsentationen in der Hierarchie besitzen - eine für jede Bedeutung.

Anforderung 2 sowie 3 werden durch die Subsumption erfüllt. Diese Kriterien entsprechen erstaunlich gut der automatischen Erzeugung einer Ontologie: Terme, die Konzepte repräsentieren, werden entsprechend ihrer Generalität in einem DAG angeordnet. Man beachte, dass es in der Originalarbeit um das Repräsentieren eines Anfrageergebnisses mittels einer Themen-Hierarchie geht. Dies ist allerdings lediglich eine Frage der zur Extraktion der Themen verwendeten Mechanismen (Anforderung 1). Sanderson und Croft nutzen hierzu Terme, die die Anfrage erweitern⁸ zzgl. Terme die überdurchschnittlich oft im Ergebnis einer Anfrage auftauchen. Die Extraktion von Phrasen ist in [Sanderson und Lawrie, 2000] beschrieben. Das Problem der Mehrdeutigkeit (Anforderung 4) von Termen betrachten die Autoren unter der Voraussetzung, dass die Dokumente im Korpus alle ein Thema behandeln, als gelöst. Für den Fall, dass der Korpus thematisch heterogen ist, wird in einer späteren Arbeit [Lawrie und Croft, 2000] gezeigt, dass die Ergebnis-Hierarchie unter initialer Erstellung semantisch homogener Dokument-Cluster deutlich verbessert werden kann.

Der Aufwand des beschriebenen Verfahrens setzt sich aus drei Summanden zusammen: dem Erstellen der Bereich-Term-Matrix ($O(ndl)$), wobei n die Anzahl der Terme, d die Anzahl der Dokumente, l die durchschnittliche Länge der Dokumente ist); dem Berechnen von Subsumption-Beziehungen ($O(n^2)$); dem Aufdecken von Transitivitäten ($O(k^3)$, wobei k die Anzahl der gefundenen Beziehungen ist). Die in den Veröffentlichungen genannten Experimente beziehen sich lediglich auf einige 100 Konzepte. Die Frage, ob unser Anwendungsfall mit diesem Verfahren in akzeptabler Laufzeit behandelt werden kann, wird in Kapitel 7 beantwortet.

⁷Die Beziehung $\text{ist-ein}(t_1, t_3)$ ist redundant, falls die Beziehungen $\text{ist-ein}(t_1, t_2)$ und $\text{ist-ein}(t_2, t_3)$ existieren.

⁸Der verwendete Mechanismus ist Local Context Analysis [Xu und Croft, 1996].

2.1.4 Context-based subsumption (CBS) nach Lawrie et al.

Die Bezeichnung CBS der folgenden Methode zur Erstellung von Term-Hierarchien stammt aus [Mani et al., 2004]. Dort wird sie im Rahmen eines Frameworks zur Zusammenfassung von Dokumenten implementiert. In [Lawrie et al., 2001] wird das Verfahren beschrieben. Es gelten folgende Anforderungen:

1. Terme höherer Ebenen tauchen gemeinsam mit vielen anderen Termen auf.
2. Das Auftreten von Termen unterer Ebenen ist abhängig vom Auftreten von Termen übergeordneter Ebenen - d.h. übergeordnete Terme besitzen Vorhersagekraft (engl. predictiveness) für untergeordnete Terme.
3. Topic-Terme sollen maximale Abdeckung von Nicht-Topic-Termen haben, d.h. sie sollen einen möglichst großen Bereich des Vokabulars vorhersagen.

Die Autoren unterscheiden also zwischen Topic- und Nicht-Topic-(Vokabular-)Termen. Topic-Terme sind diejenigen, die den Inhalt von Dokumenten repräsentieren. Sie sagen Vokabular-Terme vorher. Allgemein findet die CBS (Topic-)Terme, die möglichst viele andere (Vokabular-)Terme vorhersagen. Die Vorhersagekraft eines Termes t_1 für einen Term t_2 wird durch eine bedingte Wahrscheinlichkeit repräsentiert: $P_x(t_1|t_2)$ ist die Anzahl der Auftreten von t_1 in maximaler Entfernung x von t_2 geteilt durch die Anzahl des Auftretens von t_2 . t_1 ist ein möglicher Topic-Term und t_2 ein Vokabular-Term. In den Experimenten ist jeder möglicher Topic-Term auch ein Vokabular-Term. Das Modell der bedingten Wahrscheinlichkeiten wird nun wie folgt in einen Graph konvertiert:

- Jeder Topic-Term und Vokabular-Term wird ein Knoten.
- Ist $P_x(t_1|t_2) > 0$, wird eine Kante (t_1, t_2) mit dem Gewicht $P_x(t_1|t_2)$ eingefügt. Das Kantengewicht repräsentiert die Affinität zwischen den Termen t_1 und t_2 .
- Das Gewicht eines Knotens t_1 ist $\sum_{t_i|t_i \text{ benachbart}} P_x(t_1|t_i)$ - d.h. die Summe aller bedingten Wahrscheinlichkeiten, die Kanten zu diesem Knoten erzeugen - auch Summe der Affinitäten des Termes zu anderen Termen genannt.

Unter der Prämisse, dass eine steigende bedingte Wahrscheinlichkeit $P_x(t_1|t_2)$ für den Term t_1 als Topic-Term von t_2 spricht, ist die Frage nach Termen, die andere Terme mit möglichst großer Überdeckung vorhersagen, eine Version des *Dominating Set Problem (DSP)*⁹. Da das DSP NP-hart ist, nutzen Lawrie, Croft und Rosenberg die in Listing 2 dargestellte Heuristik DSPapprox um das Problem zu approximieren.

Die Eingabe des Algorithmus ist der Graph G , die möglichen Topic-Term *CCT* (engl. candidate topic terms) sowie die Anzahl k maximal gewünschter Topic-Terme. Nach der Initialisierung iteriert das Verfahren bis das Vokabular vollständig von Topic-Termen überdeckt ist oder die gewünschte maximale Anzahl an Topic-Termen berechnet wurde. Jeder neue Topic-Term wird aufgrund seines maximalen Gewichtes, d.h. maximale Affinität zu Vokabular-Termen, ausgewählt (Zeile 8). Zu den von ihm dominierten Vokabular-Termen muss eine Affinität größer als ein berechneter Schwellwert (*thresh*) bestehen (Zeile 9). Dadurch wird vermieden, dass zwar

⁹Gegeben ein Graph $G = (V, E)$ mit Knotengewichten w_v für alle $v \in V$, finde eine Teilmenge $D \subset V$ der Knoten (die gesuchten topic-Terme), so dass für jeden Knoten $u \in V - D$ ein Knoten $v \in D$ existiert, mit $\{u, v\} \in E$ und $\sum_{v \in D} w_v$ minimal. In diesem Fall wird die Summe maximiert.

```

1 VT = V - CCT // identify vocabulary terms
2 D = ∅ // initialize dominating topic terms
3 VocabDominated = ∅ // initialize dominated vocabulary terms
4 thresh = mean(we(CCT, VT)) // mean of all affinities
5 for each c ∈ CCT do // compute vertex weights
6     wv(c) = ∑v ∈ VT we(c, v)
7 while (VocabDominated ≠ VT and |D| < k) do
8     d = argmaxc ∈ CCT wv(c) // choose the heaviest vertex
9     vDominated = {v | we(d, v) ≥ thresh} // identify dominated terms
10    D = D ∪ d
11    CCT = CCT - d
12    VocabDominated = VocabDominated ∪ vDominated ∪ d
13    for each v ∈ vDominated do // adjust vertex weights
14        for each c ∈ CCT do
15            wv(c) = wv(c) - we(c, v)
16 return D

```

benachbarte, jedoch weit entfernte Terme dominiert werden. Um sicherzustellen, dass in jeder Iteration ein Topic-Term mit neuen adjazenten Vokabular-Termen gewählt wird, werden die Gewichte der Knoten angepasst (Zeile 13-15).

DSPapprox berechnet anfangs lediglich die erste Ebene einer Term-Hierarchie. Um weitere Ebenen zu erstellen, wird das Modell erweitert: $P_{x,t_3,y}(t_1|t_2)$ ist die Anzahl der Vorkommen von t_1 in maximaler Entfernung x von t_2 (wie bisher), wobei der Eltern-Term t_3 in maximaler Entfernung y von t_1 zu finden sein muss. Durch die Variation der Bereiche (x bzw. y) lässt sich die Granularität der Term-Hierarchie von Ebene zu Ebene beliebig abstufen. Aus den Modellen für nachfolgende Ebenen wird wieder ein Graph konstruiert, auf dem der gegebene Algorithmus operieren kann.

Sowohl für die Subsumption als auch für die CBS bildet die Bereich-Term-Matrix die Grundlage der Berechnung. Für die Subsumption wird ein Schwellwert definiert, ab dem ein Term t_1 als verwandt (im Sinne von Generalität) zu einem Term t_2 betrachtet wird. Im Gegensatz dazu findet die CBS eine Menge von Termen, die andere Terme möglichst optimal vorhersagen. Die Größe dieser Menge ist unbekannt, wird aber durch die Anwendung der Heuristik vorher festgelegt. Dies erfordert initiale Einschränkungen und ist ein Nachteil gegenüber der Subsumption. Andererseits lässt sich so leicht die Wirkung verschiedener k evaluieren. Die Ebenen werden sukzessiv erstellt, was weitere Interaktionen (zur Festlegung der Parameter) mit dem Verfahren voraussetzt. Der iterative Aufbau der Ebenen hat ebenfalls eine wiederholte Berechnung des Modells ($P_{x,t_3,y}(t_1|t_2)$) zur Folge. Der Aufwand hierfür liegt bei $O(dl)$ (d ist die Anzahl der Dokumente, l ist die durchschnittliche Länge der Dokumente), was bei genügend Dokumenten von Nachteil sein kann.

In den Experimenten wurde keine Unterscheidung von Topic- und Vokabular-Termen vorgenommen. Es kann also jeder Term ein Topic- oder Vokabular-Term sein. In unserem Zusammenhang bedeutet dies, dass wir keine Konzepte vordefinieren würden. In [Lawrie und Croft, 2003b]¹⁰ wird neben der Vorhersagekraft die Aktualität (engl. topicality) von Termen definiert.

¹⁰Eine ausführlichere Version dieser Veröffentlichung der SIGIR'03 ist [Lawrie und Croft, 2003a].

Die Autoren nutzen hierzu die Kullback-Leiber-Divergenz¹¹, wobei $P_H(t)$ der Wahrscheinlichkeit des Terms t in den Dokumenten zur Erzeugung der Hierarchie und $P_{GE}(t)$ der Wahrscheinlichkeit des Terms t in allgemeiner englischer Sprache entspricht.

$$KLdivergence(t) = topicality(t) = P_H(t) \log \frac{P_H(t)}{P_{GE}(t)}$$

Terme die eine Aktualität größer eines definierten Schwellwertes besitzen, werden in die Hierarchisierung eingebunden. Dies sind offensichtlich jene, für die $P_H(t) \gg P_{GE}(t)$ gilt.

2.1.5 CAARD nach Krishna und Krishnapuram

Der in [Krishna und Krishnapuram, 2001] vorgestellte Ansatz verwendet eine Version des Leader Clustering [Hartigan, 1975] zur Erstellung einer Konzept-Hierarchie. Allgemein gruppiert er Datenpunkte, die in asymmetrischer Beziehung zueinander stehen. Sei eine Menge von Datenpunkten $D = \{d_1..d_n\}$ (hier Konzepte), eine asymmetrische Relation $R : D \times D \rightarrow [0, 1]$ (hier Inklusion von Konzepten) sowie ein maximaler Abstand m eines Elementes vom Cluster-Leader gegeben. Gesucht ist eine Menge von Clustern $C = \{C_1..C_n\}$ mit $C_i \subset D$ sowie eine Menge von Cluster-Repräsentanten $S \subset D$ mit $s_i \in S$ ist Repräsentant für C_i .

Das gegebene Problem wird von den Autoren mittels der in Listing 3 dargestellten Heuristik CAARD (Clustering Algorithm for Asymmetrically Related Data) approximiert. Zu Beginn berechnet CAARD die lead-values der Datenpunkte. Der lead-value eines Datenpunktes gibt Auskunft über dessen Qualifikation als Cluster-Repräsentant. Der Datenpunkt mit dem höchsten lead-value wird in jedem Fall ein Cluster-Repräsentant (Zeile 4). Jeder weitere Datenpunkt wird aufgrund seiner Relation zu den bisher gefundenen Cluster-Repräsentanten in einen bestehenden Cluster eingefügt oder zur Menge der Repräsentanten hinzugefügt (Zeile 5-9), d.h. er wird Repräsentant eines neuen Clusters.

In den vorgestellten Experimenten wird CAARD zur Erstellung einer Konzept-Hierarchie für die Zusammenfassung von Dokumenten eingesetzt. Die in Relation zu setzenden Konzepte werden in Form von N -dimensionalen Vektoren \bar{v}_k als gegeben vorausgesetzt. Das i -te Element eines Vektors \bar{v}_k ist 1, falls das Dokument i das Konzept k enthält. Die Inklusion eines Konzeptes k in einem Konzept l stellt die asymmetrische Relation dar. Sie wird wie folgt quantifiziert: $R(\bar{v}_k, \bar{v}_l) = |\bar{v}_k \cdot \bar{v}_l| / |\bar{v}_k|$. Als lead-value eines Konzeptes wird seine DF oder $TFiDF$ verwendet.

Ergebnis dieses Verfahrens sind Cluster von Konzepten zzgl. je einem Repräsentanten. Die Repräsentanten bilden die erste Ebene einer Konzept-Hierarchie. Nachfolgende Ebenen werden durch die erneute Anwendung der Methode auf alle Elemente eines Clusters abzüglich dessen Repräsentanten gebildet, d.h. für jeden Knoten in der ersten Ebene wird eine Menge von Konzepten gruppiert. Da der lead-value eines Konzeptes in jeder Iteration von CAARD gleich bleibt, muss dieser nur ein Mal berechnet werden. CAARD selbst ist linear in der Anzahl der Datenpunkte, was trotz häufiger Wiederholung (entsprechend der Anzahl der internen Knoten im Baum) für eine akzeptable Laufzeit sorgen sollte. Die Wiederholung von CAARD terminiert bei Eintreten einer Abbruchbedingung - z.B. eine minimale oder maximale Anzahl an Unterkonzepten. CAARD ist ein klassisches Cluster-Verfahren. Es erzeugt eine Hierarchie von Konzepten auf einfache Art und Weise. Dies hat den Vorteil, dass die Erzeugung der Hierarchie gut nachvollziehbar ist, und sich eine Implementierung somit auf das Optimieren des lead-values sowie der asymmetrischen Relation konzentrieren kann. Der lead-value ist kritisch für die (greedy) Ab-

¹¹Die Kullback-Leiber-Divergenz misst den Abstand zweier Wahrscheinlichkeitsverteilungen: einer zufälligen Verteilung (hier $P_{GE}(t)$) sowie einer beliebigen weiteren (hier $P_H(t)$).

```

1 for each  $d \in D$  do
2   compute lead-value
3 sort  $D$  by lead-values in descending order
4 add  $d$  with highest lead-value to  $S$ 
5 for each remaining  $d \in D$  do
6   if  $R(d, s_i) > m | s_i \in S$  then
7     add  $d$  to  $C_i$  // add to cluster
8   else
9     add  $d$  to  $S$  // create new cluster
10 return  $S, C = \{C_1..C_n\}$ 

```

arbeitung der Elemente. Es ist fraglich, ob es gelingt, die Verteilung unserer multidimensionalen Daten in einem solchen Wert zu erfassen.

2.1.6 Asymmetrisches gemeinsames Auftreten nach Chen et al.

CAARD basiert auf der Definition von lead-values sowie einer asymmetrischen Relation. Der lead-value eines Konzeptes ist von Bedeutung für dessen Qualifikation als Knoten in der Konzept-Hierarchie. Die asymmetrische Relation sollte die Generalität eines Terms bzgl. anderer Terme möglichst gut erfassen. In [Peat und Willett, 1991] wird argumentiert, dass symmetrisches gemeinsames Auftreten von Termen häufig auftretende Terme bevorzugt, und diese somit oft in eine enge Beziehung zueinander setzt.

[Chen et al., 1995] definieren in ihrer Arbeit ein verfeinertes Maß $rel(t_a, t_b)$, um die Beziehung zwischen zwei Termen zu erfassen. Hierbei wird ein *TFiDF*-Maß $w_j(t_i)$ (Term t_i in Dokument d_j) definiert, welches Phrasen höher bewertet als Terme. p_i ist die Anzahl an Termen die eine Phrase t_i bilden. Des Weiteren werden Terme oder Phrasen, die häufiger auftreten, niedriger gewichtet (beachte $DF(t_i)$ in $W(t_i)$).

Die Relation $rel(t_a, t_b)$ beschreibt die Affinität eines Terms t_b zu einem gegebenen Term t_a - d.h. Terme t_b mit hohen Affinitäten zu Term t_a sind wohlmöglich gute Kandidaten für Unterkonzepte von t_a .

$$w_j(t_i) = TF_j(t_i) * \log\left(\frac{1}{DF(t_i)} * p_i\right)$$

$$W(t_i) = \frac{\log \frac{1}{DF(t_i)}}{\log N}$$

$$rel(t_a, t_b) = \frac{\sum_{k=1..n} \min(TF_k(t_a), TF_k(t_b)) \log\left(\frac{1}{DF(t_a, t_b)} * p_b\right)}{\sum_{k=1..n} w_k(t_a)} * W(t_b)$$

Die Definition einer solchen (asymmetrischen) Relation im Allgemeinen ist ausschlaggebend für die sich ergebende Konzept-Hierarchie. Sie sollte in jedem Fall (1.) die Verwandtschaft zweier Konzepte erfassen sowie (2.) Aussagen über deren Generalität treffen.

Bei allen bisher gezeigten Verfahren lässt sich letzteres schnell einsehen. Die Frage ist eher, ob die auf diesem Wege in Beziehung gesetzten Konzepte auch diejenigen sind, die semantisch verwandt sind. Zum einen werden Experimente diese Frage beantworten und zum anderen kann

der Kontext von Termen helfen, dieses Problem zu bearbeiten. Aufgrund dessen werden im folgenden die Kontext-Vektoren nach Gauch und Wang sowie Schütz und Pedersen vorgestellt.

2.2 Kontext von Termen als Grundlage der Cluster

In diesem Kapitel wird die Ähnlichkeit von Termen unter Berücksichtigung von deren Kontext definiert. Der Kontext eines Terms ist gegeben durch benachbarte Terme. Dies hat den Vorteil, dass sich unterschiedliche Bedeutungen von Homonymen unterscheiden lassen. Sanderson und Croft verlassen sich darauf, dass Terme in einem semantisch homogenen Korpus nur eine Bedeutung haben - dies ist aber nicht direkt aus der Term-Dokument- oder Term-Term-Matrix ableitbar. Ein Beispiel für ein und den selben Term mit unterschiedlicher Bedeutung ist das Wort **Bank** - das Kreditinstitut oder die Sitzgelegenheit.

2.2.1 Kontext-Vektoren nach Gauch und Wang

Entsprechend des intuitiven Ansatzes der Vektoren von *DFs*, ist es auch für den Kontext nahe liegend, ihn als Vektor zu modellieren. In [Gauch und Wang, 1996] repräsentiert ein Vektor $V_i = [v_{-3}, v_{-2}, v_{-1}, v_1, v_2, v_3]$ von Vektoren $v_k = \langle c_{k,1} \dots c_{k,l} \rangle$ den Kontext eines Terms t_i . Jeder Vektor v_k beschreibt die an der k -ten Position links bzw. rechts des Terms t_i vertretenen Kontext-bildenden Terme. Jede Komponente $c_{k,x}$ eines Vektors v_k entspricht der Anzahl der Vorkommen eines dedizierten Kontext-Terms t_x an der Position k des Terms t_i .

Die Autoren verwenden ein Fenster der Größe sieben. Die dedizierten Kontext-Terme bilden die 200 häufigsten Terme der Dokumente im Korpus. Ein den Term t_i beschreibender Vektor hat dann die Form $V_i = [\langle c_{-3,1} \dots c_{-3,200} \rangle, \langle c_{-2,1} \dots c_{-2,200} \rangle, \langle c_{-1,3} \dots c_{-1,200} \rangle, \langle c_{1,1} \dots c_{1,200} \rangle, \langle c_{2,1} \dots c_{2,200} \rangle, \langle c_{3,1} \dots c_{3,200} \rangle]$ - d.h. er besitzt 1200 Komponenten. Die durch solche Kontext-Vektoren beschriebenen Terme werden dann unter Verwendung des Kosinus der Winkel untereinander geclustert. Das Ergebnis eignet sich nicht für die Hierarchisierung, stellt aber sehr intuitive Gruppen von synonymen Termen dar.

2.2.2 Reduktion der Term-Term-Matrix nach Schütze und Pedersen

In [Schütze und Pedersen, 1997] wird der Kontext-Vektor eines Terms etwas genereller definiert. Die Annahme ist, dass semantisch verwandte Terme in genügend Dokumenten mit ähnlichen Termen gemeinsam auftreten. Die Autoren vernachlässigen die genaue Position eines benachbarten Terms und betrachten allgemein alle Terme innerhalb eines definierten Bereiches links bzw. rechts des betrachteten Terms. Dies ist offensichtlich eine Überschneidung mit den in Kapitel 2.1 vorgestellten Ansätzen. Die Vektoren, welche gemeinsame Vorkommen repräsentieren, werden allerdings dermaßen in ihrer Dimensionalität reduziert, dass man diese lediglich noch als Kontext-Repräsentanten interpretieren kann. Ausgangspunkt für diesen Ansatz ist die Term-Term-Matrix gemeinsamer Vorkommen. Da diese Datenstruktur bei genügend Termen (hier 450.000¹²) aufgrund des Platzbedarfs und des Rechenaufwands des Clustering problematisch ist, werden die Vektoren stark in ihrer Dimension reduziert. Der Prozess besteht aus vier Phasen:

(1.) Es wird für 3.000 Terme mittlerer Frequenz eine Term-Term-Matrix aufgebaut. Ein Element $c_{i,j}$ dieser Matrix entspricht der Anzahl gemeinsamer Vorkommen der Terme t_i und t_j innerhalb eines Bereiches von $k = 40$ Termen. Die Vektoren werden in 200 Gruppen (verschiedener Topics), auch A-Klassen genannt, geclustert.

¹²Diese und alle folgenden Beispiel-Angaben sind den Experimenten der Originalarbeit entnommen.

(2.) Für die 20.000 häufigsten Terme wird nun eine A-Klasse-Term-Matrix konstruiert. Die (Spalten-)Vektoren dieser Matrix werden ein weiteres Mal mittels Clustering in 200 Gruppen, jetzt B-Klassen, unterteilt.

(3.) An dieser Stelle wird für alle Terme und Phrasen eine B-Klasse-Term-Matrix M ermittelt. Ihre Dimension beträgt $200 \times v$, wobei v die Anzahl der Terme (hier 450.000) ist. Auf M wird nun *Singular Value Decomposition* (SVD)¹³, ein Verfahren zur Reduktion der Dimension einer Matrix auf p (hier $p = 20 < 200$) angewendet. Der Vorteil der Anwendung von SVD im letzten Schritt im Gegensatz zu einem weiteren Clustering liegt laut den Autoren in der Glättung der Topic-Klassen (anfangs A, später B, jetzt C), so dass jede C-Klasse möglichst ein separates Thema repräsentiert.

(4.) Die resultierenden 450.000 Kontext-Vektoren der Dimension 20 werden wieder unter Verwendung des Kosinus der Winkel geclustert. Es ergeben sich Gruppen von Kontext-Vektoren.

Unter der Prämisse, dass synonyme Terme häufig in einem ähnliche Kontext erwähnt werden, liefern solche Verfahren akkurate Gruppen von Termen verwandter Bedeutung. Diese Annahme scheint auf den ersten Blick intuitiver, als semantische Verwandtschaft mittels gemeinsamer Vorkommen zu definieren. Die vorgestellten Mechanismen liefern allerdings keine Möglichkeit der hierarchischen Einordnung der Terme. Sie eignen sie sich aber als Grundlage für hierarchisierende Arbeitsschritte.

¹³SVD wird in [Golub und Loan, 1989] beschrieben.

3 Term-Hierarchisierung mittels Dokument-Cluster-Verfahren

In den bisher beschriebenen Verfahren wird direkt mit Termen operiert, um diese in sog. Thesaurus-Klassen einzuteilen oder auf direktem Wege in eine Hierarchie einzuordnen. Die Ansätze in diesem Kapitel verfolgen ein indirektes Vorgehen. Die Dokumente eines Korpus werden gruppiert - flach oder hierarchisch. Aus den erzeugten Dokument-Klassen werden dann beschreibende Terme extrahiert, die Thesaurus-Klassen bilden. Die Art der Beziehung der Terme, sowohl innerhalb als auch zwischen den verschiedenen Klassen oder Ebenen, bleibt oft unklar. Die Prominenz dieser Verfahren in der Literatur belegt ihre Bedeutung, weswegen hier kurz auf sie eingegangen wird. Für unsere Aufgabenstellung liefern sie jedoch nicht ohne weiteres das gewünschte Ergebnis.

Auf die in den vorgestellten Methoden verwendeten Cluster-Algorithmen wird hier nicht im Detail eingegangen. Es sei hierfür auf die Originalarbeiten oder beispielsweise [Zhao und Karypis, 2002] sowie [Hartigan, 1975] verwiesen.

Der in [Crouch, 1988] beschriebene Ansatz konstruiert Thesaurus-Klassen für klassisches *Document Retrieval*: Anfragen sollen sinnvoll erweitert werden. Basierend auf dem *discriminating value model*¹⁴ werden Klassen von verwandten Termen mit geringem Vorkommen gesucht. Da das direkte Clustern der Terme laut den Autoren aufgrund unzureichender Informationen in den Dokumenten nicht sinnvoll ist, bzw. keine aussagekräftigen Cluster erzeugt, werden Dokumente geclustert. Aus den entstandenen Gruppen von Dokumenten werden die Terme geringen Auftretens je Gruppe extrahiert. Eine Thesaurus-Klasse von verwandten Termen wird dann aus der Schnittmenge der Terme der Dokumente einer Gruppe gebildet. Eine weitere Möglichkeit der Kombination der Terme aus den Dokumenten ist die Bildung der Vereinigung. Sowohl die Schnittmenge als auch die Vereinigung kann außerdem sortiert und unter Verwendung eines Schwellwertes dezimiert werden.

Eine weitere viel zitierte Arbeit ist [Cutting et al., 1992]. Hier wird ein Tool zum hierarchischen Durchsuchen einer Menge von Dokumenten beschrieben: *Scatter/Gather*. Dokumente werden dem Nutzer gruppiert präsentiert. Dieser wählt einen oder mehrere Cluster aus, deren Dokumente neu gruppiert werden, so dass eine Hierarchie von Dokument-Clustern entsteht. Die Dokument-Gruppen der unterschiedlichen Ebenen werden mittels sog. *cluster digests* beschrieben: diejenigen Terme die am häufigsten in einer Gruppe auftauchen zzgl. Ausschnitte der Dokumente die das Zentrum einer Gruppe bilden.

[Blaschke und Valencia, 2002] wenden hierarchisches Clustering auf wissenschaftliche Texte über den Organismus *Saccharomyces cerevisiae* an. Die Texte werden mittels des SRS am EBI¹⁵ extrahiert. Die Gennamen werden durch das Suchen heuristischer Muster in den Texten bestimmt, woraus eine Gen-Text-Assoziation entsteht. Für jeden Text wird eine Menge von signifikanten Termen (in [Andrade und Valencia, 1998] beschrieben) gewonnen und bewertet. Das hierarchische Clustering arbeitet mit einem einfachen Ähnlichkeitsmaß für zwei bestehende Cluster, welches das Mittel der Gewichte der gemeinsamen signifikanten Terme berechnet. Das Ergebnis dieses Verfahrens ist ein binärer Baum, dessen innere Knoten die Ähnlichkeiten der Dokument-Cluster bzw. deren assoziierter Gene (und deren Funktionen) repräsentieren. Die

¹⁴Das *discriminating value model* ordnet Termen einen Wert zu, der ihre Qualifikation zur Charakterisierung von Dokumenten widerspiegelt. Es wurde in [Salton et al., 1974] gezeigt, dass die *DF* eines Terms dieses Maß adäquat approximiert, was dessen komplexe Berechnung erspart.

¹⁵SRS: Sequence Retrieval System, EBI: European Bioinformatics Institute, srs.ebi.ac.uk

Blätter bilden Gruppen von Texten. Die Knoten des Baumes sind mit den signifikanten Termen der Cluster beschrieben. Von den Autoren werden die auf diesem Wege erstellten Ähnlichkeiten als "ontologische Information" bezeichnet.

Bei den bisher auf Clustering von Dokumenten basierenden Verfahren sind die Label der internen Knoten sowie deren Beziehungen in der Hierarchie meist unklar. Die Autoren von [Kummamuru et al., 2004] definieren in ihrer Arbeit Kriterien für die Wahl von Unterkonzepten:

- Möglichst alle Dokumente eines Korpus sollen mittels der Konzepte der Hierarchie vertreten sein: Abdeckung (engl. coverage)
- Benachbarte Konzepte einer Ebene sollen sich möglichst wenig überschneiden: Klarheit (engl. distinctiveness)

Das in Listing 4 dargestellte Vorgehen *DisCover* baut jede der k Ebenen einer Hierarchie sequentiell auf. Für jedes gegebene Konzept i werden alle Dokumente ermittelt, die dieses Konzept enthalten. Dies ist der Dokument-Cluster zu einem Konzept. Dessen Dokumente enthalten in der Regel weitere Konzepte (C in Zeile 1). Die Wahl, welche Teilmenge dieser (Unter-)Konzepte die nächste Ebene bilden soll, wird mittels Maximierung getroffen: es werden die Konzepte mit maximaler Abdeckung und Klarheit ausgewählt (*coverage* sowie *distinctiveness* in Zeile 4). In der Originalarbeit sind *coverage* und *distinctiveness* auf Mengen basierend definiert. Bei der Maximierung gehen sie gewichtet in die Summe ein: $w_c = 0,8$ und $w_d = 0,2$ wurden empirisch ermittelt. Die Originalarbeit erläutert nicht, wie die Wurzel einer Konzept-Hierarchie bestimmt wird. Dies kann allerdings leicht ein abstraktes Wurzel-Konzept sein, welches alle Konzepte unter sich vereint. Die Label einer solchen Hierarchie sind einzelne Konzepte. Die Intuition ihrer Anordnung (*coverage* sowie *distinctiveness*) ist klar, jedoch ungeeignet im Sinne von Generalität. Man beachte, dass dieses Verfahren wegen der Ordnung der (Unter-)Konzepte auch eine Version des Leader Clustering ist.

Listing 4: DisCover [Kummamuru et al., 2004] (modifiziert)

```

1  $C = \text{set of all concepts under concept } i$ 
2  $S = \text{empty sequence // of assigned subconcepts of concept } i$ 
3 for  $k = 1$  to  $|C|$  do // or some threshold  $< |C|$ 
4      $next = \text{argmax}_{c \in C} (w_c * \text{coverage}(c) + w_d * \text{distinctiveness}(c))$ 
5      $S = S \cup next$ 
6      $C = C - \{next\}$ 

```

Ebenfalls auf Leader Clustering beruht die Methode in [Mandhani et al., 2003]. Deren Grundlage ist die Dokument-Term-Matrix. Die Elemente der Matrix sind eine Form des *TFiDF*-Maßes. Der lead-value eines Dokumentes ist der Durchschnitt der *TFiDF*-Werte der im Dokument vertretenen Terme (dessen sog. Länge).

Für die Cluster werden möglichst dichte Submatrizen um die lead-Dokumente der Dokument-Term-Matrix gebildet. Diese Co-Cluster (Dokumente und Terme der Submatrix) sind die Blätter der Dokument-Term-Hierarchie. Die Ähnlichkeit zweier solcher Cluster (Submatrizen) A und B ist der Durchschnitt der Elemente, wobei Matrix A die Zeilen-Indizes der Matrix B bekommt und umgekehrt. Auf diese Weise werden Cluster für darüberliegende Ebenen vereinigt.

Das Ergebnis ist eine Hierarchie von Dokument- bzw. Term-Clustern. Als Label eines Knotens empfehlen die Autoren diejenigen Terme eines Co-Clusters, die am häufigsten in den Dokumenten eines Co-Clusters vorkommen. Die Bezeichnung der Knoten im Baum besteht wieder aus mehreren Termen, d.h. nicht einem dedizierten Konzept. Dennoch ist die Zusammenstellung der Termlisten wegen des direkten Bezugs zu den Co-Clustern nahe liegend.

4 Term-Hierarchisierung mit Hilfe linguistischer Merkmale

Die bisherigen Abschnitte beschreiben Methoden, die im Wesentlichen mittels Statistik über Terme oder Dokumente Beziehungen zwischen Termen oder Phrasen herstellen. Abgesehen von Tokenisierung¹⁶ ist keine weitere Vorverarbeitung des Textes von Nöten.

Die in diesem Kapitel beschriebenen Verfahren basieren auf der syntaktischen Analyse von Texten. Die Token eines Textes werden durch morphologische Attribute beschrieben - handelt es sich z.B. um einen Singular oder Plural. Des Weiteren werden Wortklassen (engl. part-of-speech) bestimmt und Terme normalisiert (engl. lemmatization).

Die Rolle als *head* oder *modifier* von Termen in Phrasen ist eine weitere wesentliche Charakteristik, die von solchen Verfahren ausgenutzt wird. Zur Veranschaulichung überlege man sich diverse Verbindungen, in denen das Wort **research** vorkommen kann: **market-research**, **research-project**, **research-center** usw. Im ersten Fall hat **research** die Rolle des *head* und wird durch den Term **market** modifiziert. In den weiteren Beispielen ist **research** hingegen der *modifier*.

4.1 Syntaktische Muster

Eine einfache Form der Nutzung der Syntax von Texten wird in [Hearst, 1992] beschrieben. In der Arbeit werden die in Tabelle 4 dargestellten syntaktischen Muster, eine Art regulärer Ausdrücke, in Texten gesucht (NP steht für *Noun Phrase*). Es ist intuitiv, dass jedes dieser Muster auf eine bestimmte Beziehung zwischen den Termen schließen lässt. Die Beziehungen lassen sich entsprechend des Musters sogar qualifizieren: i.d.R. geht es um Unter- bzw. Oberbegriffe, also *ist-ein*-Beziehungen. Aus diesen Informationen lässt sich somit leicht eine Term-Hierarchie erstellen.

Hearst gibt weiterhin eine informelle Beschreibung der Vorgehensweise zur Herleitung solcher syntaktischen Muster an. Man erstelle eine Liste von Term-Paaren die einer gewünschten Beziehung entsprechen. Als nächstes werden diese Term-Paare im Korpus identifiziert, so dass deren Kontexte auf Gemeinsamkeiten untersucht werden können. Bei genügend Evidenz einer Gemeinsamkeit im Kontext, kann dieser als ein neues Muster verstanden werden.

Tabelle 4: Syntaktische Muster [Hearst, 1992]

Muster	Beispiel	Beziehung
such NP as (NP,)*(or and) NP	works by such authors as Herrick, Shakespeare, ...	Herrick ← author Shakespeare ← author
NP (,NP)*(or and) other NP	bruises, wounds and other injuries	bruise ← injury wound ← injury
NP, including (NP,)*(or and) NP	countries, including Germany, and England	Germany ← country England ← country
NP, especially (NP,)*(or and) NP	countries, especially France, or England	France ← country England ← country

Der große Vorteil eines solchen Ansatzes ist, dass auf diesem Wege extrahierte Beziehungen gewissermaßen von Hand erstellt worden sind. Eine Person hat sie erfasst und niedergeschrieben. Dieses Verfahren extrahiert sie wieder. Offen bleibt allerdings die Frage nach der Vollständigkeit, da diese von der Dimension des Korpus abhängt. Ist der Korpus umfassend genug, kann eine

¹⁶Tokenisierung bezeichnet in der Computerlinguistik die Segmentierung eines Textes in Einheiten - i.d.R. Worte, manchmal auch Sätze, Absätze o.ä.

hohe Anzahl an Beziehungen aufgedeckt werden. Dies ist die Grundvoraussetzung für den Erfolg einer solchen Methode.

Der in [Caraballo, 1999] beschriebene Ansatz nutzt Konjunktionen sowie Ergänzungen, um Nomen als Vektor zu repräsentieren. Die gewonnenen Vektoren werden unter Verwendung der Kosinus-Distanz hierarchisch geclustert. Das Ergebnis ist auch hier ein binärer Baum mit Gruppen ähnlicher Terme in den Blättern.

Das Interessante an diesem Ansatz ist die Bestimmung der Label der internen Knoten. Dafür wird der Korpus nach Mustern der Form $NP(, NP)^*(or|and)otherNP'$ durchsucht. Daraus wird gefolgert, dass NP' ein Hyperonym (Oberbegriff) der Nomen in der Aufzählung ist. Für jedes Nomen in den Blättern des Baumes wird nun ein weiterer Vektor erstellt. Dessen i -te Komponente ist 1, falls NP_i als Hyperonym für dieses Nomen identifiziert worden ist. Die internen Knoten des Baumes werden mit der Summe der Vektoren der Kinder assoziiert. Das Label eines solchen Knoten ist dann jenes Hyperonym, dessen Komponente in der Summe der Vektoren am größten ist, d.h. das am häufigsten als Oberbegriff für die Kinder auftretende Nomen.

4.2 head und modifier

In [Grefenstette, 1992] wird die Syntax eines Satzes genauer untersucht. Es werden Phrasen extrahiert sowie die jeweiligen *head*- und *modifier*-Terme bestimmt. Die zugrunde liegende Annahme dieser Arbeit ähnelt sehr einer oben bereits besprochenen: Terme mit ähnlichem Kontext haben vermutlich synonyme Bedeutung. Der Kontext besteht in diesem Fall nicht aus Termen an einer bestimmten Position, sondern aus der Rolle als *head* oder *modifier* eines Terms. Zwei Terme t_1 und t_2 , die häufig den selben Term t_3 modifizieren sowie oft von einem Term t_4 modifiziert werden, haben demnach eine synonyme Bedeutung.

Es ist ebenfalls möglich, *head* und *modifier* zur Quantifizierung der Generalität bzw. Spezifität¹⁷ von Termen zu nutzen. Die Autoren von [Caraballo und Charniak, 1999] formulieren die These, dass ein Term, der häufig modifiziert wird, eher generell ist - im Gegensatz zu Termen, die seltener in modifizierter Form auftreten. Sie extrahieren *head*-Terme t_h , die von Adjektiven, Verben oder Nomen modifiziert werden. Die Generalität solcher Terme t_h wird auf folgende einfache Weise erfasst: $P(t_h) = \frac{df(t_h, t_m)}{DF(t_h)}$, wobei $df(t_h, t_m)$ die Frequenz der Vorkommen von t_h modifiziert durch einen Term t_m darstellt.

Ryu und Choi bezeichnen diesen Ansatz zur Messung der Generalität von Termen als *outside information*. Die Autoren definieren in [Ryu und Choi, 2006] außerdem die *inside information* von Phrasen. Sie ermitteln die Spezifität zusammengesetzter Phrasen aus der Spezifität des *head*-Terms zzgl. der gewichteten Spezifitäten gegebener *modifier*-Terme. In den angeführten Experimenten evaluieren die Autoren ihren Ansatz gegen einen bestehenden Thesaurus - einen Teilbaum des *MeSH*¹⁸-Thesaurus (9.430 Bezeichnungen für Krankheiten).

Wenn auch deutlich einfacher, aber ebenfalls auf *head* und *modifier* in Phrasen beruht ein in [Mani et al., 2004] genannter Ansatz. Hier werden *kind-of*-Beziehungen aus Phrasen mittels Subphrasen extrahiert. So steht beispielsweise das Protein *p68-protein* in einer *kind-of*-Beziehung zu *protein* oder *federal-income-tax* ist eine Form von *tax*.

Es ist erstaunlich, welche Menge verschiedenster Informationen sich aus *head*- und *modifier*-Beziehungen extrahieren lässt: semantische Verwandtschaft, Generalität, *kind-of*-Relationen.

¹⁷Spezifität ist ein Maß der Menge an Informationen eines Terms. Sie wird auch *Semantischer Inhalt* genannt (vgl. [Ryu und Choi, 2006]).

¹⁸*MeSH* steht für Medical Subject Headings und ist das kontrollierte Vokabular der National Library of Medicine. Siehe www.nlm.nih.gov/mesh

Grundlage dessen ist allerdings eine Infrastruktur, die diese detaillierten linguistischen Informationen extrahieren kann. Um sie dementsprechend interpretieren zu können, ist deutlich mehr Know-How als für die Verwendung syntaktischer Muster notwendig.

4.3 Logik und linguistische Merkmale

In [Maedche und Staab, 2000] wird ein Ansatz basierend auf [Srikant und Agrawal, 1995] beschrieben. Es werden dabei zu einer gegebenen Hierarchie $H \subset C \times C$ von Konzepten in C und einer Menge T von Transaktionen t lediglich neue (unqualifizierte) Beziehungen aufgedeckt, jedoch zwei interessante Maße definiert. Maedche und Staab erstellen durch linguistische Heuristiken eine Menge von Konzept-Paaren $CP = \{(c_1, c_2) | c_1, c_2 \in C\}$. Daraus definieren sie Transaktionen T sowie *support* und *confidence* einer Assoziation $A \Rightarrow B$ ($A, B \subset C, A \cap B = \{\}$). Der angegebene Algorithmus bestimmt anfangs lediglich den *support* einer Assoziation. Für den Fall, dass dieser einen definierten Grenzwert erreicht, wird deren *confidence* ermittelt. Diese muss wiederum einen Schwellwert überschreiten, damit die Assoziation $A \Rightarrow B$ im Ergebnis auftaucht.

$$T = \{\{c_1, c_2..c_l..\} | (c_1, c_2) \in CP \wedge l \geq 3 \rightarrow ((c_1, c_l) \in H \vee (c_1, c_l) \in H)\}$$

$$\text{support}(A \Rightarrow B) = \frac{|\{t | A \cup B \subseteq t, t \in T\}|}{|T|}$$

$$\text{confidence}(A \Rightarrow B) = \frac{|\{t | A \cup B \subseteq t, t \in T\}|}{|\{t | A \subseteq t, t \in T\}|}$$

Die Semantik der auf diesem Wege extrahierten Assoziation ist nicht klar. Dennoch kann es von Interesse sein, zwischen gleichartigen Konzepten (z.B. Synonyme als Cluster-Ergebnis) Beziehungen herzustellen. Je nach Definition der Kriterien, sind solche Beziehungen intuitiv. Berücksichtigen *support* und *confidence* gemeinsame Vorkommen von Konzepten, können die Assoziationen als Generalisierungen/Spezialisierungen verstanden werden. Diese Definitionen wären der von Sanderson und Croft sehr ähnlich, jedoch verschärft durch die Anforderungen an die Transaktionen in T .

Der wohl modernste Ansatz unter Verwendung der Syntax von Texten ist der in [Cimiano et al., 2005]. Es wird eine Menge von Term-Paaren (t_1, t_2) aus Texten extrahiert, wobei t_1 ein Verb und t_2 das entsprechende Subjekt oder Objekt ist. Diese Term-Paare werden dann einer formalen Konzeptanalyse (engl. Formal Concept Analysis, FCA) unterzogen. FCA ist eine Methode, um mittels Logik implizite Beziehungen zwischen Objekten, beschrieben durch Attribute, herauszustellen. Diese Menge von Beziehungen kann dann in eine kompakte Halbordnung, d.h. eine Hierarchie, transformiert werden. Ein wesentlicher Bestandteil der FCA ist der formale Kontext (engl. formal context).

Ein Tripel (G, M, I) heißt formaler Kontext, falls G und M Mengen sind und $I \subseteq G \times M$ eine binäre Relation. Die Elemente in G sind die Objekte und die in M die Attribute. I ist das Auftreten (engl. incidence) des Kontextes. Für $A \subseteq G$ und $B \subseteq M$ ist weiterhin definiert: $A' = \{m \in M | \forall g \in A : (g, m) \in I\}$ bzw. $B' = \{g \in G | \forall m \in B : (g, m) \in I\}$. Das Tupel (A, B) heißt formales Konzept von (G, M, I) , falls $A \subseteq G, B \subseteq M, A' = B$ und $A = B'$.

Zur Verdeutlichung seien die in Tabelle 5 dargestellten Objekte (in den Zeilen) mit ihren Attributen (in den Spalten) gegeben. Das Auftreten von Attributen für Objekte (I) ist durch x gekennzeichnet. Formale Konzepte sind durch die nahe liegende Unter-/Ober-Konzept-Relation

geordnet. Weitere logische Umformungen ermöglichen es mittels dieser Relation, Hierarchien von Konzepten zu erstellen.

$$(A_1, B_1) \leq (A_2, B_2) \Leftrightarrow A_1 \subseteq A_2, \text{ wobei } A_1 \subseteq A_2 \Leftrightarrow B_2 \subseteq B_1$$

Tabelle 5: Beispiel für Objekte und Attribute eines formalen Kontextes [Cimiano et al., 2005]

	bookable	rentable	drivable	rideable	joinable
hotel	x				
apartment	x	x			
car	x	x	x		
bike	x	x	x	x	
excursion	x				x
trip	x				x

Auf Logik basierende Ansätze haben den Vorteil, dass sie nicht der Interpretation bedürfen. Es ist klar, dass das in Tabelle 5 dargestellte Konzept **rentable** ein Unterkonzept von **bookable** ist. Auch das Optimieren von Parametern ist für den Ansatz von Cimiano et al. nicht nötig. Ausgangspunkt dieser logischen Analyse ist wieder eine Menge linguistischer Merkmale, für deren Analyse es einer entsprechenden Infrastruktur bedarf.

5 Weitere Ansätze zur Term-Hierarchisierung

Zwei Klassen von Verfahren zur Erzeugung von Term-Hierarchien haben in dieser Arbeit bisher keine Erwähnung gefunden:

Dies sind zum einen Methoden, die bestehende Wissensbasen, insbesondere Ontologien, wiederverwenden. Durch deren Verknüpfung und den Einsatz von Transitivität kann auf neue Beziehungen geschlossen werden. Exemplarisch sei [Cimiano et al., 2003] genannt. In dieser Arbeit werden unterschiedliche Ansätze implementiert. Es wird z.B. via Google nach Mustern der Form *t₁ such as t₂* gesucht, um die Evidenz von Beziehungen zu erhöhen.

Zum anderen gibt es Ansätze, die Lernverfahren verwenden. Es ist gut vorstellbar, dass syntaktische Muster für Hyperonym-Beziehungen durch Maschinelles Lernen ermittelt werden können [Snow et al., 2005]. Es ist ebenfalls möglich, Klassifikatoren für Beziehungen auf einer gegebenen Ontologie zu trainieren, um neue Relationen herzustellen [Maedche und Staab, 2004]. Dies verwundert nicht, da viele Probleme, die in entsprechend abstrakter Form vorliegen und für die ein Trainingsdatensatz besteht, auf diese Weise bearbeitet werden können.

6 Evaluation von Term-Hierarchien

Dieses Kapitel erläutert kurz existierende Ansätze zur automatischen Evaluation von generierten Konzept-Hierarchien. Solche Methoden sind entweder für *Document Retrieval* konzipiert, oder sie stellen einen Vergleich zwischen einer (im Idealfall als *Ground Truth* anerkannten) bestehenden Ontologie und der generierten Hierarchie an. Im Vorfeld dieser Arbeit wurden Güte-Parameter einer Term-Hierarchie als Größen von Optimierungsverfahren in Betracht gezogen. Klassisches *Document Retrieval* entspricht nicht der übergeordneten Vision dieser Arbeit. Vergleichende Verfahren setzen voraus, dass Referenz-Hierarchien vorhanden sind. Insofern sind die angesprochenen Evaluations-Mechanismen nicht für Optimierungsverfahren geeignet. Es werden eher Konzepte wie die oben erwähnte *distinctiveness* (siehe Kapitel 3) oder $P_{x,t_3,y}(t_1|t_2)$ (siehe Kapitel 2.1.4) von Interesse sein.

Vergleichenden Maße sind aber nützlich, um beispielsweise die von Hand erstellte *Mammalian Phenotype Ontologie* einem Ausschnitt (bzw. der Schnittmenge) einer automatisch erstellten Konzept-Hierarchie gegenüberzustellen.

Für die Evaluation bzgl. *Document Retrieval* seien die Maße *document coverage* und *reachability* genannt [Kummamuru et al., 2004] [Sanderson und Lawrie, 2000]. *document coverage* berechnet, wie groß der Anteil der in den Dokumenten vertretenen Konzepte in einer Hierarchie ist. *reachability* misst, wie kompliziert/einfach es für einen Nutzer ist, Dokumente mittels einer solchen Konzept-Hierarchie aufzufinden. Weiterhin kann das *Expected Mutual Information Measure (EMIM)* für die Menge der Topic- und Vokabular-Terme berechnet werden. Dieses Maß erfasst, wie weit zwei Verteilungen von stochastischer Unabhängigkeit abweichen. Je größer die Abhängigkeit, desto besser spiegelt eine Hierarchie das Vokabular bzw. die Dokumente wieder [Lawrie und Croft, 2003a].

Sollen Term-Hierarchien verglichen werden, kann die *Mutual Information (MI)* für jede Ebene der Hierarchien genutzt werden [Krishnapuram und Kummamuru, 2003]. Sie misst, inwieweit Verteilungen unabhängig voneinander sind. Beim Vergleich von Hierarchie-Ebenen bleibt aber die Frage der Kombination der Einzelwerte: Sollen z.B. höhere Ebenen stärker bewertet werden als andere?

Ein nahe liegender Ansatz ist der Vergleich von Eltern-Kind-Paaren. Die Anzahl gemeinsamer Paare kann zudem bzgl. der Gesamtanzahl der Paare oder der Anzahl der Paare einer Hierarchie normalisiert werden [Lawrie und Croft, 2000]. Dieser Ansatz wird in [Mani et al., 2004] derart erweitert, dass eine gewisse Distanz der Paare erlaubt ist: $precision(H_1, H_2, d)$ ist der Anteil der Distanz-1-Beziehungen in H_1 die maximal eine Distanz-d-Beziehung in H_2 sind.

In [Maedche und Staab, 2002] wird das Konzept nicht nur mit seinen Eltern oder Kindern betrachtet. Die dort definierte *Semantic Cotopy* zweier Konzepte wird mittels deren Super- und Sub-Konzepten erfasst. Der darauf aufbauende *Taxonomic Overlap* berechnet einen Durchschnitt dieser Werte über alle Konzepte einer Ontologie.

7 Ergebnisse der Subsumption nach Sanderson und Croft

Die zu implementierende Methode sollte auf einfache Weise Statistik über die Vorkommen von Termen ausnutzen. Es wurde die Subsumption nach Sanderson und Croft ausgewählt (siehe Kapitel 2.1.3). Dieses Verfahren teilt die Dokumente im Korpus in Bereiche der Größe w und zählt diejenigen in denen zwei Konzepte gemeinsam vorkommen. Ein Konzept *concept* ist genereller als ein Konzept *subconcept*, falls jeder Bereich der *subconcept* beinhaltet, ebenfalls *concept* enthält: $P(\text{concept}|\text{subconcept}) = 1$. In der sich ergebenden Ontologie ist *concept* ein Vorfahre von *subconcept*.

Die Subsumption wird auf den Daten aus *PhenomicDB* durchgeführt. Es stehen etwa 71.100 Dokumente mit natürlichsprachlichen Beschreibungen zur Verfügung. Diese Dokumente haben eine maximale Länge von 1000 Termen. Die in Beziehung zu setzenden Konzepte stammen aus der *GO* sowie der *MPO*. Es werden nur diejenigen Terme oder Synonyme der Ontologien betrachtet, die exakte Vorkommen in den Phänotypendaten aufweisen. Von ca. 81.200 *GO*-Konzepten (Terme zzgl. Synonyme) wurden somit lediglich 1000 Konzepte für Beziehungen in Betracht gezogen. Die Evaluation wurde mittels *ist-ein*-Beziehungen aus den Original-Ontologien durchgeführt. Es werden nur solche Beziehungen betrachtet, an denen Konzepte die exakte Vorkommen in den Phänotypendaten aufweisen beteiligt sind. Tabelle 6 gibt einen Überblick der zu bearbeitenden Daten.

Tabelle 6: Ausgangsdaten für die Subsumption: Konzepte und *ist-ein*-Beziehungen der *GO* sowie der *MPO*

	GO	MPO
Anzahl Terme/Synonyme im Original	81.197	12.552
Anzahl Terme/Synonyme mit exakten Vorkommen in PhenomicDB	998	3.176
Anteil Terme/Synonyme mit exakten Vorkommen in PhenomicDB (in %)	1,2	25,3
Anzahl ist-ein-Beziehungen im Original	33.865	6.828
Anzahl ist-ein-Beziehungen von Konzepten mit exakten Vorkommen in PhenomicDB	1.427	3.335
Anteil ist-ein-Beziehungen von Konzepten mit exakten Vorkommen in PhenomicDB (in %)	4,2	48,8
Anzahl exacter Vorkommen der Terme/Synonyme in 71.712 Phänotypen	176.501	218.278

7.1 Technische Details

Das Program, geschrieben in Java 5.0¹⁹, ist entsprechend seiner Teilaufgaben in verschiedene Klassen unterteilt: Abbildung 1 gibt einen Überblick. Das Paket `common` beinhaltet Klassen, die allgemein von Interesse sind. So ist eine Umsetzung einer Dokument-Term-Matrix (Interface `DocumentConceptMatrixIF`) sicher für unterschiedliche Ansätze von Bedeutung. Für die vorliegende Anwendung wurde eine Implementierung auf der Basis einer Hash-Tabelle²⁰ (Implementierung `DocumentConceptMatrixHT`) erarbeitet. Das Paket `obo` enthält einen Parser für das *OBO Dateiformat*²¹ in dem sowohl die *GO*²² als auch die *MPO*²³ vorliegen. Im Paket `subsumption` sind die für diese Methode spezifischen Klassen zusammengefasst: `SubsumptionFinder` berechnet entsprechend eines gegebenen Schwellwertes t die Beziehungen; `Test` fasst die einzelnen Arbeitsschritte (vom Parsen der Phänotypen bis zur Berechnung der Beziehung) zusammen.

¹⁹java.sun.com/j2se/1.5.0

²⁰java.sun.com/j2se/1.5.0/docs/api/java/util/Hashtable.html

²¹www.geneontology.org/GO.format.obo-1.2.shtml, obofoundry.org

²²obo.cvs.sourceforge.net/*checkout*/obo/obo/ontology/genomic-proteomic/gene_ontology_edit.obo

²³obo.cvs.sourceforge.net/*checkout*/obo/obo/ontology/phenotype/mammalian_phenotype.obo

Alle für das Verfahren benötigten Parameter können in `properties.xml` gesetzt werden: Ein- und Ausgabe-Dateien, w , t sowie die Art der Bereiche.

Alle Experimente wurden auf einem Linux-Server²⁴ mit vier Intel(R) Xeon(TM) 2,6GH Prozessoren und 12GB Arbeitsspeicher durchgeführt. Die zu verarbeitenden Daten konnten vollständig in den Hauptspeicher geladen werden. Die Laufzeit der Test variiert für die vorliegenden Daten von 4 bis 140 min. Sie ist im Wesentlichen (1.) durch die Größe der Bereiche w und (2.) den Parameter, ob es sich um überlappende oder voneinander verschiedene Bereich (siehe unten) handeln soll, bestimmt.

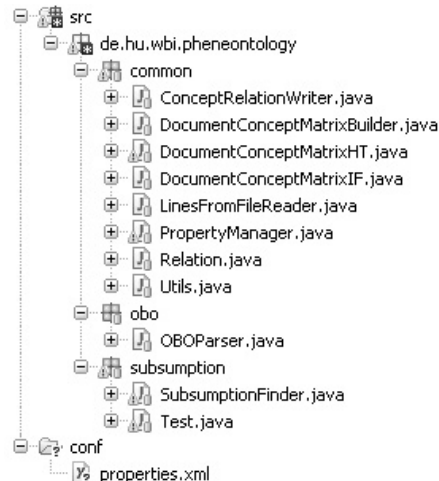


Abbildung 1: Paket-Struktur und die enthaltenen Klassen der Implementierung

7.2 Ergebnisse

Alle Experimente wurden sowohl für Konzepte der *GO* als auch der *MPO* auf den Daten aus *PhenomicDB* durchgeführt. Die Menge der betrachteten Konzepte ist diejenige Teilmenge der Terme und Synonyme aus den Original-Ontologien die in *PhenomicDB* exact vorkommen. Die Performance der Subsumption hängt von drei Parametern ab:

1. t : der Schwellwert der ersten Subsumptionsbedingung: $P(\text{concept}|\text{subconcept}) \geq t$
2. w : der Größe der Bereiche in die ein Dokument unterteilt wird
3. ob diese Bereiche 'gleitende Bereiche' (engl. sliding windows) oder nicht überlappende Bereiche sind

In den Originalarbeiten werden $t = 0,8$ und $w = 40$ vorgeschlagen. Es wird allerdings keine Aussage über das Überlappen der Bereiche getroffen. In diesen Experimenten wurden beide Varianten getestet. Die Auswirkungen der Parameter t und w sind in beiden Fällen ähnlich. Mit überlappenden Bereichen werden aber im Schnitt 10% bessere Ergebnisse erzielt. Deshalb werden die folgenden Ergebnisse auf überlappenden Bereichen beruhen. Es sei allerdings bemerkt, dass die Laufzeit natürlich höher ist, da ein Konzept bei überlappenden Bereichen in deutlich mehr Bereichen vorkommt, als bei nicht überlappenden Bereichen.

²⁴paprika.informatik.hu-berlin.de

Die Performance der verschiedenen Parametersätze (w und t) wurde mittels der in Kapitel 6 erwähnten Funktion $precision(H_{gen}, H_{truth}, d)$ evaluiert. Sie misst den Anteil der Distanz-1-Beziehungen (Term-Paare) in H_{gen} , die maximal eine Distanz- d -Beziehung in H_{truth} sind. H_{gen} entspricht der Menge der von der Subsumption erzeugten Beziehungen. H_{truth} ist die GO bzw. die MPO. Der Abstand d soll hier zum einen zwei (*distance-2*-Beziehungen) und zum anderen beliebig bis zur Wurzel (korrekte Beziehungen) sein. *distance-2*-Beziehungen haben eine sehr hohe Qualität: in diesen Fällen hat die Methode von Hand erstellte Beziehungen quasi 'nachvollzogen'.

Abbildung 2 zeigt den Anteil der korrekten und *distance-2*-Beziehungen an den erkannten Beziehungen (*Precision*). Es wird deutlich, dass bei $t = 0,9$ und $w = 10$ ein Maximum von 69,5% bzw. 46,3% in der Qualität der erkannten Beziehungen liegt. D.h. scharfe Kriterien (hohe Übereinstimmung kleiner überlappender Bereiche) erreichen die höchste Präzision. Im Gegensatz dazu erzielen schwächere Kriterien (z.B. $t = 0,5$ und $w = 40$) eine deutlich höhere absolute Anzahl an Beziehung - siehe Abbildung 3, links. Trotz dessen ist der Anstieg der (interessanten) *distance-2*-Beziehungen weitaus geringer - siehe Abbildung 3, rechts. Daraus folgt, dass mit schwächeren Parametern zwar mehr, aber überwiegend Beziehungen schlechter Qualität erzeugt werden.

Tabelle 7 zeigt den *Recall* und die *Precision* für die vielversprechenden Parameter $t = 0,9$ und $w = 10$. Der F_1 -Measure²⁵ wurde für $t = 0,5..1,0$ und $w = 10..40$ berechnet. Aufgrund eines niedrigen *Recalls* hat die sich verändernde *Precision* aber nur geringen Einfluss, weshalb der F_1 -Measure relativ gleich bleibend bei 0,24 (für korrekte Beziehungen) liegt. Erst der $F_{0,1}$ -Measure²⁶ lässt eine steigende Qualität für $t = 0,1..0,9$ und $w = 40..10$ beobachten.

Werden die Beziehungen zu einem DAG zusammengesetzt, ist festzustellen, dass es sich für strenge Kriterien um eine flache Hierarchie der Tiefe zwei handelt. Für GO-Konzepte ($w = 10$, $t = 0,9$) ergeben sich Pfade wie [formation → spindle-formation → mitotic-spindle-formation]. Diese Beziehungen basieren eindeutig auf Subphrasen. Der Großteil des Ergebnis besteht aus derartigen Beziehungen. Wird die Größe des Bereiches w erhöht, steigt die Anzahl der Beziehungen die nicht auf Subphrasen gründen (bei gleichzeitig fallender *Precision*). Die Tiefe der Hierarchie lässt sich auf diesem Wege ebenfalls erhöhen. Für GO-Konzepte ($w = 30$, $t = 0,6$) ergeben sich dann Pfade wie [pronucleus → male-pronucleus → female-pronucleus → female-pronucleus-formation]. Es ist auffällig, dass die zweite Beziehung [pronucleus → female-pronucleus] lauten sollte.

Tabelle 7: Precision und Recall für GO- und MPO-Beziehungen ($w=10$; $t=0,9$; überl. Bereiche)

	Anzahl	Precision	Recall
korrekte ist-ein-Beziehungen aus der GO	198	0.69	0.14
distance-2 ist-ein-Beziehungen aus der GO	132	0.46	0.09
korrekte ist-ein-Beziehungen aus der MPO	197	0.51	0.06
distance-2 ist-ein-Beziehungen aus der MPO	157	0.41	0.05

²⁵Der F_1 -Measure ist definiert als: $F_\alpha = \frac{(1+\alpha)*Precision*Recall}{((\alpha*Precision)+Recall)}$, mit $\alpha = 1$.

²⁶Der $F_{0,1}$ -Measure ist der F_α -Measure, mit $\alpha = 0,1$.

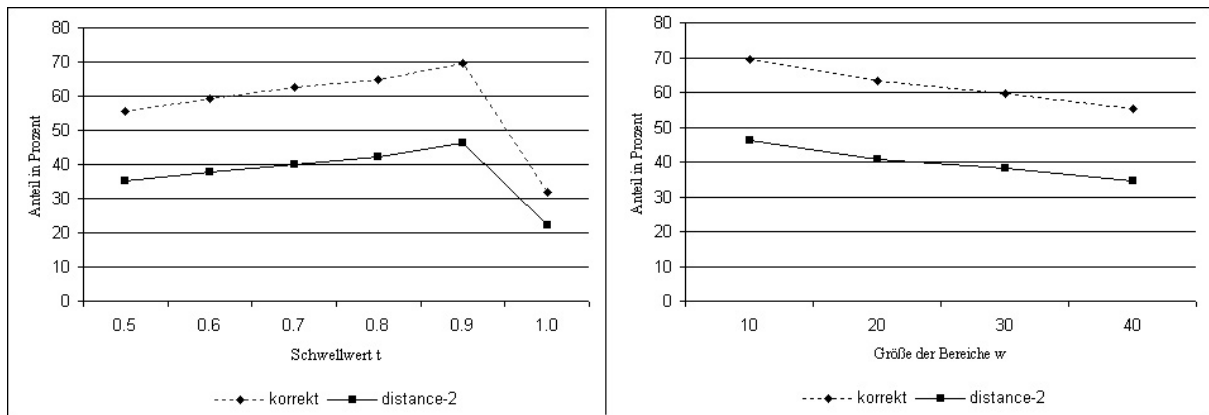


Abbildung 2: Anteil der korrekten und *distance-2*-Beziehungen (Term-Paare) an den berechneten Beziehungen der Konzepte der GO (in %) (Precision); links: Größe der Bereiche $w = 10$; rechts: Schwellwert $t = 0,9$; Bereiche sind in beiden Fällen überlappend

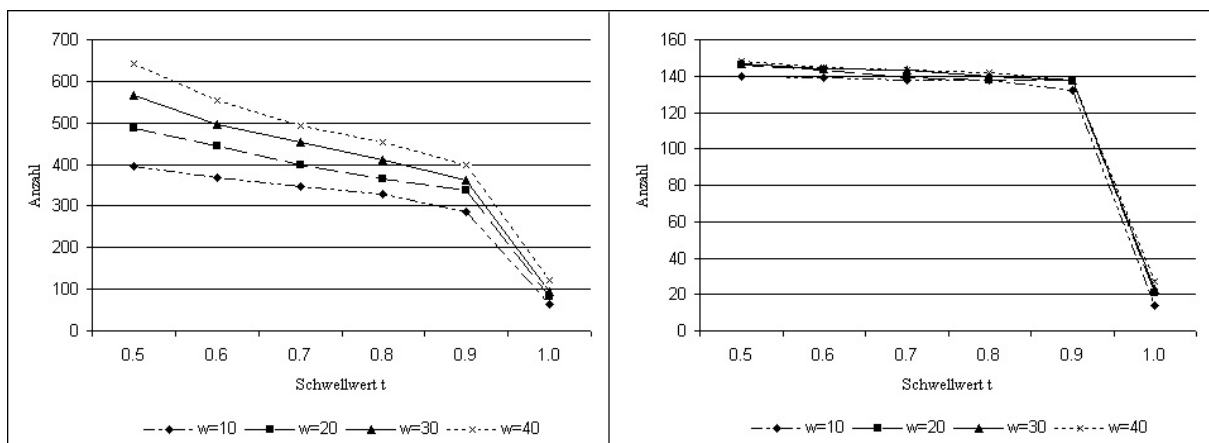


Abbildung 3: Anzahl der berechneten Beziehungen (Term-Paare) der Konzepte der GO nach Größe der Bereiche w und Schwellwert t ; links: Anzahl aller Beziehungen; rechts: Anzahl *distance-2*-Beziehungen; Bereiche sind in beiden Fällen überlappend

7.3 Bewertung und Ausblick

Das Prinzip der Subsumption lässt sich unkompliziert umsetzen. Dies begründet sich in der Einfachheit des Ansatzes: Dokumente in Bereiche unterteilen und gemeinsame Vorkommen von Konzepten zählen. Die Laufzeit ist trotz des hohen Aufwandes gering, da sowohl die etwa 71.000 Phänotypen als auch die ca. 1.000 (GO-) bzw. 3.300 (MPO-) Konzepte in den Hauptspeicher geladen werden können. Die Term-Bereich-Matrix kann bei einer deutlich größeren Anzahl an Konzepten allerdings nicht mehr ohne weiteres im Arbeitsspeicher vorgehalten werden. Dies ist z.B. der Fall bei etwa 13.700 *MeSH*-Konzepten. Soll eine entsprechende Menge an Konzepten verarbeitet werden, muss die Matrix ggf. auf dem Sekundärspeicher abgelegt werden.

Der Recall der Subsumption auf den vorliegenden Daten ist mit 14% (siehe Tabelle 7) für Beziehungen der GO etwas dürftig. Dies liegt zum einen an der Suche nach exakten Vorkommen von Phrasen. Wird die Anzahl der Vorkommen durch *fuzzyness* erhöht, können mehr Beziehungen erkannt werden. Ein weiterer Grund für den niedrigen Recall kann aber ebenfalls sein, dass gemeinsam Vorkommen nicht die gesuchten *ist-ein*-Beziehungen widerspiegeln. Demgegenüber

steht eine passable Precision von 69%. Dies ist für einen allgemeinen Ansatz auf biomedizinischen Daten ein gutes Ergebnis. Die Precision sollte stärker bewertet werden als der Recall, da es nicht der Anspruch sein kann, eine vollständige Ontologie auf automatischem Wege zu erzeugen. Ist die Qualität der erkannten Beziehungen hoch, kann dies für Anerkennung in der Community sorgen und somit ein Ausgangspunkt für (manuelle) Weiterentwicklungen sein.

In folgenden Arbeiten soll eine Methode erarbeitet werden, die neben domänenunabhängigen Eigenschaften auch die Links der Phänotypen zu den Genotypen und deren Annotationen mit Konzepten der GO verwendet. Auf diesem Wege wird sicher ein höherer Recall zu erreichen sein. Werden für diese Arbeiten ebenfalls die Konzepte auf automatischem Wege extrahiert, ist die direkte Vergleichbarkeit zwar nicht mehr gegeben; die vorliegenden Ergebnisse können aber mit den Resultaten jeder anderen Methode (angewendet auf die GO- bzw. MPO-Konzepte) verglichen werden.

8 Zusammenfassung

Diese Arbeit klassifiziert und beschreibt Methoden des *Information Retrieval* zur Hierarchisierung von Phänotyp-spezifischen Konzepten. Es wurden drei wesentliche Gruppen von Verfahren identifiziert: Term- und Dokument-Cluster-Verfahren sowie Hierarchisierung auf der Basis linguistischer Merkmale. Der Fokus dieses Überblicks liegt auf den Term-Cluster-Verfahren, da sie für die anvisierte Anwendung auf Phänotyp-Daten geeignet sind. Eine wesentliche Unterscheidung dieser Methoden liegt in der Definition der Ähnlichkeit von Termen. Hier werden der Kontext und gemeinsame Auftreten von Termen herangezogen.

In Experimenten wurde gezeigt, dass ein domänenunabhängiges Verfahren auf der Basis gemeinsamer Vorkommen von Termen eine gute *Precision* von ca. 70% bei der Erzeugung von *ist-ein*-Beziehungen erreichen kann. Für den *Recall* von derzeit maximal 14% gibt es noch Verbesserungsbedarf. Alle Messungen wurden auf Daten aus *PhenomicDB* und Konzepten der *GO* sowie der *MPO* durchgeführt.

Literatur

- [Andrade und Valencia, 1998] Andrade, M. A. and Valencia, A. (1998). Automatic extraction of keywords from scientific text: application to the knowledge domain of protein families. *Bioinformatics*, 14(7):600–607.
- [Bier et al., 2007] Bier, F., von Nickisch-Rosenegk, M., Ehrentreich-Förster, E., Reiß, E., Henkel, J., Strehlow, R., and Andresen, D. (2007). Dna microarrays. *Advances in Biochemical Engineering/Biotechnology*.
- [Blaschke und Valencia, 2002] Blaschke, C. and Valencia, A. (2002). Automatic ontology construction from the literature. *Genome Informatics*, 13:201–203.
- [Buitelaar et al., 2003a] Buitelaar, P., Cimiano, P., and Bernardo, M. (2003a). *Ontology Learning from Text: An Overview*, chapter I, pages 3–14. In [Buitelaar et al., 2003b].
- [Buitelaar et al., 2003b] Buitelaar, P., Cimiano, P., and Magnini, B., editors (2003b). *Ontology Learning from Text: Methods, Evaluation and Applications*. IOS Press Publication.
- [Caraballo, 1999] Caraballo, S. A. (1999). Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*, pages 120–126, Morristown, NJ, USA. Association for Computational Linguistics.
- [Caraballo und Charniak, 1999] Caraballo, S. A. and Charniak, E. (1999). Determining the specificity of nouns from text. In *Proceedings of the Joint SIGDAT Conference on EMNLP and Very Large Corpora*.
- [Chen et al., 1995] Chen, H., Yim, T., Fye, D., and Schatz, B. (1995). Automatic thesaurus generation for an electronic community system. *J. Am. Soc. Inf. Sci.*, 46(3):175–193.
- [Cimiano et al., 2005] Cimiano, P., Hotho, A., and Staab, S. (2005). Learning concept hierarchies from text corpora using formal concept analysis. *Journal on Artificial Intelligence Research*, 24:305–339.
- [Cimiano et al., 2003] Cimiano, P., Pivk, A., Schmidt-Thieme, L., and Staab, S. (2003). *Learning Taxonomic Relations from Heterogeneous Sources of Evidence*, chapter II.4, pages 59–76. In [Buitelaar et al., 2003b].
- [Crouch, 1988] Crouch, C. J. (1988). A cluster-based approach to thesaurus construction. In *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 309–320, New York, NY, USA. ACM Press.
- [Cutting et al., 1992] Cutting, D. R., Karger, D. R., Pedersen, J. O., and Tukey, J. W. (1992). Scatter/gather: a cluster-based approach to browsing large document collections. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 318–329, New York, NY, USA. ACM Press.
- [Doyle, 1961] Doyle, L. B. (1961). Semantic road maps for literature searchers. *J. ACM*, 8(4):553–578.
- [Forsyth und Rada, 1986] Forsyth, R. and Rada, R. (1986). *Adding an edge.*, pages 198–212. Chichester: Ellis Horwood; Halsted Press, New York.
- [Gauch und Wang, 1996] Gauch, S. and Wang, J. (1996). Corpus analysis for trec 5 query expansion. In *TREC*.
- [GO-Consortium, 2006] GO-Consortium (2006). The gene ontology (go) project in 2006. *Nucleic Acids Res.*, 34(Database-Issue):322–326.
- [Golub und Loan, 1989] Golub, G. H. and Loan, C. F. V. (1989). *Matrix Computations*. Johns Hopkins University Press, Baltimore, MD, USA, second edition.
- [Grefenstette, 1992] Grefenstette, G. (1992). Use of syntactic context to produce term association lists for text retrieval. In *SIGIR '92: Proceedings of the 15th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 89–97, New York, NY, USA. ACM Press.

- [Grossman und Frieder, 2004] Grossman, D. A. and Frieder, O. (2004). *Information Retrieval Algorithms and Heuristics*. Springer.
- [Groth et al., 2007] Groth, P., Pavlova, N., Kalev, I., Tonov, S., Georgiev, G., Pohlenz, H.-D., and Weiss, B. (2007). Phenomicdb: a new cross-species genotype/phenotype resource. *Nucleic Acids Res.*, 35(Database-Issue):696–699.
- [Hartigan, 1975] Hartigan, J. A. (1975). *Clustering Algorithms*. John Wiley & Sons, Inc., New York, NY, USA.
- [Hearst, 1992] Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th conference on Computational linguistics*, pages 539–545, Morristown, NJ, USA. Association for Computational Linguistics.
- [Krishna und Krishnapuram, 2001] Krishna, K. and Krishnapuram, R. (2001). A clustering algorithm for asymmetrically related data with applications to text mining. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 571–573, New York, NY, USA. ACM Press.
- [Krishnapuram und Kummamuru, 2003] Krishnapuram, R. and Kummamuru, K. (2003). Automatic taxonomy generation: Issues and possibilities. In Bilgiç, T., Baets, B. D., and Kaynak, O., editors, *IFSA*, volume 2715 of *Lecture Notes in Computer Science*, pages 52–63. Springer.
- [Kummamuru et al., 2004] Kummamuru, K., Lotlikar, R., Roy, S., Singal, K., and Krishnapuram, R. (2004). A hierarchical monothetic document clustering algorithm for summarization and browsing search results. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 658–665, New York, NY, USA. ACM Press.
- [Lawrie und Croft, 2000] Lawrie, D. and Croft, W. (2000). Discovering and comparing topic hierarchies. In *Proceedings of RIAO*.
- [Lawrie und Croft, 2003a] Lawrie, D. and Croft, W. (2003a). Generating hierarchical summaries for web searches.
- [Lawrie et al., 2001] Lawrie, D., Croft, W. B., and Rosenberg, A. (2001). Finding topic words for hierarchical summarization. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 349–357, New York, NY, USA. ACM Press.
- [Lawrie und Croft, 2003b] Lawrie, D. J. and Croft, W. B. (2003b). Generating hierarchical summaries for web searches. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 457–458, New York, NY, USA. ACM Press.
- [Maedche und Staab, 2000] Maedche, A. and Staab, S. (2000). Discovering conceptual relations from text. In *ECAI-2000 - European Conference on Artificial Intelligence*. IOS Press, Amsterdam.
- [Maedche und Staab, 2002] Maedche, A. and Staab, S. (2002). Measuring similarity between ontologies. In *EKAW '02: Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management. Ontologies and the Semantic Web*, pages 251–263, London, UK. Springer-Verlag.
- [Maedche und Staab, 2004] Maedche, A. and Staab, S. (2004). *Ontology Learning*, chapter 9, pages 173–190. In [Staab und Studer, 2004].
- [Mandhani et al., 2003] Mandhani, B., Joshi, S., and Kummamuru, K. (2003). A matrix density based algorithm to hierarchically co-cluster documents and words. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 511–518, New York, NY, USA. ACM Press.
- [Mani et al., 2004] Mani, I., Samuel, K., Concepcion, K., and Vogel, D. (2004). Automatically inducing ontologies from corpora. In *Proceedings of CompuTerm 2004: 3rd International Workshop on Computational Terminology, COLING'2004, Geneva*.

- [McGuinness und van Harmelen, 2004] McGuinness, D. L. and van Harmelen, F. (2004). Owl web ontology language - overview, w3c recommendation. www.w3.org/TR/2004/REC-owl-features-20040210.
- [Necib und Freytag, 2005] Necib, C. B. and Freytag, J.-C. (2005). Semantic query transformation using ontologies. In *IDEAS '05: Proceedings of the 9th International Database Engineering & Application Symposium (IDEAS'05)*, pages 187–199, Washington, DC, USA. IEEE Computer Society.
- [Peat und Willett, 1991] Peat, H. J. and Willett, P. (1991). The limitations of term co-occurrence data for query expansion in document retrieval systems. *J. Am. Soc. Inf. Sci.*, 42(5):378–383.
- [Ryu und Choi, 2006] Ryu, P.-M. and Choi, K.-S. (2006). Determining the specificity of terms using inside-outside information: a necessary condition of term hierarchy mining. *Inf. Process. Lett.*, 100(2):76–82.
- [Salton, 1971] Salton, G. (1971). *The SMART Retrieval System: Experiments in Automatic Document Processing*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- [Salton et al., 1974] Salton, G., Yang, C. S., and Yu, C. T. (1974). A theory of term importance in automatic text analysis. Technical report, Ithaca, NY, USA.
- [Sanderson und Croft, 1999] Sanderson, M. and Croft, B. (1999). Deriving concept hierarchies from text. In *SIGIR '99: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 206–213, New York, NY, USA. ACM Press.
- [Sanderson und Lawrie, 2000] Sanderson, M. and Lawrie, D. (2000). *Building, Testing, and Applying Concept Hierarchies*, chapter 1, pages 3–34. Kluwer Academic Publishers.
- [Schutze und Pedersen, 1997] Schutze, H. and Pedersen, J. O. (1997). A cooccurrence-based thesaurus and two applications to information retrieval. *Inf. Process. Manage.*, 33(3):307–318.
- [Smith et al., 2005] Smith, C. L., Goldsmith, C. A., and Eppig, J. T. (2005). The mammalian phenotype ontology as a tool for annotating, analyzing and comparing phenotypic information. *Genome Biology*, 6(1).
- [Snow et al., 2005] Snow, R., Jurafsky, D., and Ng, A. Y. (2005). Learning syntactic patterns for automatic hypernym discovery. In Saul, L. K., Weiss, Y., and Bottou, L., editors, *Advances in Neural Information Processing Systems 17*, pages 1297–1304. MIT Press, Cambridge, MA.
- [Srikant und Agrawal, 1995] Srikant, R. and Agrawal, R. (1995). Mining generalized association rules. In Dayal, U., Gray, P. M. D., and Nishio, S., editors, *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases, September 11-15, 1995, Zurich, Switzerland*, pages 407–419. Morgan Kaufmann.
- [Staab und Studer, 2004] Staab, S. and Studer, R. (2004). *Handbook on Ontologies (International Handbooks on Information Systems)*. International Handbooks on Information Systems. Springer.
- [Walinski, 2004] Walinski, H. (2004). Studying gene function: Creating knockout mice. www.scq.ubc.ca/studying-gene-function-creating-knockout-mice. Review from the Science Creative Quarterly.
- [Xu und Croft, 1996] Xu, J. and Croft, W. B. (1996). Query expansion using local and global document analysis. In *SIGIR '96: Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 4–11, New York, NY, USA. ACM Press.
- [Zhao und Karypis, 2002] Zhao, Y. and Karypis, G. (2002). Evaluation of hierarchical clustering algorithms for document datasets. In *CIKM '02: Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524, New York, NY, USA. ACM.