

SEMANTIK IN RECOMMENDER SYSTEMEN

SINN ODER UNSINN?

EINE ANALYSE ANHAND PERSONALISIERTER EMPFEHLUNGSSYSTEME FÜR MULTIMEDIA-DATEN

DIPLOMARBEIT

ZUR ERLANGUNG DES GRADES EINES DIPLOM-INFORMATIKERS

28. DEZEMBER 2011

FREIE UNIVERSITÄT BERLIN
INSTITUT FÜR MATHEMATIK UND INFORMATIK
ARBEITSGRUPPE CORPORATE SEMANTIC WEB

Name	Marko Harasic	Gutachter	Prof. Dr. Adrian Paschke
Matrikelnummer	3579580		Prof. Dr. Elfriede Fehr
Adresse	Lauenburgerstr.44 12169 Berlin		
E-Mail	harasic@inf.fu-berlin.de		

APP STORE



TORNADOGUARD
FROM DROIDCODER2187

PLAYS A LOUD ALERT SOUND
WHEN THERE IS A TORNADO
WARNING FOR YOUR AREA.

RATING: ★★★★★
BASED ON 4 REVIEWS

USER REVIEWS:



★★★★★ GOOD UI!
MANY ALERT CHOICES.



★★★★★ RUNNING
GREAT, NO CRASHES



★★★★★ I LIKE HOW YOU
CAN SET MULTIPLE LOCATIONS



☆☆☆☆☆ APP DID NOT
WARN ME ABOUT TORNADO.

THE PROBLEM WITH
AVERAGING STAR RATINGS

Eidesstattliche Erklärung

Hiermit erkläre ich an Eides Statt, dass ich die vorliegende Arbeit selbständig und ohne unerlaubte fremde Hilfe angefertigt, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen und Hilfsmitteln wörtlich oder inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Ort, Datum, Unterschrift

Zusammenfassung

Personalisierungstechniken ermöglichen den Benutzern eine vereinfachte Suche nach Informationen und Produkten basierend auf ihren jeweiligen subjektiven Vorlieben. Weiterhin bieten sie gerade den Betreibern von *Online-Shops* eine Möglichkeit, Kunden direkt anzusprechen und ihnen maßgeschneiderte Lösungen anbieten zu können. Jedoch weisen die traditionellen Verfahren der *Recommender Systeme* in ihrem Betrieb Defizite auf. In den letzten Jahren wurde vermehrt eine Erweiterung der traditionellen Verfahren mit Technologien des *Semantic Webs* untersucht, wobei eine Verbesserung der Genauigkeit ihrer erzeugten Empfehlungen beobachtet werden konnte. Insbesondere kann durch den Einsatz *semantischer* Technologien die Abbildung der Benutzer und Elemente durch weiteres Wissen über die Bedeutung von Vorlieben und Eigenschaften verbessert werden. Jedoch wurde dies bisher in einer objektiven Evaluierung noch nicht untersucht. In der vorliegenden Arbeit wurde ein umfassender Überblick über die verschiedenen Verfahren sowie ihre jeweiligen Stärken und Schwächen geschaffen. Ferner wurde eine Klassifikation erarbeitet, um verschiedene Verfahren anhand des Grades ihrer verwendeten *Semantik* einzuteilen. Für jeden identifizierten Grad wurden ausgewählte Algorithmen prototypisch implementiert und diese anhand verschiedener im Betrieb eines *Recommender Systems* relevanter Kriterien quantitativ miteinander verglichen. Dabei wurde untersucht, inwieweit der Einsatz *semantischer* Technologien einen Vorteil gegenüber den traditionellen Verfahren ermöglicht.

Inhaltsverzeichnis

1	Einleitung	1
1.1	E-Commerce und Personalisierung	1
1.2	Motivation	4
1.3	Gliederung	5
2	Recommender Systeme	7
2.1	Grundtypen der Recommender	9
2.1.1	Collaborative Filtering	9
2.1.2	Content-Based Recommender	12
2.1.3	Demographic Recommender	14
2.1.4	Knowledge-Based Recommender	15
2.1.5	Vor- und Nachteile der Grundsysteme	17
2.2	Hybrid Recommender	17
2.3	Komponenten der Recommender	19
2.4	Zusammenfassung	23
3	Semantic Web	25
3.1	Technologien des Semantic Webs	26
3.1.1	Identifikatoren	27
3.1.2	Syntax und Serialisierung	28
3.1.3	Datenaustausch	29
3.1.4	Taxonomien	31
3.1.5	Ontologien	33
3.1.6	Abfragen	36
3.2	Semantic Recommender	38
3.3	Zusammenfassung	41
4	Semantik in Recommender Systemen	43
4.1	Benutzermodelle	44
4.1.1	Stereotype Modelle	44
4.1.2	Feature-basierte Modelle	45
4.1.3	Vor- und Nachteile der Modelle	47
4.2	Repräsentationen von Benutzern und Elementen	48

4.2.1	Bewertungsvektoren	48
4.2.2	Vector Space Modelle	50
4.2.3	Taxonomische Repräsentationen	53
4.2.4	Semantische Repräsentationen	57
4.3	Zusammenfassung	59
5	Algorithmen der Recommender	63
5.1	Formale Definition der Recommender	64
5.2	Algorithmen	65
5.2.1	Mean-Based Recommender	65
5.2.2	Collaborative Filtering	66
5.2.3	Content-Based Recommender	70
5.2.4	Taxonomic Recommender	72
5.2.5	Semantic Recommender	78
6	Evaluierung der Verfahren	81
6.1	Evaluierungsdaten	82
6.2	Implementierung	84
6.3	Evaluierungsmetriken	89
6.3.1	Genauigkeit	90
6.3.2	Performanz	93
6.3.3	Coverage	94
6.4	Ermittlung der Umgebungsvariablen	95
6.4.1	Collaborative Filtering	95
6.4.2	Content-Based Recommender	97
6.4.3	Taxonomic Recommender	99
6.4.4	Semantic Recommender	101
6.5	Evaluierung der Verfahren	102
6.5.1	Genauigkeit	103
6.5.2	Lernverhalten der Benutzerprofile	108
6.5.3	Lernverhalten der Verfahren	110
6.5.4	Skalierbarkeit	112
6.5.5	Coverage	114
6.6	Diskussion	116
7	Zusammenfassung und Ausblick	125
7.1	Zusammenfassung	125
7.2	Ausblick	126
	Literaturverzeichnis	129

Tabellenverzeichnis

2.1	Vorteile und Nachteile der verschiedenen Grundtypen der <i>Recommender Systeme</i>	18
2.2	Daten und Struktur der <i>Recommender</i> Grundtypen	20
3.1	Typen von <i>SPARQL</i> -Queries	37
4.1	<i>Semantik</i> in Benutzerprofilen und Elementrepräsentationen	60
4.2	Äquivalenzen zwischen den Repräsentationen und dem <i>Semantic Web</i>	62
6.1	Klassifikation eines einzelnen Vorschlages für einen Benutzer	91
6.2	Verteilung der Größen der <i>Synsets</i>	99
6.3	Parameter der Verfahren für ihre jeweils höchste Genauigkeit der Vorhersagen . .	103
6.4	Leistungsdaten des bei der Evaluierung eingesetzten Computers	112
6.5	Größe der Empfehlungsmenge zur Erzielung der höchsten <i>Precision</i>	114

Abbildungsverzeichnis

2.1	Der Prozess der <i>Adaption</i>	8
2.2	Architektur der <i>Recommender Systeme</i>	21
3.1	Der Semantic Web Cake	27
3.2	Beispiel eines <i>RDF</i> -Graphen	30
4.1	Beispiel einer Sammlung von Bewertungsvektoren (Bewertungsmatrix)	49
4.2	Beispiel eines <i>Vector Space Modell</i> basierten Benutzerprofils	51
4.3	Beispiel eines <i>taxonomischen</i> Benutzerprofils	56
4.4	Beispiel eines <i>semantischen</i> Benutzerprofils	58
6.1	Paket-Übersicht des Systems in <i>UML</i> -Notation	85
6.2	Schematische Darstellung der verwendeten <i>RDFS</i> -Modellierung	87
6.3	Screenshot der Web-Oberfläche des Frontends	89
6.4	Genauigkeit der Vorhersagen des <i>User to User Collaborative Filterings</i>	95
6.5	Genauigkeit der Vorhersagen des <i>Item to Item Collaborative Filterings</i>	96
6.6	Ausschnitt aus der Verteilung der Termhäufigkeiten	98
6.7	Genauigkeit der Vorhersagen des <i>Content-Based Recommenders</i>	98
6.8	Genauigkeit der Vorhersagen des <i>Synset</i> -basierten <i>Taxonomic Recommenders</i> .	100
6.9	Genauigkeit der Vorhersagen des <i>RDF</i> -basierten <i>Taxonomic Recommenders</i> . .	101
6.10	Genauigkeit der Vorhersagen des <i>Semantic Recommenders</i>	102
6.11	Genauigkeit der verschiedenen Algorithmen nach der <i>MAE</i> -Metrik	104
6.12	Anzahl der <i>False Positives</i> und <i>False Negatives</i> der Vorhersagen	105
6.13	<i>Precision</i> und <i>Recall</i> der verschiedenen Algorithmen	107
6.14	Genauigkeit der verschiedenen Algorithmen nach der <i>F1</i> -Metrik	108
6.15	Genauigkeit der Algorithmen in Abhängigkeit zu der Anzahl der Bewertungen . .	109
6.16	Genauigkeit der Algorithmen in Abhängigkeit zur Größe der Trainingsdaten . . .	111
6.17	Skalierbarkeit der Verfahren	113
6.18	Coverage der Verfahren	115

Verzeichnis der Quellcode-Beispiele

3.1	Beispiel einer <i>URI</i> nach dem <i>HTTP</i> -Schema	28
3.2	Auszug einer <i>XML</i> -Datei	29
3.3	Auszug einer <i>RDF-XML</i> -Datei	31
3.4	Auszug einer <i>FOAF</i> -Datei	32
3.5	Beispiel einer <i>SPARQL</i> -Query	37

Kapitel 1

Einleitung

Durch den steigenden Informationsüberfluss im *World Wide Web* ist es für die Benutzer zunehmend schwerer geworden, dort gewünschte Informationen zu finden. Neben der aufwändigen Suche innerhalb von Milliarden von Dokumenten, stellt selbst die an sich überschaubare Suche nach einem gewünschten Gegenstand innerhalb eines *Online-Shops* ein immer größer werdendes Problem für den Benutzer dar. Durch die stetig wachsende Anzahl an Einträgen wird von diesem ein höherer Zeitaufwand gefordert. So listet alleine der Online-Shop *Amazon.com*¹ in seinem Katalog bereits mehrere Millionen Gegenstände, unter denen der potentielle Kunde nun seine Suche durchführen und seinen gewünschten Gegenstand finden soll.

Heutzutage sortieren *Online-Shops* ihre gelisteten Produkte meist in einem einfachen Katalog nach ihrer Kategorie und überlassen es ihren Benutzern, die gewünschten Gegenstände aus der unüberschaubaren Masse an Einträgen selbstständig zu ermitteln. Inmitten dieser Menge an Daten verlieren die Benutzer oftmals die Übersicht, finden das Gewünschte nicht und verlassen nach einer gewissen Verweildauer den Shop. Überzeugt davon, dass ihre Bedürfnisse auch in Zukunft nicht befriedigt werden können, würden sie diesen voraussichtlich nicht mehr betreten, obwohl der gesuchte Gegenstand durchaus im Katalog zu finden gewesen wäre. In Zeiten, in denen die Konkurrenz lediglich einen Klick entfernt ist, ist dieser Umstand unbedingt zu vermeiden, weil dadurch reale Umsatzeinbußen für den einzelnen *Online-Shop* eintreten.

1.1 E-Commerce und Personalisierung

Mit der immer stärkeren Verbreitung von Internet-Anschlüssen in der Bevölkerung und der damit verbundenen Nutzung des *World Wide Webs* verstärkt sich auch die Bedeutung des *E-Commerce*. So wurde in der am Anfang des Jahres 2011 vom Bundesverband des Deutschen Versandhandels in Auftrag gegebenen TNS Infratest-Studie² ein Wachstum der Umsätze im *Online-Versandhandel* zwischen den Jahren 2006 und 2010 um über 80 % ermittelt. Für das Jahr 2011 wurde eine weitere Steigerung des Umsatzes um 2,8 Milliarden Euro prognostiziert.

Das Wachstum des Handels in *Online-Shops* lässt sich neben der zunehmenden Verbreitung und gesteigerten Nutzung des Internets durch Vorteile erklären, die *Online-Shops* gegenüber dem

¹<http://www.amazon.com> Zugriff: 22. Mai 2011

²<http://www.versandhandel.org/zahlen-und-fakten/allgemeines/> Zugriff: 14. November 2011

1.1. E-Commerce und Personalisierung

traditionellen *Offline*-Handel aufweisen. So haben im *World Wide Web* sowohl geografische als auch zeitliche Distanzen keinerlei Bedeutung mehr. Für die Kunden besteht durch die Anzahl an Geschäften ein große Auswahl und ein einfacher Zugriff auf eine immer weiter wachsende Menge an unterschiedlichen Produkten.

Allerdings stellt die neuartige Interaktion zwischen Käufern und Anbietern beim Handel im *World Wide Web* die Betreiber von *Online*-Shops auch vor neue Herausforderungen, die in der *Offline*-Welt nicht auftreten. Das klassische, auf der *Face-2-Face*-Kommunikation basierende, Konzept des Vertrauensaufbaus zwischen Kunden und Verkäufern kann wegen der Unpersönlichkeit des *World Wide Webs* nicht stattfinden [Goy et al., 2007]. Aufgrund des Fehlens menschlicher Verkäufer, welche die Vorlieben eines einzelnen Kunden über einen längeren Zeitraum kennen gelernt haben, werden allen Kunden immer nur exakt dieselben Dinge eines Kataloges angeboten (das *One-Size-Fits-All*-Problem). Zudem sind die Kunden in der immer größer werdenden Menge an Produkten auf sich alleine gestellt und finden in den Katalogen oftmals nicht das Gewünschte. In einem klassischen Geschäft könnte ihnen der Verkäufer anhand seiner Erfahrungen aus vielen durchgeführten Verkäufen das gewünschte Produkt vorschlagen.

„Wenn ich 2 Millionen Kunden hätte, so bräuchte ich auch 2 Millionen Geschäfte.“

Jeff Bezos, CEO Amazon.com

Mit diesem Satz beschrieb Jeff Bezos, CEO von Amazon.com, die Notwendigkeit und Arbeitsweise von *Personalisierungen* im Umfeld des *E-Commerces* (zitiert von [Schafer et al., 1999]) kurz und treffend. Nach [Pine, 1999] müssen Anbieter, um konkurrenzfähig zu bleiben, heutzutage vielfältige Produkte entwickeln, welche die Bedürfnisse einer Vielzahl von Kunden befriedigen und diese Menge an Kunden auch direkt ansprechen. Das klassische *One-Size-Fits-All* Konzept hat von daher im *World Wide Web* nur noch einen geringen Nutzen.

Eine persönliche Interaktion mit dem Kunden findet beim sogenannten *One-To-One-Marketing* statt. Dieses Verfahren versucht also die Unpersönlichkeit des Vorgangs des Einkaufs im *World Wide Web* zu beseitigen. Dazu werden den *Online*-Shops Techniken zur Verfügung gestellt, die es ermöglichen, jeden Kunden im Einzelnen zu betrachten. Dabei werden die Vorlieben, Wünsche und Ziele eines jeden Kunden individuell erfasst [Peppers and Rogers, 1997]. Unter Berücksichtigung dieser Punkte versuchen die im *E-Commerce* Umfeld verwendeten *Personalisierungstechniken*, die Beziehung, die in einem *Offline*-Geschäft zwischen einem Kunden und einem Verkäufer normalerweise auftritt, nachzubilden. Nach [Pine, 1999] ziehen Kunden *Online*-Shops, welche eine persönliche Interaktion anbieten, gegenüber solchen vor, die sich zum Kunden nur anonym verhalten.

In einer Studie von [Tam and Ho, 2003] gaben neun von zehn Kunden an, dass sie sich eine Art der *Personalisierung* wünschen. Diese Ergebnisse decken sich nahezu mit denen einer im Jahr 2005 durchgeführten Umfrage von ChoiceStream Inc., bei der 80 % der Kunden diesen Wunsch angaben [Goy et al., 2007]. In einer von Jupiter Communications durchgeführten Studie zur Untersuchung des Einflusses von *Personalisierungen* in *Online*-Shops konnte ermittelt werden, dass die Benutzer, denen die *Personalisierung* zur Verfügung stand, mehr als doppelt so

viele Produkte bezogen, wie diejenigen, die diese nicht hatten [Hof et al., 1998]. Weiterhin konnte beobachtet werden, dass die Zahl der zu Käufern werdenden Besucher in *Personalisierung* anbietenden *Online-Shops* signifikant höher ist, als in *Online-Shops* ohne *Personalisierung* [Fink and Kobsa, 2000].

In einer von [Gaudin, 2003] durchgeführten Umfrage über die wichtigsten Kriterien beim Einkauf in einem *Online-Shop*, gaben die Testpersonen zwar vorwiegend eine übersichtliche Navigation oder schnell ladende Seiten als ausschlaggebend an, jedoch waren 14 % der befragten Personen bereit, aufgrund der angebotenen *Personalisierung* in einem *Online-Shop* etwas zu kaufen. Auch diese Zahl zeigt den positiven Einfluss der *Personalisierung* auf die Kundenresonanz.

Alle aufgeführten Untersuchungen kommen zu dem Ergebnis, dass eine Form der *Personalisierung* von den Kunden gewünscht ist und einen Mehrwert für sie, und dadurch auch für den Betreiber der *Online-Shops*, darstellt. Dieser Mehrwert manifestiert sich durch eine erhöhte Zufriedenheit der Kunden, was zu einer erhöhten Anzahl an Einkäufen führt.

[Schafer et al., 2001] identifizierten drei Vorteile, welche *Personalisierungstechniken* bei ihrem Einsatz den Betreibern von *Online-Shops* bieten. Sie:

- *bauen Loyalität auf*: Die Kunden kommen wieder, da sie bemerken, dass sie bei mehrmaliger Benutzung zunehmend besser beraten werden.
- *machen aus Suchern Käufer*: Die Kunden sind eher bereit etwas zu kaufen, wenn es sich schnell finden lässt und den eigenen Vorlieben entspricht.
- *erhöhen Kreuzverkäufe*: Die Kunden kaufen zusätzliche Gegenstände, wenn ihnen weitere, zu den im Warenkorb vorhandenen Produkten passende Produkte, gezeigt werden.

Aufbau von Loyalität

Personalisierte Systeme lernen nach und nach die Interessen und Vorlieben des einzelnen Kunden anhand seiner Aktionen kennen und verbessern mit der Zeit ihre Vorhersagen. Kunden honorieren dieses Verhalten, indem sie solche Systeme immer wieder benutzen (*Loyalitätsaufbau*).

Insbesondere der Aufbau von *Loyalität* ist heutzutage ein wichtiger Punkt für die Betreiber von *Online-Shops*, da eine hohe Kundenbindung in Zeiten, in denen die Konkurrenz lediglich wenige Klicks entfernt ist, einen entscheidenden Faktor darstellt [Reichheld and Sasser, 1990]. Selbst wenn es einem Mitbewerber gelingen würde, ein System zu erstellen, welches die gleichen Fähigkeiten aufweist, müssten Kunden dort viel Arbeit und Zeit investieren, um das neue System mit dem anzulernen, was dem alten *Online-Shop* bereits bekannt ist [Pine, 1995; Schafer et al., 2001]. *Personalisierung* stellt somit eine Möglichkeit dar, die Loyalität der Kunden gegenüber einem Anbieter zu erhalten [Kazienko and Kolodziejcki, 2006].

Mehr Käufer aufgrund verbesserter Navigation

Eine Unterstützung in der Navigation, welche auf die Vorlieben eines einzelnen Benutzers abgestimmt ist, lässt diesen schneller an sein Ziel kommen und steigert dadurch seine subjektive Zufriedenheit [Kaplan et al., 1993; Höök, 1997]. Da die *Kundenzufriedenheit* einen Einfluss auf

1.2. Motivation

das Kaufverhalten hat³, steigert eine *personalisierte* und damit verbesserte Suche die Umsätze eines *Online-Shops*.

Mit den klassischen Suchverfahren in den *Online-Shops* lassen sich für einen einzelnen Benutzer Elemente, welche sich im sogenannten *Long-Tail*⁴ befinden und seinen Interessen und Vorlieben entsprechen, nur schwer in der großen Anzahl an Nischenprodukten aufspüren. Oftmals ist der Benutzer sich seiner Vorlieben nicht eindeutig genug bewusst und es stellt für ihn daher ein großes Problem dar, seine speziellen Interessen und Vorlieben in der Sprache der Oberfläche eines *Online-Shops* exakt zu formulieren. Unter Anwendung einer verbesserten und *personalisierten* Navigation hingegen lassen sich diese Produkte für den einzelnen Benutzer leicht auffindbar machen.

Erhöhung von Kreuzverkäufen

Wenn ein Kunde in einem *Offline-Geschäft* ein Produkt kaufen will, so steht ihm oft ein Verkäufer beratend zur Seite und schlägt ihm dabei Produkte vor, die zu dem aktuell gesuchten passen oder ihm einen zusätzlichen Nutzen geben. Dieser Verkauf von zusätzlichen Produkten wird *Kreuzverkauf* genannt. Im Katalog eines *Online-Shops* ist normalerweise jeder Kunde bei seiner Suche auf sich alleine gestellt und findet solche zusätzlichen Produkte deshalb nicht. Daher wird mit *Personalisierungstechniken* versucht, einen virtuellen Verkäufer zu schaffen [Schafer et al., 2001]. Dieses System vereint die Erfahrung von vielen Verkaufsvorgängen mit verschiedenen Kunden (*Wisdom of Crowds* bzw. *Weisheit der Massen*) und kann so einem einzelnen Kunden zusätzliche Produkte vorschlagen. Diese werden dann oft gemeinsam mit den Produkten, die sich aktuell im Warenkorb befinden, gekauft (wie beispielsweise beim „*Kunden, die diesen Artikel gekauft haben, kauften auch*“-Feature von Amazon.comTM).

1.2 Motivation

Aufgrund der verschiedenen, durch Einsatz von *Personalisierungstechniken* ermöglichten Vorteile, finden diese ein immer größer werdendes Interesse sowohl bei den Benutzern, als auch bei den Anbietern von Informationsportalen und *Online-Shops*. So kann durch *Personalisierung* die Kundenzufriedenheit gesteigert werden, was zu einer Erhöhung der Umsätze führt. Eine weit verbreitete Technik, um dies zu erreichen, ist die Verwendung von *personalisierten Empfehlungssystemen*, sogenannten *Recommendern*. Diese realisieren ein zum einzelnen Benutzer *adaptives* und damit *personalisiertes* System, indem sie seine Vorlieben durch eine Analyse seiner Aktionen erfassen und mit Hilfe von Beschreibungen der gespeicherten Elemente ein den Benutzer repräsentierendes Profil erzeugen. Anhand der im Profil enthaltenen Informationen können sie auf die Interessen des jeweiligen Benutzers abgestimmte Empfehlungen generieren.

Trotz vieler sich bietender Vorzüge, weisen die etablierten Verfahren Defizite in ihrer Arbeitsweise auf. Insbesondere benötigen sie aussagekräftige und exakt formulierte Elementrepräsen-

³es wurde ein solcher Zusammenhang zwischen dem American Customer Satisfaction Index (ACSI) und der Kundenzufriedenheit festgestellt [Vikas and Carly, 2010]

⁴Der *Long-Tail* bezeichnet die Produkte, welche aufgrund ihres Nischen Charakters zwar, im Gegensatz zu den wenigen Bestsellern, selten verkauft werden, jedoch aufgrund ihrer großen Anzahl in der Summe jedoch für den größten Teil des Umsatzes sorgen [Anderson, 2004].

tationen, da sich ohne diese ihre Bedeutungen und auch die Benutzerinteressen nicht eindeutig für die *Recommender Systeme* erschließen lassen. Aufgrund der immer größeren Verbreitung des *Semantic Webs* zur Auszeichnung von Wissen und die die dadurch ermöglichten Vorteile liegt es nahe, *semantische* Technologien auch im Kontext der *Recommender Systeme* zu verwenden. Speziell seit der Unterstützung der *E-Commerce* Ontologie *Good Relations*⁵ durch alle verbreiteten Suchmaschinen, finden *semantische* Technologien einen zunehmenden Einsatz im *E-Commerce*. Mittels formal definierter Vokabulare lassen sich die Bedeutungen (*Semantik*) von Elementen und Benutzerinteressen maschinenlesbar und eindeutig beschreiben. Dabei können auch weitere, im Betrieb eines *Recommenders* auftretende Probleme wie beispielsweise die Datenintegration gelöst werden.

Die Verfahren, welche *personalisierte Empfehlungssysteme* mit dem *Semantic Web* kombinieren, werden *Semantic Recommender* genannt. In der Literatur sind verschiedene Systeme beschrieben, welche durch den Einsatz *semantischer* Technologien bei der Generierung von Empfehlungen sowie der Modellierung von Benutzern und Elementen in ihrer jeweiligen Domäne große Verbesserungen zu den traditionellen *Recommender Systemen* erzielen. Die *semantischen* Systeme wurden bisher jedoch lediglich isoliert evaluiert unter Verwendung von Evaluierungsdaten, die nicht öffentlich verfügbar sind. Somit ist keine objektive Vergleichbarkeit mit den traditionellen Verfahren gegeben.

Ziel der vorliegenden Arbeit ist es, einen solchen objektiven Vergleich verschiedener, *semantische* Technologien verwendender *Recommender Systeme* sowohl untereinander als auch gegenüber traditionellen Verfahren unter dem Einsatz des jeweils identischen, öffentlich zugänglichen Datensatzes durchzuführen. Dazu wird zunächst der Grad der ermöglichten *Semantik* durch die verschiedenen Formate zur Abbildung von Benutzern und Elementen analysiert und dabei ein Schema erarbeitet, welches die Algorithmen klassifiziert. Anhand dieses Schemas wird dann bestimmt, inwieweit der Einsatz von *semantischen* Technologien, neben einer Verbesserung der Genauigkeit der Empfehlungen, weitere Vorteile im Bereich der *Recommender Systemen* bietet, wie beispielsweise bei der Beschreibung von Benutzern oder Elementen. Dazu werden für jeden identifizierten Grad der *Semantik* ausgesuchte Algorithmen implementiert, diese jeweils durch zusätzliche *Semantik* erweitert und dann anhand von etablierten, aus der *Recommender-Literatur* entnommenen Metriken miteinander quantitativ verglichen. So kann abschließend analysiert werden, inwieweit der Einsatz von *Semantik* in *Recommender Systemen* tatsächlich eine Verbesserung erzielt, sowohl in der Gesamtheit als auch in Bezug auf spezielle Problematiken.

1.3 Gliederung

Die vorliegende Arbeit ist wie folgt gegliedert. In Kapitel 2 werden die am weitesten verbreiteten Verfahren zur *Personalisierung* von Informationsportalen und Suchsystemen, die *Recommender Systeme*, detailliert beschrieben. Es werden die verschiedenen Techniken sowie deren jeweilige Stärken und Schwächen aufgezeigt. Eine Lösung, um die Schwächen der jeweiligen Systeme zu

⁵<http://www.heppnetz.de/ontologies/goodrelations/v1>; Zugriff 25. Mai 2011

1.3. Gliederung

umgehen stellen die *Hybrid Recommender* dar. Eine besondere Bedeutung kommt dabei *Hybrid Recommender* zu, die Domänenwissen zur Modellierung von Benutzern und Elementen beinhalten. Diese Modellierungen sowie das von ihnen verwendete Domänenwissen werden zunehmend mittels Techniken des *Semantic Webs* bereitgestellt. Im Kapitel 3 wird zunächst eine kurze Einführung in diese Techniken gegeben und anschließend werden Beispiele ihrer Anwendung im Forschungsumfeld der *Recommender Systeme* genannt. Unter Verwendung der Techniken des *Semantic Webs* können aussagekräftige Repräsentationen der Benutzer und Elemente erreicht werden.

Kapitel 4 stellt verschiedene Methoden zur Repräsentation der Benutzer und der vorzuschlagenden Elemente vor und liefert eine Klassifizierung dieser Methoden anhand des Grades ihrer *Semantik*. Aufgrund der sich unterscheidenden Strukturen der Repräsentationen kommen dabei jeweils auf ihre speziellen Eigenarten angepasste Algorithmen zum Einsatz. Für jedes einzelne der Formate wird in Kapitel 5 exemplarisch ein Verfahren vorgestellt. Eine detaillierte Evaluierung dieser Verfahren anhand verschiedener für den Betrieb von *Recommender Systemen* wichtigen Kriterien wird in Kapitel 6 durchgeführt. Abschließend werden die Ergebnisse der Arbeit diskutiert und in Kapitel 7 ein Ausblick auf die dabei identifizierten noch offenen Problemstellungen und möglichen Erweiterungen gegeben.

Kapitel 2

Recommender Systeme

Die *Personalisierung* eines Systems kann auf zwei verschiedene Arten realisiert werden: mittels *Adaptierbarkeit* oder *Adaptivität*. Während *adaptierbare* Systeme dem Benutzer erlauben, verschiedene Parameter wie beispielsweise die verwendete Sprache oder seine bevorzugte Kategorien im Katalog selbst anzugeben, versuchen die *adaptiven* Systeme eine *Personalisierung* selbstständig, durch die Analyse der Aktionen des Benutzers, ohne sein direktes Eingreifen in die Parameter des Systems zu erreichen [Oppermann, 1994].

Die am häufigsten verwendete Technik, um ein zum Benutzer *adaptives* System zu realisieren, sind die *Recommender Systeme* (im Folgenden kurz *Recommender* genannt). Seit ihrer erstmaligen Erwähnung in der Literatur Mitte der 90er Jahre durch [Resnick et al., 1994; Hill et al., 1995; Shardanand and Maes, 1995] konnten sie in verschiedenen Einsatzgebieten von informationsverarbeitenden Verfahren Vorteile gegenüber den traditionellen Methoden des *Information Retrievals (IR)* erzielen. Sie wurden von daher zu einem eigenständigen und immer populäreren Forschungsbereich. Heutzutage kommen *Recommender* in verschiedenen Bereichen des *World Wide Webs* zur Anwendung und bilden immer öfter den Kernbestandteil von Informationsportalen, bei denen die Benutzer eine Unterstützung beim Finden von Gegenständen oder Informationen benötigen. Sie helfen den Benutzern aktiv in ihrer Entscheidungsfindung, sei es durch Empfehlungen von Produkten in *Online-Shops* wie zum Beispiel bei Amazon.comTM [Linden et al., 2003], durch Hinweise auf Nachrichten in einer *Online-Zeitung*, die von Interesse sein könnten [Billsus et al., 2002] oder durch das Generieren von Vorschlägen bei der Wahl von Filmen in MovieLens [Miller et al., 2003]. Insbesondere durch die in Kapitel 1.1 aufgeführten Vorteile von *Personalisierungen* in den *Online-Shops*, finden *Recommender Systeme* im *E-Commerce* Umfeld mittlerweile ein stetig steigendes wirtschaftliches Interesse.

Recommender Systeme können zum einen als eine Ausprägung und Erweiterung der von [Salton, 1989] beschriebenen *Information Retrieval*-Verfahren betrachtet werden, zum anderen aber auch als ein *adaptives* System, welches sich an die subjektiven Vorlieben und Interessen der Benutzer anpasst. Solche *adaptiven* Systeme unterstützen den Benutzer bei seiner Suche innerhalb einer Elementmenge durch die Generierung von Vorschlägen, welche auf seine jeweiligen Bedürfnisse und Wünsche abgestimmt sind [Ha, 2002]. Durch eine auf den Vorlieben und Abneigungen des jeweiligen Benutzers basierende Reduzierung des Suchraums, wird dieser bei seiner Navigation durch eine große Menge an Einträgen aktiv geleitet. Dabei werden immer mehr Elemente

von geringem persönlichen Nutzen oder Interesse ausgeblendet, das System verhält sich *adaptiv* zum Benutzer.

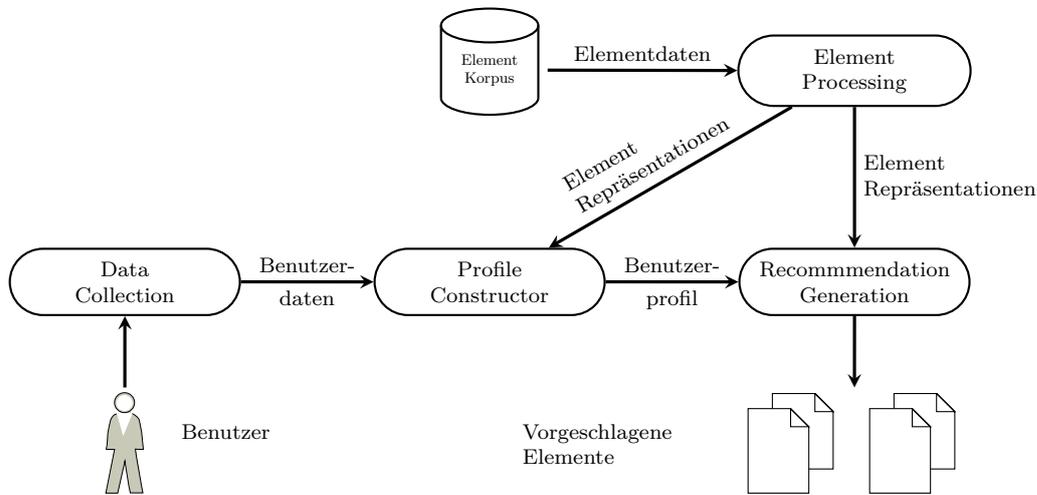


Abbildung 2.1 – Der Prozess der Adaption

[Brusilovsky, 1996] definiert drei Stufen im Prozess der *Adaption*, welcher in Abbildung 2.1 dargestellt ist. In den ersten beiden Stufen des Adaptionsprozesses werden Daten über den Benutzer gesammelt sowie die Elemente in das Repräsentationsformat des Systems überführt, was über eine Extraktion und Bestimmung ihrer Eigenschaften durch entsprechende Verfahren geschieht (siehe Kapitel 4). Anhand seiner Benutzerdaten sowie den Elementrepräsentationen wird ein den Benutzer repräsentierendes Profil basierend auf seinen subjektiven Interessen und Vorlieben gebildet. Im Allgemeinen lässt sich die Arbeitsweise der *Recommender* dadurch beschreiben, dass diese unter der Verwendung des Profils des aktuellen Benutzers versuchen, Bewertungen auf von ihm noch nicht betrachtete oder bewertete Elementen vorherzusagen. Nach der Berechnung dieser Vorhersagen werden ihm dann entweder die N -besten Elemente empfohlen oder aber nur das Element vorgeschlagen, welches die für ihn höchste errechnete Bewertung bezüglich der Bewertungsskala des Systems erzielt hat.

Die dabei erzeugten Benutzerprofile lassen sich in zwei Klassen aufteilen: in *kurzlebige* und in *persistente* Profile. Kurzlebige Profile basieren ausschließlich auf während einer einzelnen Sitzung gesammelter Daten, wie beispielsweise dem aktuellen Inhalt des Einkaufswagens oder dem Navigationsverhalten des Benutzers und bildet dabei die aktuellen Ziele ab. Diese können sich unter Umständen bei jedem Besuch unterscheiden.

Persistente Benutzerprofile hingegen basieren auf der Generierung einer aussagekräftigen und langlebigen Abbildung der Vorlieben und Interessen eines einzelnen Benutzers. Dies geschieht durch die Interpretation der über einen langen Zeitraum im System erfassten Aktionen des Benutzers sowie durch *explizite* Angaben seiner subjektiven Interessen. Mit zunehmender Verwendung passt sich dabei das Profil immer besser an den einzelnen Benutzer an. Insbesondere Bewertungen spielen eine große Rolle bei der Erfassung der Vorlieben eines Benutzers. Sie entsprechen subjektiven Aussagen über den persönlichen Nutzen eines Elementes. Diese Aus-

sagen werden entweder durch eine Bewertung des Elementes durch den Benutzer anhand der Bewertungsskala des Systems (*explizite* Bewertung) oder durch eine Analyse seines Verhaltens bei der Benutzung des Systems (*implizite* Bewertung) getroffen. Anhand dieser Informationen erzeugen die *Recommender* für jeden einzelnen Benutzer ein ihn repräsentierendes Profil. Eine detaillierte Beschreibung der Erfassung von verschiedenen Daten über die Benutzer findet sich in Kapitel 4.1.

2.1 Grundtypen der Recommender

Bei der Berechnung der vorzuschlagenden Elemente für die Benutzer kommen verschiedene Techniken zum Einsatz, welche sich durch die benötigten Eingaben sowie deren Verarbeitung grundlegend unterscheiden und auch jeweils andere Vor- und Nachteile aufweisen. Aus diesem Grund hat jede Technik andere Anwendungsbereiche, für die sie am Besten geeignet ist. Allen Techniken gemeinsam ist der Ansatz, ein den einzelnen Benutzer beschreibendes Profil zu erzeugen. Dies ist in den einzelnen Techniken dann aber unterschiedlich modelliert und ausgeprägt. Das Profil wird anhand *expliziter* Angaben⁶ oder auch anhand des Verhaltens⁷ des Benutzers erstellt und repräsentiert seine Interessen und Vorlieben im System.

[Adomavicius and Tuzhilin, 2005] stellte eine Klassifikation über die einzelnen Ansätze und deren vielfältige Implementierungen auf und schlug dabei mögliche Verbesserungen der aktuellen Verfahren vor. Hierbei identifizierte er die beiden Techniken *Collaborative Filtering* sowie *Content-Based Recommender*, welche heutzutage die höchste Verbreitung aufweisen.

[Burke, 2002] unterscheidet in seiner Arbeit über *Hybrid Recommender* nochmals differenzierter und nennt drei weitere grundlegende Techniken der *Recommender*, die *Demographic Recommender*, die *Knowledge-Based Recommender* sowie die *Utility-Based Recommender*. Im folgenden Teil werden diese Techniken genauer beschrieben, und es wird auf ihre jeweiligen Eigenschaften und die damit verbundenen spezifischen Vor- und Nachteile im Detail eingegangen. Die *Utility-Based Recommender* werden aufgrund ihrer nur geringen Verbreitung und einem eingeschränkten akademischen Interesse im Folgenden nicht beschrieben.

2.1.1 Collaborative Filtering

Collaborative Filtering (CF), erstmalig erwähnt in den Arbeiten von [Goldberg et al., 1992; Resnick et al., 1994; Hill et al., 1995; Shardanand and Maes, 1995], stellt heutzutage die am weitesten verbreitete Technik beim Einsatz von *Recommendern* dar. Dabei werden die Profile der Benutzer aus ihren *expliziten* bzw. *impliziten* Bewertungen von Elementen generiert und damit eine Abbildung ihrer Vorlieben und Abneigungen erzeugt. Die Grundannahme dieses Verfahrens basiert auf der Tatsache, dass ähnliche Personen meist auch ähnliche Vorlieben und Abneigungen haben. Vom aktuellen Benutzer noch unbewertete Elemente, welche andere zu ihm ähnliche Personen favorisieren, werden, dieser Annahme folgend, mit hoher Wahrscheinlichkeit auch von ihm bevorzugt werden. Anhand einer Nachbarschaftssuche, beispielsweise durch das wohl po-

⁶beispielsweise über Vorlieben oder durch subjektive Bewertungen auf Elemente

⁷beispielsweise Betrachtungsdauer der Elemente, ausgewählte und abgelehnte Elemente

2.1. Grundtypen der Recommender

pulärste Verfahren, den *k-nearest-Neighbour*-Algorithmus [Fix and Hodges, 1951], werden die zum aktuellen Benutzer ähnlichsten Personen bestimmt. Die Ähnlichkeit zwischen zwei Benutzern basiert hierbei entweder auf dem Cosinusmaß [Breese et al., 1998; Sarwar et al., 2001], welches den Cosinus des Winkels zwischen den als Vektoren aufgefassten Bewertungen zweier Benutzer bestimmt⁸, oder auf dem *Pearson*-Korrelationskoeffizienten [Pearson, 1895], welcher im korrelationsbasierten Ansatz verwendet wird [Resnick et al., 1994; Shardanand and Maes, 1995]. Sind die ähnlichsten Benutzer gefunden worden, werden unter Einflussnahme der von ihnen bisher bewerteten Elemente Vorhersagen für die Bewertungen der vom aktuellen Benutzer noch unbewerteten Elementen getroffen. So werden ihm dann die *N*-Elemente mit der höchsten errechneten Bewertung vorgeschlagen.

Die größte Stärke des *Collaborative Filtering* liegt in der Unabhängigkeit des Verfahrens von der Art und Struktur der vorliegenden zu empfehlenden Elemente. Somit brauchen die Elemente nicht durch aufwändige Verfahren im Vorfeld analysiert und vorbereitet werden, da dieser für Maschinen oftmals schwierige Teil von den Benutzern *explizit* bei der Betrachtung und Bewertung dieser Elemente übernommen wird. Somit lässt sich das Verfahren beispielsweise leicht für Multimedia-Daten wie Musik (*Ringo* [Shardanand and Maes, 1995]) oder Filme (*Video Recommender* [Hill et al., 1995]) anwenden.

Eine andere Stärke des Verfahrens ist die sogenannte *Serendipität*, welche das Verhalten bezeichnet, Vorschläge zu erzeugen, die an sich nicht nahe liegend sind. Dies kann nur durch das Betrachten von Vorlieben außerhalb der Sicht eines Einzelnen unter Einbeziehung der Weisheit der Massen (*Wisdom of the Crowd*) ermöglicht werden.

Ein einzelner Benutzer hat anfangs noch keinen direkten Nutzen aus seinen eigenen Bewertungen, so dass es schwer sein kann, genug Profilinformationen für qualitativ hochwertige Vorschläge zu erhalten. Ein *Collaborative Filtering*-System kann erst dann gute Vorschläge generieren, wenn eine gewisse Menge an Benutzerprofilen und durchgeführter Bewertungen von Elementen vorliegt. Diese Problematik in Bezug auf die Benutzer wird das *New User*-Problem genannt, da sie bei neuen Benutzern ohne eine gewisse Anzahl von ihnen erfasster Bewertungen auftritt [Claypool et al., 1999]. In Bezug auf die Elemente wird vom *New Item*-Problem [Rashid et al., 2002] bei neuen Elementen mit zu wenigen oder keinen auf sie abgegebenen Bewertungen gesprochen. Beide Problematiken werden unter dem Oberbegriff des *Cold-Start* Problems zusammen gefasst [Maltz and Ehrlich, 1995; Konstan et al., 1998]. Nichtsdestotrotz lernt der *Recommender* nach und nach die Vorlieben kennen, verfeinert die Profile der Benutzer und verbessert dadurch die Qualität der generierten Vorhersagen. Dies führt jedoch zu einem weiteren Problem, der *Plastizität vs. Stabilität*. Sobald sich ein Benutzerprofil „gefestigt“ hat (*Stabilität*), ist es schwer, die jeweiligen Vorlieben zu ändern. Wenn beispielsweise ein Benutzer nach jahrelanger Vorliebe für Steak-Restaurants beschließen sollte, Vegetarier zu werden, benötigt das System erst eine gewisse Menge an neuen Bewertungen, bis es die geänderten Präferenzen bemerkt und entsprechend auch in geänderte Vorschläge umsetzt. Aus diesem Grund wurde in einigen Systemen ein Dämpfungsfaktor eingeführt [Schwab et al., 2001; Billsus and Pazzani, 2000], der aber unweigerlich auch zu einer Verschlechterung der Ergebnisse führen kann, da lan-

⁸analog zu dem im *Information Retrieval* verwendeten Cosinusmaß [Salton et al., 1975]

ge nicht mehr verwendete Bewertungen mit der Zeit immer schwächer werden und schließlich kaum noch einen Einfluss haben (*Plastizität*).

Aufgrund des *New User*-Problems können Gelegenheitsnutzer nicht die kompletten Vorteile des *Recommenders* erfahren. Obwohl es möglich ist, einfache Vorhersagen zu treffen, die beispielsweise auf dem Inhalt des aktuellen Einkaufswagens des Benutzers basieren, entfalten *Recommender* dennoch erst dann ihr ganzes Potenzial, wenn der Benutzer einen gewissen Aufwand in die Pflege seines Profils investiert und damit dem System seine Vorlieben und Abneigungen verständlich gemacht hat.

Die unterschiedlichen Verfahren des *Collaborative Filterings* lassen sich ferner in eine Benutzer (*User to User*)- [Resnick et al., 1994] sowie in eine Element (*Item to Item*)-zentrierte [Sarwar et al., 2001; Linden et al., 2003] Technik einteilen. Die Element-zentrierten Ansätze vermeiden Skalierbarkeitsprobleme, welche bei den Benutzer-zentrierten Systemen aufgrund der großen Anzahl der Vergleichsoperationen, die bei großen Menge an Benutzer- und Elementen in der *Online*-Berechnung auftreten. Aus diesem Grund wurden Methoden, wie beispielsweise die bei der *Latent Semantic Analysis* zur Reduzierung der Matrixgröße benutzte Technik *Singular Value Decomposition* [Deerwester et al., 1990], verwendet werden, um den Rechenaufwand zu reduzieren [Billsus and Pazzani, 1998; Sarwar et al., 2000a; Sarwar et al., 2002]. Dennoch kann erst die Auslagerung der Ähnlichkeitsberechnungen in ein *Offline*-Verfahren, welches diesen rechen- und zeitintensiven Prozess aus der Echtzeit-Komponente entfernt, die Anwendung von *Collaborative Filtering* auf Kataloge mit mehreren Millionen Einträgen ermöglichen. Zudem zeigen Untersuchungen, dass die Element-basierten Algorithmen bessere Vorschläge generieren als die Benutzer-basierten Verfahren [Sarwar et al., 2001].

Unter Verwendung der *Singular Value Decomposition* lässt sich ein weiteres Problem, welches *Data-Sparsity* genannt wird, einschränken. *Data-Sparsity* tritt auf, wenn im System zu viele Elemente im Vergleich zu der Anzahl der Benutzer vorhanden sind. Es bezeichnet also das Vorhandensein von zu vielen Leerstellen in der Bewertungsmatrix, was in zu wenig Überlappungen bei den Bewertungen der Benutzer resultiert. Da die Funktionsweise des *Collaborative Filtering* zwingend diese Überlappungen voraussetzt, kann es unter Umständen unmöglich sein, genug Gemeinsamkeiten zu finden, um ähnliche andere Benutzer zu bestimmen. Somit fällt die Genauigkeit der Vorhersagen aufgrund mangelnder Übereinstimmungen. Unter der Verwendung der *Singular Value Decomposition* kann die Matrix komprimiert werden, wodurch diese Leerstellen teilweise gefüllt werden. Logischerweise tritt auch hier in der Anfangsphase (*Ramp-Up-Phase*) des Systems, wenn noch nicht genügend Benutzer Bewertungen abgegeben haben, das *Data-Sparsity*-Problem auf, es verschwindet aber nach einer ausreichenden Füllung der Matrix durch Bewertungen allmählich.

Viele *Online*-Shops, insbesondere der prominenteste Vertreter und Vorreiter Amazon.com⁹, nutzen bereits heute *Recommender Systeme*, wobei das *Item to Item Collaborative Filtering* von [Linden et al., 2003] von den meisten *Online*-Shops verwendet wird. In der wissenschaftlichen Welt ist MovieLens [Herlocker et al., 2000] das am meisten verbreitete System, da die Elemente sowie ihre dazugehörigen Bewertungen der Benutzer und auch die demografischen Nutzerdaten

⁹<http://www.amazon.com> Zugriff: 22. Mai 2011

2.1. Grundtypen der Recommender

öffentlich zugänglich sind¹⁰. Diese Daten und das *Collaborative Filtering* des MovieLens Algorithmus werden von vielen wissenschaftlichen Arbeiten im Bereich der *Recommender*-Forschung als Baseline-Verfahren bei der Evaluierung der eigenen Verfahren herangezogen.

2.1.2 Content-Based Recommender

Content-Based Recommender (CBR), erstmalig in den Systemen Newsweeder [Lang, 1995] sowie in Syskill and Webert [Pazzani et al., 1996; Pazzani and Billsus, 1997] erwähnt, haben ihre Wurzeln im Forschungsumfeld des *Information Filterings* [Belkin and Croft, 1987]. Sie generieren Vorschläge durch die Verwendung von zwei verschiedenen Informationsquellen. Zum einen analysieren sie die Elemente und extrahieren daraus die sie beschreibenden Eigenschaften, zum anderen erzeugen sie aus den Eigenschaften der von einem Benutzer bereits bewerteten Elemente ein individuelles Profil, bestehend aus seinen gewünschten Elementeigenschaften. Anhand dieses Profils werden dann mit Hilfe eines Ähnlichkeitsmaßes diejenigen noch unbetrachteten Elemente bestimmt, welche den angegebenen gewünschten Eigenschaften am ehesten entsprechen. [Schafer et al., 2001] bezeichnen dies als *item to item* Korrelation. Dadurch werden Vorschläge erzeugt, die zum bisher ermittelten Geschmack am ehesten passen, sowie diejenigen Elemente ausgeblendet, die den Vorlieben des Benutzers nicht entsprechen.

Das verwendete Ähnlichkeitsmaß vergleicht die Elemente auf Basis ihrer Eigenschaften, welche zum Beispiel aus den Beschreibungen von Gegenständen oder den Textbausteinen von Nachrichten und Büchern extrahiert werden [Lang, 1995; Pazzani et al., 1996; Balabanovic and Shoham, 1997; Mooney and Roy, 2000]. In solchen Systemen werden die Texte durch Techniken des *Information Retrievals* vor ihrer Verwendung analysiert und in das Repräsentationsformat des Systems überführt. Dabei werden sogenannten *Stopworte* (Worte, welche ohne Bedeutung für den Inhalt des Textes sind, wie „sie“, „und“, „sein“, ...) sowie *HTML*-Tags, falls es sich beispielsweise um den Quellcode einer Webseite handeln sollte, entfernt und die restlichen Worte unter Verwendung von *Stemming*-Algorithmen [Porter, 1980] auf ihren Wortstamm reduziert. Die extrahierten Terme werden dann mittels *Information Retrieval*-Techniken nach ihrer Relevanz für den Inhalt des Textes gewichtet, so zum Beispiel durch die weit verbreitete *TF-IDF*-Metrik [Salton, 1989]. Diese Eigenschafts-Vektoren, bestehend aus den gewichteten und aussagekräftigsten Worten eines Textes, repräsentieren dann das Element im System. Als Ähnlichkeit zwischen den Texten wird hierbei der Cosinus des Winkels zwischen den Vektoren aufgefaßt [Salton et al., 1975]. Zu beachten ist jedoch, dass sich diese Verfahren nur für aus Dokumenten einer einzigen Sprache bestehende Textkorpora eignen. Die Regeln des *Stemming*-Algorithmus sind sprachabhängig und auch die repräsentierenden Terme können sich in verschiedenen Sprachen unterscheiden, so dass beim Vorhandensein von Dokumenten in unterschiedlichen Sprachen keine Übereinstimmungen zwischen ihnen gefunden werden können.

Bedingt durch die Notwendigkeit einer Analyse der Elemente, kann ein *Content-Based Recommender* nur in solchen Anwendungsfällen zum Einsatz kommen, bei denen eine maschinellen Verarbeitung der Elemente leicht möglich ist. Während es mit dem eben beschriebenen Verfahren verhältnismässig einfach ist, einen Text zu verarbeiten, stellt dies für Multimediadaten wie Musik

¹⁰<http://www.grouplens.org/node/73> Zugriff: 15.Juli 2011

oder Filme, aufgrund der Komplexität der Analyse eine große Problematik dar. Von daher wird in diesen Domänen oftmals auf von Experten manuell ausgezeichnete Repräsentationen ausgewichen [Shardanand and Maes, 1995]. Auch wenn dadurch brauchbare Ergebnisse erzielt werden, ist das Verfahren wegen des hohen personellen Aufwands eher unpraktikabel. In diesem Fall kann die Verwendung von Daten, welche durch eine große Anzahl an Benutzern in einem offenen Prozess zur Auszeichnung der Elemente bereitgestellt wurden, eine Lösung darstellen [Szomszor et al., 2007].

Somit ist die größte Herausforderung bei der Realisierung eines *Content-Based Recommenders* das Finden einer möglichst aussagekräftigen Repräsentation der Elemente und eines darauf basierenden guten Ähnlichkeitsmaßes. Die vereinfachte Repräsentation der Elemente muss ferner eine gewisse Unterscheidbarkeit zwischen ihnen beinhalten, da Elemente, welche durch dieselben Eigenschafts-Vektoren beschrieben werden aber nicht identisch sind, nicht mehr vom *Recommender* zu differenzieren sind. Eng verbunden mit der Struktur der Repräsentation der Elemente ist auch die Repräsentation der Benutzerprofile, wobei letztere jedoch leicht als spezielle Ausprägung eines Elementes mit gleicher Struktur des beschreibenden Eigenschafts-Vektors aufgefasst werden können. In diesen können beispielsweise Vorlieben durch positive Werte und Abneigungen durch negative Werte für die jeweiligen Terme des Eigenschaft-Vektors repräsentiert werden.

Durch Analysen, welche die Qualität der Empfehlungen des *Collaborative Filterings* mit denen der *Content-Based Recommender* verglichen haben [Alspector et al., 1997], konnte herausgefunden werden, dass, selbst bei Vorhandensein von aussagekräftigen Beschreibungen für die Elemente, die Vorschläge vom *Collaborative Filtering* in den getesteten Domänen denen der *Content-Based Recommender* in ihrer Genauigkeit überlegen sind.

Nichtsdestotrotz haben die *Content-Based Recommender* den Vorteil, dass es aufgrund der vorhergehenden Analyse neuer Elemente unproblematisch ist, diese in das System zu integrieren. Direkt nach ihrer Verarbeitung stehen sie zur Verfügung und können den Benutzern vorgeschlagen werden. Aufgrund dieses Vorteils werden die *Content-Based Recommender* dort eingesetzt, wo die vorzuschlagenden Elemente nur eine kurze Lebensdauer aufweisen, wie beispielsweise bei Nachrichten in einer *Online-Zeitung*. Gerade wegen der Kurzlebigkeit der Elemente können auf diese nicht genug Bewertungen abgegeben werden, so dass diese unter Verwendung des *Collaborative Filterings* nicht vorgeschlagen werden können. Das auftretende *New Item-Problem* macht eine Anwendung dieses Verfahrens in einer Domäne mit einem sich stetig ändernden Elementkorpus nahezu unmöglich.

Sowohl bei den *Content-Based Recommendern* als auch beim *Collaborative Filtering* muss der einzelne Benutzer ausreichend viele Elemente bewertet haben, damit das System seine subjektiven Vorlieben und Interessen bestimmen kann. Damit weisen auch die *Content-Based Recommender* als lernende Systeme das in Kapitel 2.1.1 beschriebene *New User-Problem* auf. Sobald sich ein Profil „gefestigt“ hat, kann das Verfahren dann effektiv Vorschläge für den Benutzer erzeugen. Auch hier tritt das in Kapitel 2.1.1 beschriebene *Plastizitäts vs. Stabilitäts-Problem* auf, kann aber mit den gleichen, dort aufgeführten Ansätzen, wie beispielsweise einer zeitlichen Dämpfung der Bewertungen, gelöst werden.

2.1. Grundtypen der Recommender

Content-Based Recommender können aufgrund ihrer Arbeitsweise dem Benutzer nur die Elemente vorschlagen, welche anhand des verwendeten Ähnlichkeitsmaßes am Besten zu den bisherigen von ihm bewerteten Elementen passen. Von daher würden einem Benutzer niemals Elemente angezeigt werden, die zwar seinen Vorlieben und Interessen entsprechen, sich aber von den bisherigen betrachteten Elementen zu stark unterscheiden. Diese Problematik wird als *Überspezialisierung* bezeichnet, was dazu führt, dass dem Benutzer nur noch Elemente aus einem sehr eng gefasstem Bereich empfohlen werden.

Zusätzlich zu dieser Problematik kann es vorkommen, dass sich selbst die einzelnen Elemente der generierten Empfehlungsmenge untereinander zu sehr gleichen. Im Idealfall sollten dem Benutzer mehrere Auswahlmöglichkeiten gezeigt werden und nicht lediglich eine Menge von an sich identischen Alternativen. Beispielsweise filtert der Daily-Learner von [Billsus and Pazzani, 2000] Elemente aus der Ergebnismenge, welche sich untereinander zu stark ähneln. Die von [Smyth and McClave, 2001] vorgeschlagenen Algorithmen bauen die Ergebnismenge unter Beachtung eines Qualitätsmaßes nach und nach auf, wobei für jedes aufzunehmende Element zusätzlich neben der Ähnlichkeit gegenüber dem Benutzerprofil auch seine Unähnlichkeit gegenüber den Elementen der aktuellen Ergebnismenge erforderlich ist. In Untersuchungen konnte nachgewiesen werden, dass durch diese Techniken die subjektive Qualität der Vorhersagen gesteigert werden kann, ohne das Ergebnis dabei zu sehr zu verfälschen [Smyth and McClave, 2001; Bradley and Smyth, 2001].

2.1.3 Demographic Recommender

Demographic Recommender (DR) generieren Vorschläge anhand des demografischen Profils (z.B. Wohnort, Alter, Geschlecht, etc.) des Benutzers [Krulwich, 1997; Pazzani, 1999; Nguyen et al., 2007]. Diese Daten gibt dieser in der Regel bei seiner erstmaligen Anwendung an, sie werden im weiteren Verlauf der Nutzung des Systems jedoch selten von ihm aktualisiert.

Unter Verwendung der angegebenen Profilinformatoren wird die Teilmenge der Benutzer, welche die zum aktuellen Benutzer demografisch ähnlichsten anderen Benutzer beinhaltet, bestimmt. Diese Teilmenge wird als Nachbarschaft oder Cluster bezeichnet. Anhand der Bewertungen, welche die Personen des ermittelten Clusters über die Elemente getroffen haben, werden dann Vorschläge für den aktuellen Benutzer generiert [Krulwich, 1997]. Die Ermittlung der Ähnlichkeiten der Personen untereinander anhand ihrer demografischen Daten kann in einem *Offline*-Prozess durchgeführt werden, so dass die rechen- und dadurch zeitintensive Bestimmung der Nachbarschaften unabhängig von der Anfrage bereits im Vorfeld erfolgt, wodurch eine große Steigerung der Performanz erreicht werden kann. Im Gegensatz zu den klassischen *stereotypen* Modellen (siehe Kapitel 4.1.1) werden zwar die demografischen Cluster ebenfalls im Vorfeld definiert, die passenden Elemente werden hierbei jedoch dynamisch anhand der Bewertungen der Benutzer ermittelt und sind nicht einem Cluster statisch zugeordnet.

[Pazzani, 1999] verfolgte den Ansatz, die demografischen Daten anhand im Netz publizierter Informationen, wie z.B. der Homepages der Benutzer, mittels eines automatischen *Crawlers*¹¹ zu

¹¹Ein Crawler bezeichnet einen Prozess, welcher selbstständig im *World Wide Web* Webseiten besucht, diese speichert und mittels der daraus extrahierten Hyperlinks weitere Webseiten besucht.

aquirieren. Er verwendete dabei *Machine Learning*-Algorithmen, um einen Klassifikator für die Einteilung der Benutzer zu trainieren. Dieser erzeugt Vorschläge ähnlich dem zuvor genannten Verfahren, bei dem jeweils die Benutzer bestimmt werden, welche dem aktuellen Benutzer am Ähnlichsten sind.

Die Techniken der *Demographic Recommender* erzeugen eine *user to user* Korrelation, ähnlich der des *Collaborative Filterings*, ziehen dabei jedoch jeweils anderes Wissen in Form der demografischen Daten im Gegensatz zur Ähnlichkeit anhand abgegebener Bewertungen zur Generierung der Nachbarschaften der Benutzer heran. Von daher kann auch hier eine Unabhängigkeit von der Struktur der vorzuschlagenden Elemente sowie die in Kapitel 2.1.3 beschriebene *Serendipität* beobachtet werden. Der größte Vorteil der *Demographic Recommender* liegt allerdings darin, dass sie prinzipbedingt keine, wie von den zuvor beschriebenen Verfahren verlangte, Bewertungshistorie des Benutzers benötigen und somit das *New User*-Problem nicht auftritt.

Trotz der anscheinend großen Vorteile der *Demographic Recommender* hat es sich als große Problematik herausgestellt, dass zuerst die demografischen Daten für die Benutzerprofile zusammen getragen werden müssen, sei es durch *explizite* Angaben seitens der Benutzer selbst oder durch automatische Verfahren, mittels des bereits beschriebenen *Crawlers*. Mit der stetig zunehmenden Sensibilisierung der Bevölkerung bezüglich ihrer persönlichen Daten werden die Benutzer immer vorsichtiger, ihre persönlichen Daten im *World Wide Web* zu verbreiten. Dabei sind ausgerechnet die Angaben, welche die besten Vorhersagen ermöglichen würden, diejenigen, die von den Nutzern lieber „unter Verschluss“ gehalten werden.

Abschliessend lässt sich für die *Demographic Recommender* festhalten, dass sie die meisten Vor- und Nachteile des *Collaborative Filterings* teilen, jedoch aufgrund des auftretenden Datenschutzproblems Schwierigkeiten aufweisen, eine ähnlich große Verbreitung wie die zuvor beschriebenen Verfahren zu erreichen. Auch lassen sich die Vorlieben eines einzelnen Benutzers nur schwer anhand demografischer Daten wie beispielsweise seines Alters, Geschlechtes oder des Berufes bestimmen, so dass die Vorhersagen der *Demographic Recommender* nicht die Qualität der beiden vorher genannten Techniken erreichen. Nichtsdestotrotz können für beide Techniken demografische Daten herangezogen werden, um ein einfaches Profil für einen neu hinzugekommenen Benutzer zu initialisieren und damit das *New User*-Problem zu umgehen [Kurapati and Gutta, 2002].

2.1.4 Knowledge-Based Recommender

Die von [Burke, 2000; Towle and Quinn, 2000] erstmalig erwähnten *Knowledge-Based Recommender (KBR)* generieren ihre Vorhersagen, indem sie versuchen, aus den Vorlieben und Bedürfnissen eines Benutzer Schlüsse über gewünschte Eigenschaften der Elemente zu ziehen. Unter Verwendung einer Sammlung von Regeln, die beschreiben, inwieweit die Eigenschaft eines Elementes einer subjektiven Vorliebe oder einem Wunsch eines Benutzers entspricht, werden dann die für den aktuellen Benutzer vorzuschlagenden Elemente bestimmt. Diese Sammlung wird auch die *Wissensbasis* des *Recommenders* genannt.

Unter gewissen Einschränkungen kann postuliert werden, dass alle *Recommender Systeme* eine Art von Inferenz von Vorlieben des Benutzers auf Elemente durchführen. Allerdings haben nur

2.1. Grundtypen der Recommender

Knowledge-Based Recommender dieses Wissen in *expliziter* Form vorliegen. Unter Zuhilfenahme von *Meta-Daten* und zusätzlichen Informationen über die Domäne fließt bei diesem Verfahren *explizites* Wissen in die Generierung der Vorschläge ein. Dies ermöglicht eine mit einer Bedeutung behaftete Suche im Datenraum, welche über eine einfache Verwendung von *Schlüsselworten* hinaus geht. Die direkte Abhängigkeit von *explizitem* Domänenwissen ist jedoch gleichzeitig das schwerwiegendste Problem der *Knowledge-Based Recommender*, welches allerdings bei allen wissensbasierten Systemen auftritt. Das benötigte Wissen lässt sich nach [Burke, 2000] in drei Typen unterteilen:

- *Katalog-Wissen*: Wissen über die Elemente und deren Eigenschaften (z.B. sollte das System wissen, dass vegane Küche auch vegetarische Küche ist)
- *Funktionales Wissen*: Wissen darüber, inwieweit Eigenschaften ein Bedürfnis befriedigen können (der Wunsch nach einer romantischen Location kann durch ein ruhiges Restaurant mit Meerblick erfüllt werden)
- *Wissen über den Benutzer*: Wissen darüber, welche Bedürfnisse und Wünsche der Benutzer hat

Die ersten beiden Typen von Wissen müssen im Vorfeld von Domänenexperten in der Sprache des Systems modelliert werden. Von daher steht und fällt die Güte der Vorhersagen mit der Qualität der modellierten *Wissensbasis*. Dieses Wissen wird jedoch in das System „hart-codiert“ und ist von daher nur schwer erweiterbar. Somit ist ein auf diese Art und Weise realisiertes System normalerweise nur für einen festgelegten Fall nutzbar.

Das Wissen über die Bedürfnisse und Wünsche des Benutzers muss anderweitig erfasst werden und stellt die größte Herausforderung beim Design und Einsatz von *Knowledge-Based Recommendern* dar. Oftmals geschieht dies unter Verwendung einer *User-Feedback*-Schleife [Shimazu, 2002], bei welcher der Benutzer durch mehrmaliges Nachfragen den Suchraum immer weiter einschränkt oder erweitert, bis er auf die gewünschten Elemente trifft. Hierbei wird, im Gegensatz zu einer klassischen Suche, der Benutzer aktiv durch domänenspezifisches Wissen in der Navigation unterstützt. Er muss somit nicht eigenständig die Suchanfrage in der Query-Sprache des Systems formulieren oder gar selber Experte in der Domäne sein. Allerdings entsteht aufgrund der Verwendung einer *User-Feedback*-Schleife das *Stonewalling* genannte Problem [Smyth, 2007], welches die übrigen Systeme nicht aufweisen. Wenn ein Benutzer bei seiner Suche aufgrund eines zu stark eingeschränkten Suchraums eine leere Ergebnismenge bekommt, erkennt dieser nicht unbedingt das Filterkriterium, dessen Bereich er erweitern muss, um doch Ergebnisse zu erhalten. Eine spezielle Art der Ergebnissuche (*Order-Based Retrieval*), bei der nicht nur strikt nach den eingegebenen Werten der Suchfilter (*Filter-Based Retrieval*) gesucht wird, brachte eine elegante und praktikable Lösung für dieses Problem [Bridge, 2001]. Hierbei werden auch scheinbar unpassende Elemente, welche nicht alle Filterkriterien erfüllen, mit in die Ergebnisliste einsortiert.

Typbedingt vermeiden die *Knowledge-Based Recommender* viele der bisher aufgeführten Probleme, welche die übrigen *Recommender Systeme* aufweisen. So haben sie zum Beispiel nicht die

Cold Start-Problematik der lernenden Systeme, da die generierten Vorschläge nicht auf den bisherigen Bewertungen der übrigen Benutzer basieren, sondern die Vorschläge direkt aus den Wünschen und Bedürfnissen unter Verwendung der *User-Feedback*-Schleife abgeleitet werden. Allerdings erzeugt das System kein dauerhaftes Profil des einzelnen Benutzers, so dass dies für einen *Heavy-User*¹² einen Nachteil darstellen kann. Für einen Benutzer, der das System nur selten verwendet kann diese Tatsache von Vorteil sein, da dieser sofort Ergebnisse erhält und nicht erst sein Profil über einem längeren Zeitraum trainieren muss.

2.1.5 Vor- und Nachteile der Grundsysteme

Jeder einzelne zuvor beschriebene Grundtyp eines *Recommender Systems* weist aufgrund seiner jeweiligen Arbeitsweise unterschiedliche Stärken und Schwächen auf. Von daher kommt jeder einzelne Typ in Domänen zum Einsatz, in denen seine jeweiligen Vorteile zum Tragen kommen und die Nachteile weniger stark ins Gewicht fallen. So eignet sich beispielsweise *Collaborative Filtering* um schwer zu verarbeitende Elemente wie Filme vorzuschlagen, während *Content-Based Recommender* dort zum Einsatz kommen, wo die vorzuschlagenden Elemente nur eine kurze Lebensdauer haben wie beispielsweise in Nachrichtenportalen.

Aus der Aufstellung der Vor- und Nachteile der einzelnen Grundsysteme in Tabelle 2.1 wird ersichtlich, dass keines ohne Einschränkungen arbeitet und dadurch verschiedene Nachteile aufweist. Auch wenn sich bereits viele Arbeiten mit den einzelnen Problemen und deren Lösungen beschäftigt haben, konnte bisher keine die grundlegenden Typen der *Recommender Systeme* in einem solchen Maße verbessern, dass die aufgeführten Nachteile nicht mehr auftraten. Dies führte zu der Idee der Verbindung zweier oder mehrerer unterschiedlicher Systeme, um damit ein neues Gesamtsystem zu schaffen. Solche kombinierten *Recommender Systeme* werden *Hybrid Recommender* genannt.

2.2 Hybrid Recommender

Hybrid Recommender stellen, im Gegensatz zu den im Kapitel 2.1 aufgeführten Systemen, keine eigenständige *Recommender* Technik dar. Sie kombinieren vielmehr zwei oder mehrere der vorgestellten *Recommender* Techniken und bilden damit ein neues Gesamtsystem [Balabanovic and Shoham, 1997; Terveen et al., 1997; Basu et al., 1998; Good et al., 1999]. Im Idealfall entsteht dabei ein System, welches alle Stärken der benutzen Einzelsysteme vereint, ohne deren jeweiligen Einschränkungen zu übernehmen oder diese zumindest minimieren [Burke, 1999].

Für eine Kombination der Grundsysteme existieren verschiedene Möglichkeiten, die in der Arbeit von [Burke, 2007] identifiziert wurden. Die dort vorgestellten Ansätze zur Realisierung eines *Hybrid Recommenders* verbinden jeweils zwei der in Kapitel 2.1 vorgestellten Grundtypen zu einem neuen Gesamtsystem. Die einzelnen Verfahren wurden gegen ein Baseline-Systemen, basierend auf einem einzelnen *Collaborative Filtering Recommender*, evaluiert. Im Folgenden werden einige dieser Ansätze, exemplarisch nach aufsteigender Qualität ihrer Vorhersagen geordnet, vorgestellt.

¹² *Heavy-User* bezeichnet Benutzer, die ein bestimmtes System wiederholt und intensiv benutzen

2.2. Hybrid Recommender

Grundsysteme	Vorteil	Nachteil
Collaborative Filtering (2.1.1)	<ul style="list-style-type: none"> • Kann cross-genre Nischen identifizieren (<i>Serendipität</i>) • Kein Domänenwissen erforderlich • Adaptiv: Qualität der Vorhersagen verbessern sich mit steigender Nutzung • Implizites Feedback oft ausreichend 	<ul style="list-style-type: none"> • <i>New User</i> Problem • <i>New Item</i> Problem • Qualität der Vorhersagen abhängig von einem großen Datensatz von Bewertungen aller Benutzer • <i>Plastizitäts vs. Stabilitäts</i>-Problem • <i>Data-Sparsity</i>-Problem
Content-Based Recommender (2.1.2)	<ul style="list-style-type: none"> • Kein Domänenwissen erforderlich • Adaptiv: Qualität der Vorhersagen verbessern sich mit steigender Nutzung • Implizites Feedback oft ausreichend 	<ul style="list-style-type: none"> • <i>New User</i> Problem • <i>Plastizitäts vs. Stabilitäts</i>-Problem • Qualität der Vorhersagen für den Benutzer abhängig von seinen abgegebenen Bewertungen • <i>Überspezialisierungs</i>-Problematik
Demographic Recommender (2.1.3)	<ul style="list-style-type: none"> • Kann cross-genre Nischen identifizieren (<i>Serendipität</i>) • Kein Domänenwissen erforderlich • Adaptiv: Qualität der Vorhersagen verbessern sich mit steigender Nutzung 	<ul style="list-style-type: none"> • <i>New Item</i> Problem • Qualität der Vorhersagen abhängig von einem großen Datensatz von Bewertungen aller Benutzer • <i>Plastizitäts vs. Stabilitäts</i>-Problem • <i>Data-Sparsity</i>-Problem • Demografische Daten müssen ermittelt werden
Knowledge-Based Recommender (2.1.4)	<ul style="list-style-type: none"> • Kein <i>Cold-Start</i> Problem • Änderungen in den Vorlieben haben direkte Auswirkungen • Verwendung von Informationen über die direkten Elementeneigenschaften hinaus • Kann direkt von den Vorlieben auf die Elemente schließen 	<ul style="list-style-type: none"> • Vorschlagsgenerierung statisch (System lernt nicht) • <i>Stonewalling</i> aufgrund der <i>User-Feedback</i> Schleife • Modellierung von Domänenwissen erforderlich

Tabelle 2.1 – Vorteile und Nachteile der verschiedenen Grundtypen der Recommender Systeme

Switching und Mixed-Hybrid

Die einfachsten *Hybrid Recommender* Architekturen sind die *Switching* und *Mixed-Hybrid* Verfahren. Beim erstgenannten Verfahren wird immer einer der im System beinhalteten *Recommender* zufällig ausgewählt, dessen Ergebnisse dann alleinige Verwendung finden. Beim *Mixed-Hybrid* Verfahren hingegen werden die Ergebnisse jedes einzelnen *Recommenders* in die Ergebnismenge aufgenommen und aus diesen dann die N -besten Elemente bezüglich der Skala des Systems ausgewählt. Beide Verfahren bieten allerdings nur eine geringere Verbesserung gegenüber den einfachen *Recommendern*, da das Verhalten des Gesamtsystems sich nur wenig von dem der einzelnen Systeme unterscheidet.

Weighted-Hybrid

Bessere Ergebnisse als die bereits beschriebenen Systeme ermöglicht das *Weighted-Hybrid* Verfahren. Bei diesem errechnet jeder einzelne *Recommender* für die Elemente seine eigenen Vorhersagen. Diese Werte fließen dann als gewichtete Faktoren in das Endergebnis für die Vorhersage eines Elementes ein. Das Verfahren weist eine erste Verbesserung gegenüber den Einzelsystemen auf, da hier mehrere Kriterien für die Auswahl eines Elementes benutzt werden. Interessanterweise bieten die *Weighted-Hybride* Verfahren unter Verwendung von *Knowledge-Based Recommenders* eine Möglichkeit, die *Cold-Start*-Problematik zu lösen [Claypool et al., 1999]. Liegen noch nicht genug Bewertungen für einen Benutzer oder ein Element vor, können durch das Wissen des *Knowledge-Based Recommenders* immer noch hinreichend gute Ergebnisse generiert werden. Die größte Schwierigkeit beim Design eines *Weighted-Hybrid Recommenders* ist die Bestimmung der Gewichtungen der Teilsysteme. Diese Werte hängen stark vom konkreten Anwendungsfall ab [Kazienko and Kolodziejski, 2006].

Cascade und Feature Augmentation-Hybrid

Die besten Ergebnisse bezüglich der Qualität der Vorhersagen erreichten in der Evaluierung von [Burke, 2007] die *Cascade-Hybrid* sowie die *Feature Augmentation-Hybrid* Verfahren. In einem *Cascade-Hybrid* Verfahren werden die vorgeschlagenen Elemente des grundlegenden *Recommenders* als Ergebnismenge verwendet. Im nachfolgenden Prozess wird dann der nächste *Recommender* verwendet, um bei identischen Vorhersageergebnissen bezüglich der Skala des Systems die betroffenen Elemente entsprechend neu zu ordnen. Einen ähnlichen Weg geht das Verfahren des *Feature Augmentation-Hybrids*. Hier werden jedoch die vom ersten *Recommender* erzeugte Ergebnismenge sowie zusätzliche Informationen über die Elemente dieser Menge als Eingabe des nächsten *Recommenders* verwendet. Aufgrund dieser mehrstufigen Arbeitsweise fließt bei der Erzeugung der Vorschläge Wissen in den Prozess mit ein, das auf den verschiedenen Ansätzen der verwendeten *Recommender* beruht.

Semantic Recommender

Ein interessantes Ergebnis der Evaluierung ist die wiederholt auftretende Verbesserung der Vorhersagen, welche bei der Kombination eines *Recommenders* mit einem *Knowledge-Based Recommender* erzielt wird. Aus diesen *Hybrid Recommender* ging eine spezielle Klasse hervor, die *Semantic Recommender*. Diese Systeme verwenden eine *Wissensbasis*, die auf Konzepten und Methoden des *Semantic Webs* basiert. Die Techniken und Konzepte des *Semantic Webs* werden in Kapitel 3 vorgestellt und ihre Anwendungen innerhalb der *Semantic Recommender* in Kapitel 3.2 genauer betrachtet.

2.3 Komponenten der Recommender

Bei einer detaillierten Analyse in Hinblick auf die jeweiligen Komponenten der grundlegenden Techniken der vorgestellten *Recommender Systeme*, konnte bei allen vier *Recommendern*

2.3. Komponenten der Recommender

gemeinsame Komponenten identifiziert werden¹³, auch wenn die jeweiligen Ausführungen unterschiedlich in Bezug auf Design sowie Komplexität sind. Während beispielsweise die Komponente, welche in einem *Content-Based Recommender* die Elemente vorverarbeitet und speichert, entsprechend komplex realisiert werden muss, kann diese beim *Collaborative Filtering* oder einem *Demographic Recommender* viel einfacher gestaltet sein. Bei diesen beiden *Recommender*-Typen reicht sogar eine einfache Sammlung von Kennziffern zur Unterscheidung der Elemente für den Betrieb aus. Zusätzlich zu den für die Generierung der Vorschläge benötigten Informationen werden in den entsprechenden Speichersystemen für Elemente und Benutzer oftmals weitere Daten, wie zum Beispiel Bilder der Elemente oder Benutzernamen und deren Adressen gespeichert. Diese werden jedoch nur vom *Frontend* für die Darstellung einer ansprechenden Benutzeroberfläche benötigt und haben keinen Einfluss auf die Generierung der Ergebnisse.

Somit benötigen die verschiedenen Typen der *Recommender* jeweils eine unterschiedliche Menge an Informationen über die Benutzer sowie die Elemente und weisen damit auch ein jeweils anderes internes Repräsentationsformat auf. Während das eine Verfahren eine aussagekräftige Repräsentation erzeugt, verwendet ein anderes dieselben Informationen lediglich zur Darstellung der Elemente im *Frontend*. Tabelle 2.2 zeigt die von den jeweiligen Grundsystemen verwendeten Daten sowie die Formate der Benutzerprofile und Elementrepräsentationen.

Recommender Technik	Benutzerprofil	Elementrepräsentation	zusätzliche Daten
Collaborative Filtering (2.1.1)	Bewertungsvektor des Benutzers	Element ID	Bewertungen aller anderen Benutzer
Content-Based (2.1.2)	Erfasste Vorlieben des Benutzers anhand seiner Elementbewertungen	Element ID sowie Eigenschaften des Elementes	explizite Angaben des Benutzers
Demographic (2.1.3)	Demografische Daten des Benutzers	Element ID	Bewertungen aller anderen Benutzer der demografischen Nachbarschaft
Knowledge-Based (2.1.4)	Interpretation der Vorlieben anhand der Navigation des Benutzers	Eigenschaften des Elementes	Wissensmodell mit den Bedeutungen der Elementeeigenschaften sowie weiteren Regeln

Tabelle 2.2 – *Verwendete Daten und ihre Struktur in verschiedenen Recommender Techniken*

Die einzelnen Systeme benötigen zumindest eine eindeutige Kennzeichnung um Benutzer beziehungsweise Elemente jeweils voneinander unterscheiden zu können. Eine Sonderstellung nimmt hierbei der *Knowledge-Based Recommender* ein, da dieser normalerweise seine Vorschläge unter Verwendung von *User-Feedback*-Schleifen generiert [Burke, 2000] und keine persistente Benutzerdaten oder Bewertungen benötigt. Jedoch kann er sich die Bewertungen sowie die

¹³*Hybride Recommender* stellen keine eigenständige Technik dar, da sie auf den vier Grundtypen basieren und aus deren Komponenten bestehen.

Kennzeichnung des aktuellen Benutzers merken, um zu einem späteren Zeitpunkt die Suche des Benutzers von einem ihm bereits bekannten und positiv bewerteten Element aus beginnen zu lassen.

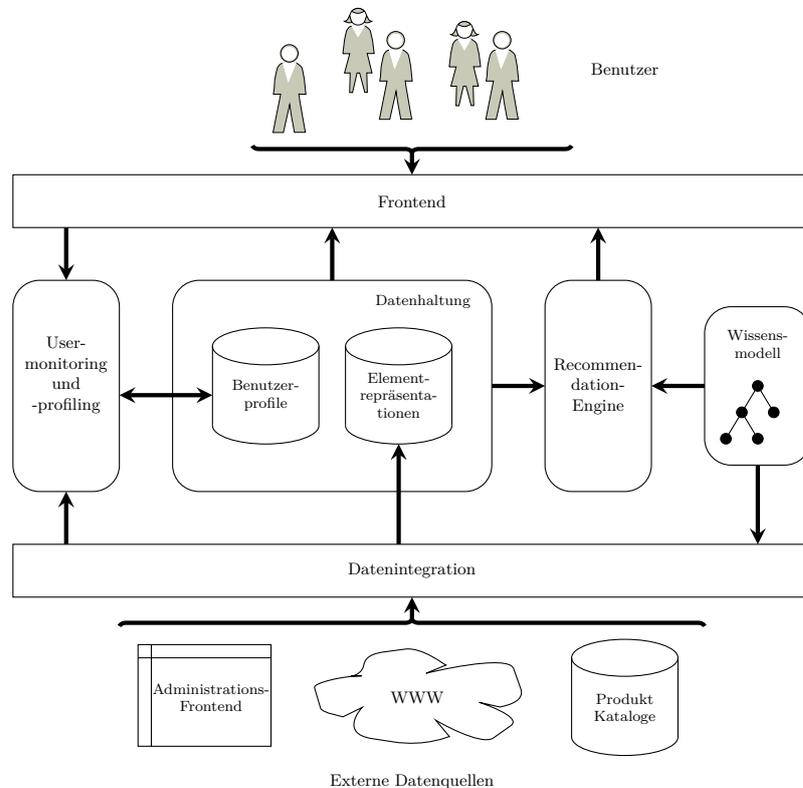


Abbildung 2.2 – Architektur der Recommender Systeme

Abbildung 2.2 zeigt die generische Architektur eines *Recommender Systems*. Neben den aufgeführten Komponenten ist auch ihr Zusammenspiel untereinander dargestellt. Dabei entsprechen die Richtungen der Pfeile den Richtungen des Informationsflusses von und zu den einzelnen Komponenten. Jede einzelne hat dabei eine tragende Rolle im Gesamtsystem. Wie zuvor beschrieben, beinhalten die *Knowledge-Based Recommender* aufgrund ihrer Sonderstellung durch die Verwendung von *User-Feedback*-Schleifen nur eine Teilmenge der Komponenten. Im Folgenden sind diese Komponenten sowie ihre jeweiligen Aufgaben aufgeführt.

Datenhaltung

Diese Komponente dient dem persistenten Speichern der Repräsentationen von Benutzern sowie Elementen und lässt sich weiter in zwei Teilkomponenten aufteilen, welche jeweils Zugriff auf die von ihnen gespeicherten Repräsentationen bieten.

- **Benutzerprofile:** Diese Komponente speichert alle Profile der Benutzer und stellt sie der *Recommendation-Engine* sowie dem *Frontend* zur Verfügung. Benutzerprofile bestehen zumindest aus einer Identifikation, um die Benutzer unterschieden zu können, und beinhalten meist weitere Daten wie Adressen, Geburtstage oder ausgeübte Berufe sowie *explizite*

2.3. Komponenten der Recommender

Angaben zu Vorlieben und Wünschen. Sie dienen als Repräsentation eines Benutzers im System. Neben den genannten Informationen werden innerhalb der Benutzerprofile auch die von den Benutzern abgegebenen Bewertungen auf Elemente gespeichert, beziehungsweise die daraus abgeleiteten Vorlieben und Interessen. Weiterhin können Informationen aus der vom Benutzer erfolgten Aktionen erfasst werden.

- **Elementrepräsentationen:** Diese Komponente beinhaltet zumindest eine Form der Identifikation der Elemente, sowie weitere Daten, die ihrer Darstellung im *Frontend* dienen. Je nach Verfahren werden diese weiteren Daten auch bei der Generierung der Vorschläge für die Benutzer verwendet. Dabei sind die Elemente in einer unterschiedlich stark strukturierten Form beschrieben.

Wissensmodell

Das *Wissensmodell* beinhaltet Informationen und Regeln, wie die Eigenschaften der Elemente sowie die Vorlieben der Benutzer bei der Vorschlagsgenerierung zu verarbeiten sind. In einer *expliziten* Form liegt das verwendete Wissen nur bei den *Knowledge-Based Recommendern* vor, die anderen Typen leiten es aus den Benutzerprofilen und Elementrepräsentationen ab. Technisch gesehen liegt das *Wissensmodell* je nach Systemarchitektur entweder in Form von Algorithmen oder Regeln in der *Recommendation-Engine* vor, oder aber als ein in einer Repräsentationssprache beschriebenes austauschbares Modell in der *Datenhaltungs-Komponente*. Dabei ist es jedoch logisch gesehen ein eigenständiger Bestandteil eines *Recommender Systems*.

Recommendation-Engine

Diese Komponente beinhaltet den eigentlichen Algorithmus des *Recommenders* und stellt die Kernkomponente des Systems dar. Die *Recommendation-Engine* greift lesend auf die *Datenhaltungs-Komponente* zu, um anhand der Benutzerprofile sowie der Elementrepräsentationen unter Verwendung des *Wissensmodells* Vorschläge für den Benutzer zu generieren. Dabei benötigen die unterschiedlichen Grundtypen verschiedene Daten (vgl. Tabelle 2.2) für die Erzeugung der Vorschläge.

Frontend

Diese Komponente stellt den dem Benutzer sichtbaren Teil dar. Die Benutzer können ihre *expliziten* Angaben über Vorlieben oder *explizite* Bewertungen auf ihnen angezeigte Elemente in der Oberfläche des *Frontends* eingeben. Weiterhin werden in ihr auch die Aktionen der Benutzer bei ihrer Verwendung des Systems erfasst. Sie visualisiert die Elemente, sowie die durch die *Recommendation-Engine* generierten Vorschläge. Weiterhin bietet das *Frontend* eine Navigation an, die es den Benutzern ermöglicht, sich in der Elementmenge zu bewegen.

Usermonitoring und -profiling

Die *Usermonitoring und -profiling-Komponente* hat in allen Grundtypen der *Recommender* eine besondere Bedeutung, da sie für die Erzeugung und Aktualisierung der Benutzerprofile zuständig ist und in ihrem Design in starkem Maße vom eingesetzten Verfahren der *Recommendation-Engine* abhängt. Das Verfahren der *Recommendation-Engine* gibt auch das

Format der Benutzerprofile vor. Um diese zu generieren, sammelt die *Usermonitoring und -profiling*-Komponente im *Frontend* die *expliziten* Elementbewertungen der Benutzer oder interpretiert aus ihren Aktionen die jeweiligen *impliziten* Elementbewertungen. Diese Daten werden benutzt, um das bereits bestehende Profil des Benutzers zu aktualisieren und somit ein gegenwärtiges Abbild seiner Vorlieben und Interessen zu erzeugen und in der *Datenhaltungs*-Komponente abzulegen. Zusätzliche Daten über den Benutzer kann die *Usermonitoring und -profiling*-Komponente auch durch die *Datenintegrations*-Komponente erhalten.

Datenintegration

Diese Komponente integriert neue Elemente oder aktualisiert bereits bestehende Daten. Dabei stammen die Informationen aus externen Quellen wie beispielsweise aus Eingaben in ein Administrationsinterface oder auch aus Informationen, welche mittels eines *Crawlers* im *World Wide Web* erfasst wurden. Dabei kann die *Datenintegrations*-Komponente auch die im *World Wide Web* gesammelten Daten über die Benutzer an das *Usermonitoring und -profiling* weiterreichen, welches damit die Benutzerprofile aktualisiert. Des Weiteren dient die *Datenintegrations*-Komponente auch dem Übersetzen der Informationen in das Repräsentationsformat der Elemente und ihrer Integration in das System, falls sie nicht in den Formaten des Systems vorliegen. Dazu verwendet die *Datenintegrations*-Komponente Informationen, die vom *Wissensmodell* zur Verfügung gestellt werden.

2.4 Zusammenfassung

Wie in Kapitel 2.2 beschrieben, können mit *Hybrid Recommendern* Systeme realisiert werden, welche die Probleme der traditionellen Verfahren umgehen und dabei die Qualität der gemachten Vorschläge erhöhen. Insbesondere die detaillierte Evaluierung von [Burke, 2007] bezüglich der *Hybrid Recommender* zeigt ihre Vorzüge gegenüber dem Baseline-Verfahren, welches auf einem einzelnen System basiert. Auch wenn der Ansatz des *Hybrid Recommenders* viele der bereits beschriebenen Probleme lösen oder zumindest abschwächen kann, besteht immer noch die Problematik des *Cold-Starts*, welche bei den meisten lernenden Systemen auftritt. Liegt jedoch bereits eine *Wissensbasis* für die Domäne vor, kann auch mit dem anfangs begrenzten Wissen über die Vorlieben des Benutzers auf erwünschte Eigenschaften für weitere Elemente geschlossen werden. Weiterhin ermöglicht die *Wissensbasis* die Bestimmung von ähnlichen Elementen anhand der Bedeutung ihrer Eigenschaften. Dies ist mit der alleinigen Verwendung der Bewertungshistorie des Benutzers oder eines auf einfachen Vergleichen der Elementeigenschaften basierenden Verfahrens nicht möglich. Die gewünschten Fähigkeiten stellen die in Kapitel 2.1.4 bereits vorgestellten *Knowledge-Based Recommender* bereit, welche ihre Vorhersagen anhand ihrer im System integrierten *Wissensbasis* treffen. Zwar passen sich die meisten der *Knowledge-Based Recommender* dem aktuellen Benutzer an, sie erfassen seine Vorlieben aber nicht dauerhaft. Somit liegt es nahe, einen *Hybrid Recommender* bestehend aus einem *Knowledge-Based Recommender* und einem oder sogar mehreren beliebigen anderen *Recommendern* zu verwenden. Beispielsweise weist ein *Hybrid Recommender*, welcher als zusätzliches System *Collaborative Filtering* verwendet, den *Serendipitäts*-Effekt aufgrund des *Collaborative Filterings* auf und vermeidet gleichzeitig

2.4. Zusammenfassung

dessen *Cold-Start*-Problem mit Hilfe des vorhandenen Domänenwissens des *Knowledge-Based Recommenders*. Nach [Burke, 1999] stellt dies den idealen *Hybrid Recommender* dar. Allerdings muss die *Wissensbasis* des *Knowledge-Based Recommenders* von Experten im Vorfeld modelliert werden.

Gerade wegen der Abhängigkeit von Domänenwissen weisen die *Knowledge-Based Recommender* das in der Anwendungsentwicklung oftmals anzutreffende Problem auf, dass beim Design und der Implementierung von Applikationen normalerweise keine Domänenexperten hinzugezogen werden. Das Wissen, das sie zur Verfügung stellen, kann oft nur schwerlich in den Entstehungsprozess miteinbezogen werden, und selbst wenn dies geschieht, so ist diese Art des Modellierung von Wissen über die Domäne starr und im Nachhinein nur schwer wart- oder erweiterbar. Aus diesem Grund wird eine Methodik benötigt, die dieses Wissen besser modellierbar und insbesondere austauschbar macht. Mit den Konzepten und Techniken des *Semantic Webs* existiert ein mächtiges Werkzeug, um Domänenwissen zu beschreiben und für Maschinen eindeutig interpretierbar zu machen, was in der Vergangenheit bereits in verschiedenen Domänen und Systemen gezeigt werden konnte. Für einige Domänen existieren bereits vorgefertigte und etablierte Modellierungen, so dass die ansonsten zwar notwendige, dabei aber aufwändige und oftmals fehlerbehaftete Modellierungsarbeit entfallen kann.

Ein weiterer Nutzen des *Semantic Webs* liegt in der Typisierung und eindeutigen Auszeichnung der Daten. Dadurch können die Benutzer sowie die Elemente für die Maschinen verständlich modelliert werden. Somit brauchen die Elemente beispielsweise nur ein einziges mal von ihrem Hersteller oder Anbieter beschrieben werden und können dann von verschiedenen Systemen direkt verwendet werden. Liegen entsprechende Mappings zwischen den Elementbeschreibungen sowie dem eigenen Repräsentationsformat vor, so können neue Elemente mit nur geringem Mehraufwand in das eigene System integriert werden und liegen sofort zugreifbar bereit.

Im folgenden Kapitel werden die Techniken und Konzepte des *Semantic Webs* vorgestellt und einige Beispiele genannt, in denen sie bereits *Recommender Systeme* in ihrer Arbeitsweise unterstützen. Diese Systeme erzeugen Vorhersagen von hoher Güte und können die Anzahl der fehlerhaften Vorhersagen deutlich reduzieren. Weiterhin stellen sie Lösungen für die klassischen Probleme der *Überspezialisierung*, des *Cold-Starts* sowie der *Data-Sparsity* dar.

Kapitel 3

Semantic Web

Die Idee des *Semantic Webs* basiert auf einem Vorschlag von Sir Tim Berners-Lee [Berners-Lee et al., 2001], um das klassische, aus Dokumenten bestehende *World Wide Web* um eine Methode zur Auszeichnung von Wissen zu erweitern. Unter Verwendung der Techniken und Methoden des *Semantic Webs* sollen Menschen beim Suchen von Informationen in der steigenden Menge der Daten durch Maschinen unterstützt werden. Weiterhin soll das Verbreiten und Verknüpfen von Informationen und Wissen mit Hilfe der maschinellen Verarbeitung vereinfacht werden. Während es für einen Menschen keinerlei Schwierigkeiten bedeutet, auf einer Webseite die Bedeutung von Informationen, wie zum Beispiel einen Namen oder eine Anschrift zu erkennen oder gar Inhalte von Bildern zu erfassen, können Maschinen diese Informationen nur schwer erschließen. Die genaue Bedeutung (*Semantik*) von Textbausteinen oder Bildern innerhalb eines Dokumentes kann maschinell, falls überhaupt möglich, nur unter größtem Aufwand bestimmt werden. Selbst mit fortschrittlichen Methoden aus der Computerlinguistik bzw. der Bildverarbeitung ist die Extraktion oftmals mit Fehlern behaftet, was im Hinblick auf die aktuelle und weiter wachsende Anzahl von mehreren Milliarden von Dokumenten im *World Wide Web* ein nicht unerhebliches Problem darstellt. Insbesondere Multimedia-Daten wie Bilder, Musikstücke oder Filme lassen sich in ihrer Bedeutung nur schwer erfassen. Dies erschwert die weitere maschinelle Verarbeitung von Informationen, welche zur Unterstützung der Menschen aber erstrebenswert wäre. Mit Hilfe von Technologien des *Semantic Web* können Daten mit einer *expliziten* Bedeutung ausgezeichnet werden, wodurch eine unproblematische weitergehende maschinelle Verarbeitung ermöglicht wird.

Unter Verwendung von *Semantic Web*-Technologien werden Informationen in Form von ausgezeichneten Konzepten angegeben. Ganz im Gegensatz zum üblichen Format von Webseiten, deren eigentliche Informationen nur aus unstrukturierten Textelementen bestehen¹⁴, werden diese Informationen unter Verwendung spezieller Vokabulare formal typisiert (z.B. als Zahl, Zeichenkette oder Konzept ausgezeichnet). Jedes einzelne Konzept, sowie die Werte seiner Attribute, falls diese wieder auf andere Konzepte verweisen, lassen sich mit Hilfe der Verwendung von global eindeutigen Identifikatoren referenzieren. Somit können die Daten leicht verteilt, eindeutig miteinander verknüpft und wiederverwendet werden. Erst durch die Verknüpfung der Daten un-

¹⁴die vorhandenen *HTML*-Tags dienen zur ansprechenden Darstellung des Inhalts für Menschen

3.1. Technologien des Semantic Webs

tereinander steigert sich der „Wert“ des Netzwerkes [Hendler and Golbeck, 2008]. Dieser steigt quadratisch mit der Anzahl der Teilnehmer [Metcalf, 1995].

Im Gegensatz zu den Verfahren des *Information Retrievals (IR)* [Salton, 1989], welche unter Verwendung von verschiedenen Techniken aus der Computerlinguistik oder aus dem Bereich der Künstlichen Intelligenzen versuchen, aus unstrukturiertem Text Wissen bezüglich einer im Vorhinein bekannten Domäne zu extrahieren, stellen die Techniken des *Semantic Webs* Meta-Konstrukte für die Repräsentation und Auszeichnung von Wissen bereit. Die Bedeutung dieses Wissens wird unter Benutzung von speziellen Auszeichnungssprachen *explizit* angegeben.

Vergleichbar mit der Idee des *World Wide Webs*, Dokumente miteinander zu vernetzen, versucht das *Semantic Web*, dies auch mit Informationen auf der Ebene ihrer *semantischen* Bedeutung zu tun. Während im klassischen *World Wide Web* die Hyperlinks zwischen den Dokumenten keine spezifische, über eine einfache Verknüpfung hinausgehende Bedeutung haben, sind die Verknüpfungen zwischen den einzelnen Konzepten innerhalb des *Semantic Webs* analog zu den Konzepteigenschaften *explizit* mit einer Bedeutung versehen. Durch diese Verknüpfung und ihre jeweiligen Bedeutung kann mit Hilfe entsprechender Techniken die Erschließung von neuen Zusammenhängen, die so nicht *explizit* aufgeführt sind, ermöglicht werden. Somit tritt auch im Kontext des *Semantic Webs* der bereits in Kapitel 2.1.2 beschriebene *Serendipitäts*-Effekt auf.

3.1 Technologien des Semantic Webs

Die einzelnen Methoden und Techniken des *Semantic Webs* sind in einem Schichtenmodell in syntaktisch voneinander unabhängigen, aber funktional aufeinander aufbauenden Ebenen (*Layern*) hierarchisch angeordnet. So wie in der Architektur des *ISO/OSI*-Schichtenmodells für Netzwerkkommunikation (siehe *ISO 7498-1*¹⁵) oder des *UNIX*-Kernels nutzen die jeweiligen *Layer* die Funktionen und Konzepte der direkt darunter liegenden Schicht, erweitern diese um weitere Methoden und Möglichkeiten und stellen sie dann für die direkt darüber liegende Schicht bereit. Das Schichtenmodell des *Semantic Webs*, *Semantic Web Stack* genannt, ist im sogenannten *Semantic Web Layer Cake* angeordnet. Die Abbildung 3.1 stellt die Hierarchie der einzelnen *Layer* dar und gibt die für die Konzepte des jeweiligen *Layers* gebräuchlichste Sprache an.

Im folgenden Abschnitt werden die Sprachen des *Semantic Webs* bis hin zum *Ontology-Layer* vorgestellt und ihre Konzepte sowie jeweiligen Funktionen detailliert beschrieben. Die konzeptuellen *Layer* oberhalb des *Ontology-Layers* werden hier nicht beschrieben, da sie noch nicht vom *W3C (World Wide Web Konsortium)*¹⁶ standardisiert sind und kontinuierlichen Änderungen unterliegen.

Das *W3C* entwickelt Spezifikationen und Richtlinien für das *World Wide Web* betreffende Techniken. Auch wenn das *W3C* bereits zahlreiche Standards festgelegt hat, so ist es keine zwischenstaatlich anerkannte Organisation wie beispielsweise die *International Organization of Standardization*¹⁷ und damit an sich nicht berechtigt, Normen festzulegen. Von daher werden

¹⁵<http://www.ecma-international.org/activities/Communications/TG11/s020269e.pdf>; Zugriff: 05. Juni 2011

¹⁶<http://www.w3.org/>Zugriff: 07. Juni 2011

¹⁷<http://www.iso.org>Zugriff: 07. Juni 2011

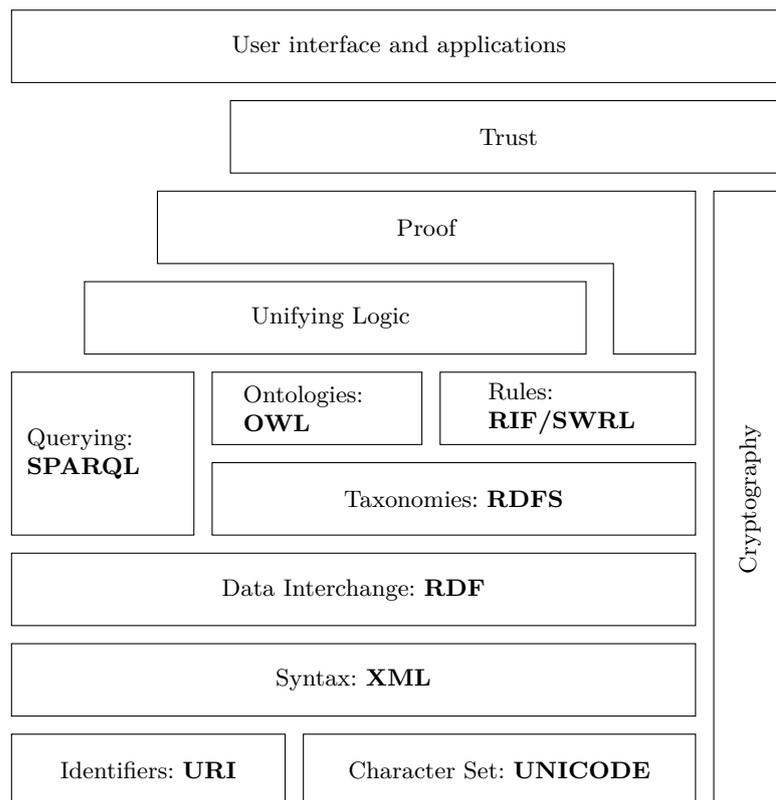


Abbildung 3.1 – Der Semantic Web Cake

die vom *W3C* herausgegebenen Standards *W3C-Recommendations* genannt. Zu diesen „Empfehlungen“ gehören die im *World Wide Web* weit verbreiteten de facto Standards wie *HTML*, *CSS*, *PNG* oder *SVG*.

3.1.1 Indentifikatoren

Ein *Uniform Resource Identifier (URI)* ist eine Zeichenkette, die für die Identifikation einer abstrakten oder real existierenden Ressource im Internet benutzt wird. *URIs* werden für die Bezeichnung von im *World Wide Web* verfügbaren Ressourcen wie beispielsweise Webseiten oder E-Mail-Adressen verwendet. Erst eine solche Identifikation ermöglicht die weltweit eindeutige Referenzierung auf ein einzelnes spezifisches Konzept. Ursprünglich von Sir Tim Berners-Lee 1994 im RFC1630 [Berners-Lee, 1994] unter dem Namen *Universal Resource Identifier* eingeführt, wurde dieser Standard seitdem mehrfach überarbeitet. Die generische Syntax der *URIs* folgt einer bestimmten, *URI Schema* genannten Definition, deren Syntax heutzutage im RFC 3986 [Berners-Lee et al., 2005] festgelegt ist.

URIs können als Namen (*URN*), als Ort (*URL*) oder sogar als beides aufgefasst werden. Beispielsweise kann der *Uniform Resource Name (URN)* [Moats, 1997] als der eigentliche Name einer Person interpretiert werden und der *Uniform Resource Locator (URL)* [Berners-Lee et al., 1994] als ihr Wohnsitz. Formeller ausgedrückt: die *URN* definiert anhand eines vorhandenen oder frei vergebenen Namen die Identität eines Elementes, während die *URL* den Mechanismus

3.1. Technologien des Semantic Webs

beschreibt, auf welche Art und Weise auf das Element zugegriffen werden kann, so zum Beispiel per *HTTP*- oder *FTP*-Protokoll.

Eine solche, anhand des *HTTP*-Schemas [Fielding et al., 1999] definierte *URI* ist in Beispiel 3.1 aufgeführt.

`http://myname.com/is#JohnDoe`

Quellcode-Beispiel 3.1 – Beispiel einer URI nach dem HTTP-Schema

Unter der *URI* `http://myname.com/is#JohnDoe` kann nun ein Element eindeutig gefunden werden. Seine genaue Bedeutung sowie weitergehende Informationen oder Relationen können allerdings nicht durch die alleinige Verwendung von *URIs* beschrieben werden. Um dies zu ermöglichen, werden weitere, über die *URIs* hinausgehende, Konstrukte benötigt. Nichtsdestotrotz bildet die Verwendung von *HTTP-URIs* einen zentralen Bestandteil des *Semantic Webs*, um Dokumente sowie Konzepte exakt identifizieren zu können.

3.1.2 Syntax und Serialisierung

Die *Extensible Markup Language (XML)* [Bray et al., 2008] ist eine Auszeichnungssprache für die Repräsentation hierarchisch strukturierter Daten in Form von einfachen Textdateien. Diese Textdateien enthalten ausschließlich Textzeichen, meist *UTF-8* [Yergeau, 2003] codierte Zeichen des *ASCII*-Alphabets [Cerf, 1969]. Eine Verwendung von Binärzeichen ist aufgrund des *XML*-Standards nicht vorgesehen.

XML wird heutzutage meist für einen plattformunabhängigen Austausch von Informationen zwischen verschiedenen Computersystemen, insbesondere über das *World Wide Web*, benutzt. So nutzt beispielsweise der *SOAP*-Standard [Mitra and Lafon, 2007] *XML* als Serialisierungsmethode für den Austausch von Parametern und Antworten zwischen verschiedenen Computern. Gerade wegen der durch die Verwendung von *XML* ermöglichten Plattformunabhängigkeit fand der *SOAP*-Standard seine weite Verbreitung im *World Wide Web* als Protokoll für Webservices und wird von den verschiedensten Programmiersprachen unterstützt.

Zwar kann mit Hilfe von *Document Type Definitions (DTDs)*¹⁸ eine formale Struktur für ein *XML* Dokument definiert werden, gegen welche sich die Dokumente verifizieren müssen (*Validität*), jedoch sind die einzelnen Attribute und Textknoten nicht typisiert. Die Einführung von *XML-Schema* [Fallside, 2001] brachte die geforderte Typisierung der Elementinhalte, dennoch sind auch hier keinerlei Referenzen auf andere Konzepte möglich.

Im folgenden einfachen Beispiel 3.2 ist eine Person in einem *XML*-konformen Dateifragment beschrieben.

¹⁸<http://dtdl.org/> Zugriff: 10. Juni 2011

```

<person>
  <name>John Doe</name>
  <mail>JohnDoe@myname.com</mail>
  <id>http://myname.com/is#JohnDoe</id>
</person>

```

Quellcode-Beispiel 3.2 – Auszug einer XML-Datei

Das *XML*-Fragment beschreibt zwar eine bestimmte Person, jedoch kann diese Struktur ohne weitere Angaben bezüglich ihrer Bedeutung nicht eindeutig interpretiert werden. Damit auch andere Maschinen dies tun können, existieren zu den am häufigsten verwendeten *XML*-Vokabularen meist informelle Spezifikationen in Form einer zusätzlicher von Menschen lesbaren Dokumentation. Sie beschreibt, welche Bedeutung die Werte, die Attribute oder die Textknoten innerhalb der *XML-Tags* aufweisen, so dass diese dann nach einer manuellen Integration verarbeitet werden können. Weiterhin existieren im *XML*-Standard keine global eindeutigen Identifikatoren, so dass es Schwierigkeiten bereiten kann, Daten aus verschiedenen Quellen zu integrieren. Zwar sehen einige der Vokabulare auch Mechanismen zur eindeutigen Identifikation vor, diese können dann jedoch nur die in diesem einen Vokabular modellierte Elemente identifizieren und ermöglichen keine Verbindung zwischen den Beschreibungen des selben Elementes in unterschiedlichen Vokabularen. Dies kann erst durch manuelle erzeugte Sammlungen von Abbildungen zwischen den Identifikatoren der Elemente für verschiedene Vokabulare erreicht werden.

Trotzdem bietet der *XML*-Standard die zuvor angesprochenen Plattformunabhängigkeit, was zu seiner weiten Verbreitung geführt hat. Gerade aus diesem Grund findet *XML* seine Verwendung als Serialisierungsmechanismus für Informationen im *Semantic Web*. Des Weiteren existieren mit der *Simple API for XML (SAX)*¹⁹ und dem *Document Object Model (DOM)* [Stenback and Heninger, 2004] zwei verschiedene *Parser*-Architekturen, welche als etabliertes Verfahren für die Serialisierung und Deserialisierung von *XML*-Dateien bereits in den meisten Programmiersprachen zu Verfügung stehen und damit eine entsprechende Verbreitung gefunden haben. Somit können viele der *XML*-Dateien verarbeitenden Applikationen leicht an die Verwendung von *Semantic Web*-Daten angepasst werden.

3.1.3 Datenaustausch

Das *Resource Description Framework (RDF)* [Hayes, 2004] als Sprache zur Auszeichnung von Daten basiert auf sogenannten *Statements*. Diese *Statements* haben die Form von Tripeln und bestehen aus einem *Subjekt (subject)*, einem *Prädikat (property)* und einem *Objekt (object)*. Bei einem solchen Tripel handelt es sich um ein um eine von der *property* gegebene Bedeutung erweitertes Key-Value-Pair. Somit kann eine mit einer Bedeutung versehene Aussage über ein einzelnes Konzept (im Folgenden *Ressource* genannt) getroffen werden.

¹⁹<http://www.saxproject.org/> Zugriff: 12. Juni 2011

3.1. Technologien des Semantic Webs

Im Gegensatz zu *XML* ist unter Verwendung einer *URI* als Identifikator für die *Ressource* sowie dem *Prädikat* die Möglichkeit gegeben, eine weltweit eindeutig interpretierbare Aussage darüber zu treffen. Dabei erhält die *Ressource* (*subject*) eine benannte Eigenschaft (*property*) mit einem durch das *Objekt* gegebenen Wert (*object*). Der Wert des *Objekts* kann entweder ein *Literal* (beispielsweise ein einfacher Datentyp wie eine Zahl oder eine Zeichenkette) oder selber wieder eine durch eine *URI* referenzierbare *Ressource* sein.

Das Besondere am *RDF*-Modell liegt darin, dass die als *Objekt* benutzten *Ressourcen* wiederum selber ein *Subjekt* sein können, über das dann wieder weitere Aussagen gemacht werden können. Das ermöglicht es, die verschiedenen *Ressourcen* untereinander in Relation zu setzen und damit ein Netzwerk einzelner Aussagen zu bilden, welches dann das modellierte Wissen der abgebildeten Domäne repräsentiert. Diese Sammlung von *RDF*-Tripeln bildet einen gerichteten Graphen, welcher auch als *RDF*-Graph bezeichnet wird.

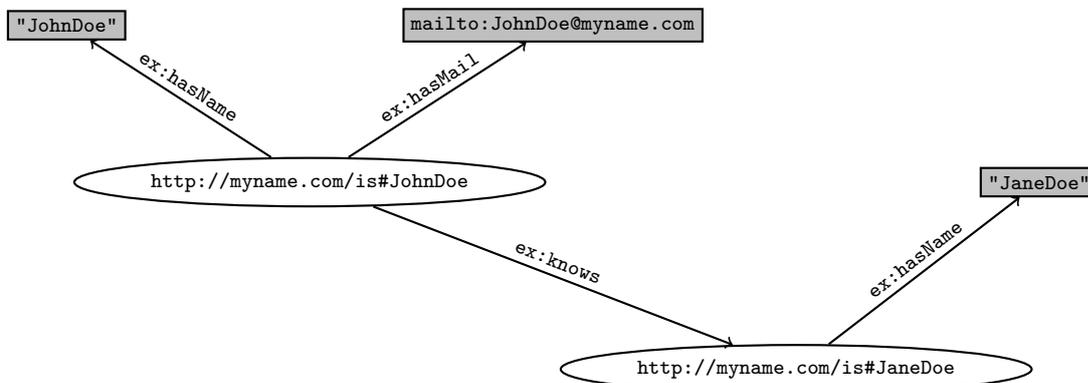


Abbildung 3.2 – Beispiel eines RDF-Graphen

Ein einfaches Beispiel eines solchen *RDF*-Graphen ist in Abbildung 3.2 gezeigt. Dieser *RDF*-Graph beschreibt eine Person (`http://myname.com/is#JohnDoe`), welche zwei Eigenschaften hat (`ex:hasName`, sowie `ex:hasMail`) und in Relation mit einer weiteren Person (`http://myname.com/is#JaneDoe`) steht. Dabei sind die *Ressourcen* als Ellipsen, die *Literale* als graue Rechtecke und die *Properties* als gerichtete Kanten zwischen beiden repräsentiert.

Das *RDF*-Modell ist unabhängig von dem Format einer speziellen Repräsentation. Obwohl meist wegen der in Kapitel 3.1.2 genannten Vorzüge das vom *W3C* spezifizierte und auf *XML* basierende Format *RDF-XML* [Carroll and Klyne, 2004] als Repräsentationsformat gewählt wird, existieren auch weitere Formate zur Serialisierung von *RDF*-Graphen, wie zum Beispiel *RDF/N3* (*Turtle Terse RDF Triple Language*) [Beckett and Berners-Lee, 2006] oder *N-Triple* [Beckett and Grant, 2004]. Erst unter Verwendung solcher Formate ist es möglich, *RDF*-Graphen zu serialisieren und für verschiedene Maschinen über das *World Wide Web* zugänglich zu machen. Beispiel 3.3 stellt eine in *RDF-XML* formulierte Serialisierung des *RDF*-Graphen aus Abbildung 3.2 dar.

```

<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://example.org/">
  <rdf:Description rdf:about="http://myname.com/is#JohnDoe">
    <ex:hasName>"John Doe"</ex:hasName>
    <ex:hasMail>mailto:JohnDoe@myname.com</ex:hasMail>
    <ex:knows>http://myname.com/is#JaneDoe</ex:knows>
  </rdf:Description>
  <rdf:Description rdf:about="http://myname.com/is#JaneDoe">
    <ex:hasName>"Jane Doe"</ex:hasName>
  </rdf:Description>
</rdf:RDF>

```

Quellcode-Beispiel 3.3 – Auszug einer RDF-XML-Datei

Eine andere Form der Verwaltung von *RDF*-Graphen besteht in der Verwendung der sogenannten *Triple*-Stores wie beispielsweise Virtuoso²⁰ oder Sesame²¹. Normale Mengen-basierte, relationale Datenbanksysteme unterstützen das spezielle Graph-basierte Format von *RDF* nur unzureichend und sind damit für die Speicherung und den Zugriff von Daten im *RDF*-Format ineffizient. *Triple*-Stores hingegen sind auf das Speichern und Abfragen von großen Datenmengen im *RDF*-Format hin optimiert. Allerdings existieren seit kurzer Zeit auch Bibliotheken wie ARC²², um *RDF*-Daten einfach in relationale Datenbanksysteme zu importieren und diese zumindest lesend zugreifbar zu machen. Diese erreichen jedoch nicht die Performanz der *nativen* Speichersysteme und weisen meist Einschränkungen in der Menge der unterstützten Operationen auf. Weiterhin existieren Wrappersysteme wie D2RQ [Bizer and Seaborne, 2004], welche es ermöglichen, eine *RDF*-Sicht auf die Daten einer relationalen Datenbank zu erzeugen. Auf die gespeicherten Daten wird in *Triple*-Stores durch spezielle Abfragesprachen wie beispielsweise *SPARQL* zugegriffen (siehe Kapitel 3.1.6), welche sich in ihrer Syntax der Datenbank-Sprache *SQL* ähnelt.

3.1.4 Taxonomien

Mit dem Modell des *Resource Description Framework* ist nun die Möglichkeit gegeben, Konzepte anhand von Statements zu beschreiben und damit eine Domäne zu modellieren. Allerdings bietet *RDF* keine Möglichkeit, eine Menge ähnlicher Konzepte wie z.B. Filme oder Bücher mit denselben Eigenschaften formal auszuzeichnen. Zum Austausch und zur gemeinsamen Interpretation von in *RDF* formulierten Statements wird also ein gemeinsames Vokabular benötigt, in welchem die Bedeutungen der einzelnen Konzepte, ihre Eigenschaften sowie ihre Beziehungen untereinander festgelegt sind.

Unter Benutzung von *RDF-Schema (RDFS)* [Guha and Brickley, 2004] ist es nun möglich, ein formales Vokabular für einen bestimmten Wissensbereich (Domäne) zu definieren und dadurch primitive Ontologien zu erzeugen. Wichtig anzumerken ist, dass die Klassen (`rdfs:Class`)

²⁰<http://virtuoso.openlinksw.com/> Zugriff: 17. Juni 2011

²¹<http://www.openrdf.org/> Zugriff: 17. Juni 2011

²²<https://github.com/semsol/arc2/wiki> Zugriff: 19. Juni 2011

3.1. Technologien des Semantic Webs

und deren Eigenschaften (`rdf:Property`) getrennt voneinander modelliert sind. Mit Hilfe der beiden Eigenschaften `rdfs:subClassOf` und `rdfs:subPropertyOf` werden Klassen bzw. Eigenschaften der Klassen jeweils untereinander in eine hierarchische Beziehung gesetzt. Diese Sprachkonstrukte ermöglichen die Erzeugung von einfachen Ontologien mit einer Baumstruktur, den sogenannten Taxonomien (auch Thesauri genannt).

Anhand der durch *RDF-Schema* zur Verfügung gestellten Sprachkonstrukte, kann die exakte Bedeutung der *Statements* formal und eindeutig definiert werden. So wird für jedes *predicate* (`rdf:Property`) festgelegt, welcher Klasse bzw. welchem einfachen Datentyp die Werte des *objects* angehören dürfen (`rdfs:range`) und welches *subject* mit dieser Eigenschaft beschrieben werden darf (`rdfs:domain`).

Mit Hilfe der sogenannten *Reasoner* lässt sich neues Wissen ableiten, da mit *RDFS* eine formale Struktur des Domänenmodells geschaffen werden kann. Ein *Reasoner* inferiert mit Hilfe seiner Ableitungsregeln automatisch aus dem vorhandenen Modell neue Informationen. So kann er beispielsweise die Transitivität der `rdfs:subClassOf`-Eigenschaft ausnutzen um aus der Aussage A ist Unterklasse von B und B ist Unterklasse von C die Aussage A ist auch Unterklasse von C abzuleiten. Es lässt sich auch Wissen ableiten, welches nicht ausschließlich auf den Sprachkonstrukten von *RDFS* basiert. So kann ein *Reasoner* aus der Aussage: „Eine Katze ist ein Tier und Tiere sind sterblich“ auf die Aussage „Katzen sind sterblich“ schließen. Neben den Inferenzen dienen die *Reasoner* auch dazu, die „logische“ *Validität* der Taxonomien zu überprüfen, welche nicht alleine durch die Konstrukte von *RDFS* sicher gestellt sein kann.

```
<foaf:Person rdf:about="http://myname.com/is#JohnDoe">
  <foaf:name>"John Doe"</foaf:name>
  <foaf:mbox rdf:resource="mailto:JohnDoe@myname.com"/>
  <foaf:knows>
    <foaf:Person rdf:about="http://myname.com/is#JaneDoe">
      <foaf:name>"Jane Doe"</foaf:name>
    </foaf:Person>
  </foaf:knows>
</foaf:Person>
```

Quellcode-Beispiel 3.4 – Auszug einer FOAF-Datei

Durch das mittels *RDF-Schema* definierte Vokabular für die Beschreibung von Personen *FOAF* [Brickley and Miller, 2007], kann nun die im Beispiel 3.3 genannte Person nach einer formalen Struktur modelliert werden. Dadurch können weltweit alle Computersysteme, welche dieses Vokabular verwenden, die Eigenschaften dieser Person gleich interpretieren und entsprechend verarbeiten. In Beispiel 3.4 wird die Person mit dem Namen 'JohnDoe' aus Beispiel 3.3 anhand des *FOAF*-Vokabulars beschrieben. Zusätzlich wird angegeben, dass sie über die Relation `foaf:knows` mit einer Person namens 'JaneDoe' verbunden ist.

Trotz alledem muss das zur Modellierung eingesetzte Vokabular auch entsprechend weit verbreitet sein und verwendet werden, da es sonst wenig Sinn macht, die Informationen dem Vo-

kabular folgend zu beschreiben. Dabei tritt jedoch das bekannte „Henne-Ei“-Problem auf. Nur durch eine weite Verbreitung wird das Vokabular von neuen Teilnehmern eingesetzt, es findet jedoch nur eine größere Verbreitung, wenn eine ausreichende Anzahl an Teilnehmern das Vokabular bereits benutzt. Denn erst durch eine gemeinsame Verwendung dieses Vokabulars kann die eindeutige Formulierung der Informationen von verschiedenen Maschinen identisch aufgefasst werden, welche die Bedeutung der Aussagen dann jeweils anwendungsspezifisch verarbeiten.

3.1.5 Ontologien

*Eine Ontologie ist ein Thesaurus auf Steroiden.*²³

Ontologien erweitern die bereits vorgestellten Taxonomien um weitere Konzepte und Methoden, damit eine Domäne noch präziser beschrieben werden kann. Dabei stellen die Ontologien den Oberbegriff dar, denn jede Taxonomie ist auch eine Ontologie.

Formal ausgedrückt ist eine Ontologie eine explizite und formale Spezifikation der Konzeptualisierung einer fest umrissenen Domäne für einen definierten Zweck, auf die sich eine Gruppe von Akteuren geeinigt hat [Studer et al., 1998]. Sie dient dazu, die Kommunikation zwischen den Akteuren zu ermöglichen oder zu vereinfachen, da sich diese auf ein gemeinsames Vokabular, die Ontologie, einigen (*Einigungsaspekt*). Die Akteure können dabei sowohl Menschen als auch Computer sein.

Eine weitere Definition besagt, dass eine Ontologie die Konzeptualisierung einer Domäne durch ein maschinenlesbares Format ist [Guarino and Garetta, 1995]. Normalerweise nimmt sie dabei die Form eines gerichteten Graphen, bestehend aus Konzepten, deren Attributen und Beziehungen untereinander sowie weiteren Axiomen, an. Die Axiome beschreiben Wertebereiche der Beziehungen und Attribute sowie weitere Eigenschaften der Beziehungen wie beispielsweise Kardinalitäten, Transitivität oder Symmetrie (*Konzeptualisierung*). Die Konzeptualisierung findet in einer eindeutigen und möglichst wenig fehlinterpretierbaren Form statt (*explizite Spezifikation*). Dies geschieht im Normalfall durch spezielle Ontologie-Repräsentationssprachen, welche in ihrer *Semantik* vollständig und eindeutig spezifiziert sind.

Klassifikation von Ontologien

Es existieren verschiedene Möglichkeiten, Ontologien zu klassifizieren. Ontologien lassen sich anhand ihrer Struktur und Aussagemächtigkeit in zwei Typen einteilen, zum einen in die *lightweight*-Ontologien, zum anderen in die *heavyweight*-Ontologien. Im Gegensatz zu den weniger komplexen *lightweight*-Ontologien, welche in Form von Taxonomien auftreten und damit nur eine einfache hierarchische Gliederung zwischen den einzelnen Konzepten und deren Eigenschaften ermöglichen, stellen *heavyweight*-Ontologien ein Netzwerk von Informationen mit Relationen dar. Dabei erweitern die *heavyweight*-Ontologien die *lightweight*-Ontologien um Einschränkungen und Axiome, um eine höhere Aussagekraft für das Modell der Domäne zu erreichen und so die Bedeutungen der Aussagen innerhalb der Ontologie klarer darzustellen [Gómez-Pérez et al., 2004].

²³aus dem Vortrag von Chiara Fox: The Semantic Web: What IAs Need to Know About Web 3.0, IA Summit 2009

3.1. Technologien des Semantic Webs

Eine weit verbreitete Klassifikation stammt von [Guarino, 1998]. Er unterscheidet Ontologien anhand ihrer Generizität:

- *Upper-Level-Ontologien* beschreiben sehr allgemeine Konzepte, wie Raum, Zeit, Aktion, Ereignisse etc., welche unabhängig von einer speziellen Problemstellung oder einer Domäne sind.
- *Domänen-Ontologien* beschreiben die verwendeten Konzepte einer allgemeinen Domäne, wie beispielsweise Medizin oder Automobilbau, indem sie die durch die *Upper-Level-Ontologien* eingeführten Konzepte spezialisieren.
- *Aufgaben-Ontologien* beschreiben die verwendeten Konzepte eines Prozesses oder einer Aufgabe, indem sie die durch die *Upper-Level-Ontologien* eingeführten Konzepte spezialisieren.
- *Applikations-Ontologien* beschreiben die Konzepte sowohl einer Domäne als auch eines zu dieser Domäne zugehörigen Prozesses, indem sie die Konzepte der *Domänen-Ontologien* sowie der *Aufgaben-Ontologien* spezialisieren.

Weiterhin lassen sich Ontologien anhand der Formalität der zur Modellierung verwendeten Sprache einteilen. [Uschold and Gruninger, 1996] klassifizieren sie wie folgt mit aufsteigender Formalität:

- *informell*: wenn sie in natürlicher Sprache ausgedrückt sind;
- *semi-informell*: wenn sie in einer eingeschränkten und strukturierten Teilmenge der natürlichen Sprache ausgedrückt sind;
- *semi-formell*: wenn sie in einer semi-formell definierten Sprache ausgedrückt sind;
- *formell*: wenn sie in einer formell definierten Sprache ausgedrückt sind.

Wenn Ontologien benutzt werden, um Wissen zwischen Menschen auszutauschen, sollten eher *informelle* oder *semi-informelle* Ontologien verwendet werden. Wenn die Partner jedoch Maschinen sind, sollten mindestens *semi-formelle* Ontologien zum Einsatz kommen, da diese leichter von Maschinen verstanden und verarbeitet werden können [Yildiz, 2006].

Ontology-Engineering

Ontology-Engineering ist formal definiert als die Menge an Aktivitäten, welche sich mit dem Entstehungsprozess, dem Lebenszyklus sowie den Methoden, Werkzeugen und Sprachen bei der Erschaffung von Ontologien beschäftigt [Gómez-Pérez et al., 2003]. [Simperl et al., 2009] unterscheiden zwei verschiedene Arten des *Ontology-Engineerings*, den *zentralisierten* und den *dezentralisierten* Engineering-Ansatz. Im *zentralisierten* Engineering findet der Entstehungsprozess an einem Ort statt, und der Austausch zwischen den beteiligten Entwicklern und Domänenexperten geschieht in regelmäßig abgehaltenen *Face-2-Face-Meetings*²⁴. Beim *dezentralisierten*

²⁴*Face-2-Face-Meetings* bezeichnen Besprechungen von Angesicht zu Angesicht an einem gemeinsamen Ort

Engineering hingegen geschieht der Entwicklungsprozess räumlich und zeitlich verteilt. So müssen an Methoden, Werkzeuge sowie an die Kommunikation, welche asynchron abläuft, andere Bedingungen gestellt werden, als beim *zentralisierten* Engineering. Dabei dient die gemeinsame Ontologie als Verständigungsmechanismus zwischen den Partnern und stellt die Interoperabilität zwischen den Menschen, den Maschinen oder beidem sicher. Einige dem *zentralisierten* Engineering zuzuordnende Verfahren sind beispielsweise IDEF5 [Menzel et al., 1993], METHONTOLOGY [Fernandez-Lopez et al., 1997] und OTK [Sure et al., 2002]. DILIGENT [Vrandečić et al., 2005; Tempich et al., 2007] hingegen basiert auf den kollaborativen Prinzipien des *dezentralisierten* Engineering. Eine weitere Möglichkeit, eine Ontologie zu erzeugen, stellt das *crowd sourcing* Konzept dar. Während in den vorgestellten Methoden normalerweise nur eine kleine Gruppe von Domänenexperten an der Erstellung beteiligt ist, geschieht dies beim *crowd sourcing* durch eine große Gruppe „normaler“ Benutzer [Siorpaes and Hepp, 2008]. Dabei ist jeder einzelne Benutzer jeweils nur an der Entstehung eines kleinen Aspektes der gesamten Ontologie beteiligt. Aufgrund der *Weisheit der Massen* können so oftmals Fehler aufgespürt werden, welche sonst nur durch kosten- und zeitintensive Validierungsprozesse bestimmbar gewesen wären [Chua et al., 2010].

Der Aufwand, welcher nötig ist, um benutzbare Ontologien für eine spezifische Domäne zu erzeugen, zu unterhalten und damit gut modelliertes und strukturiertes *RDF* beziehungsweise *OWL* zu erhalten, kann verhältnismäßig hoch sein [Simperl et al., 2006]. Von daher ist es sinnvoll, bereits weit verbreitete und etablierte Vokabulare und Ontologien zu verwenden. Dies spart Kosten, da diese Ontologien in einem solchen Fall von den entsprechenden Domänenexperten gewartet und aktualisiert werden. Des Weiteren entspricht dies auch der eigentlichen Idee des *Semantic Webs*, Wissen aus unterschiedlichen Quellen zu verknüpfen, und erhöht so den Wert des Netzwerkes.

Das Ziel der Verknüpfung von Informationsquellen verfolgt das *Linked Data*-Projekt²⁵. *Linked Data* bezeichnet ein Netzwerk, in welchem die Daten per *URI* identifizierbar und über diese *URI* auch dereferenzierbar sind. Die Daten sollten idealerweise im *RDF*-Format vorliegen, so dass sie leicht untereinander verknüpft werden können. Somit wird ein weltumspannendes Netz gebildet, welches auch als *Linked (Open)²⁶ Data Cloud* bezeichnet wird.

Ontologiesprache

Die am weitesten verbreitete und vom *W3C* propagierte Sprache für die formale Beschreibung von Ontologien ist die *Web Ontology Language (OWL)* [Hayes et al., 2004]. *OWL* basiert technisch auf *RDFS* und erweitert es um weitere Sprachkonstrukte zur Formulierung von Ausdrücken ähnlich der Prädikatenlogik. Dadurch geht die Ausdrucksmächtigkeit von *OWL* über die von *RDFS* weit hinaus.

Die *OWL*-Spezifikation des *W3C* unterscheidet drei Untersprachen von *OWL*, wobei jede eine unterschiedlichen Aussagemächtigkeit aufweist: *OWL Lite*, *OWL-DL* sowie *OWL Full*. Jede Untersprache ist eine syntaktische Erweiterung der jeweils einfacheren vorhergehenden und führt

²⁵<http://linkeddata.org/> Zugriff: 23. Juni 2011

²⁶Open bezeichnet verknüpfte Daten, die frei zugänglich sind. Dort, wo Daten nur unter Einschränkung zugreifbar sind, wird von *Linked Data* gesprochen.

3.1. Technologien des Semantic Webs

weitere Sprachkonstrukte und Möglichkeiten ein, welche die Aussagemächtigkeit der Sprache erhöhen.

- *OWL Lite* wurde in erster Linie definiert, um eine einfach zu implementierende und verarbeitbare Teilmenge von *OWL* zu haben. Allerdings reicht ihre Mächtigkeit nur wenig über *RDFS* hinaus, weshalb sie kaum Verwendung findet.
- *OWL-DL* beinhaltet alle Sprachkonstrukte von *OWL*. *DL* steht für *Description Logic SHOID (D)*, welche eine entscheidbare Untermenge der Prädikatenlogik ist. Um eine Entscheidbarkeit dieser Sprache zu ermöglichen, existieren diverse Einschränkungen für den Einsatz von *RDFS*. Gerade aufgrund ihrer Entscheidbarkeit und ihrer großen Mächtigkeit, ist *OWL-DL* der am meisten genutzte Dialekt von *OWL*.
- *OWL Full* beinhaltet dieselben syntaktischen Konstrukte wie *OWL-DL*, verzichtet jedoch auf die dortigen Einschränkungen. Dadurch können sehr mächtige Ontologien erzeugt werden, welche jedoch unentscheidbar sein können. Von daher findet *OWL-Full* wenig Verwendung bei Anwendungen und ist eher von akademischem Interesse.

Vor kurzem wurde vom *W3C* eine Überarbeitung und Erweiterung des *OWL*-Standards namens *OWL2* [Motik et al., 2009] verabschiedet. In *OWL2* wurden aufgrund der mit *OWL* gebildeten Erfahrungen bei dessen Einsatz jede der *OWL*-Varianten überarbeitet und die Ausdruckstärke durch Einführung zusätzlicher Axiome weiter verstärkt.

Datenintegration mittels Ontologien

Nicht nur, dass mit Ontologien Wissen einer Domäne durch Modellierung von Konzepten und Relationen beschreiben werden kann, sie sind auch Werkzeuge zur Datenintegration. Mit ihnen können Abbildungen (sogenannte *Mappings*) zwischen Konzepten beziehungsweise Daten, die an verschiedenen Orten und in unterschiedlichen Formaten vorliegen, realisiert werden. Dadurch können diese heterogenen Daten in ein einheitliches Format überführt und trotz unterschiedlicher Ursprungsformate und -orte einheitlich und transparent zugreifbar gemacht werden. Solche *Mappings* unterliegen jedoch einem aufwändigen und überwiegend manuellen Entstehungsprozess, der noch nicht vollständig automatisch ablaufen kann. Nichtsdestotrotz kann, sobald ein solches *Mapping* erzeugt wurde und eingesetzt wird, der Nutzen die Kosten übersteigen, weil ab diesem Zeitpunkt die Integration von zukünftigen Daten aus den Ursprungsformaten automatisch vonstatten gehen kann.

3.1.6 Abfragen

SPARQL [Prud'hommeaux and Seaborne, 2008] ist eine Sprache, um Anfragen an ein *RDF*-Modell stellen zu können. Dabei ist der Name ein rekursives Akronym und steht für *SPARQL Protocol and RDF Query Language*, welche sich in ihrer Syntax an *SQL* orientiert. Die Prädikatenlogik-basierte Abfragesprache *SQL* erlaubt Anfragen an Daten, welche in relationalen Datenbanken in Form von Tabellen gespeichert sind. Die Einträge in Tabellen sind durch eindeutige Schlüssel unterscheidbar und lassen sich mit Einträgen aus anderen Mengen durch

die sogenannten Fremdschlüssel verknüpfen. Fremdschlüssel bezeichnen Verweise von Einträgen einer Tabelle auf die Einträge einer zweiten Tabelle anhand ihrer Schlüssel. Eine *SQL*-Query besteht aus einer *SELECT*-Clause, welche die aus dem Modell abgefragten Variablen angibt, gefolgt von einer *WHERE*-Clause, in der Parameter auf Gleichheit mit einer gegebenen Konstante oder auf Inhalt eines Teilstrings sowie weitere Vergleichsoperationen getestet werden können.

Zwar beinhaltet *SPARQL* die gleichen Schlüsselworte, ist jedoch aufgrund der Graphenstruktur des *RDF*-Modells eine Graph-basierte Abfragesprache. Es existiert dabei nicht das Konzept der Fremdschlüssel, da Verweise zwischen den Konzepten in Form von *Properties* realisiert sind, so dass die *WHERE*-Clause entsprechend anders aufgebaut ist. Hier werden die Daten mit den sogenannten *Basic-Graph-Patterns* (*BGP*) abgefragt, die Teilgraphen eines *RDF*-Graphen beschreiben. Diese bestehen aus festen sowie variablen Teilen und sind analog zur Tripel-Struktur des *RDF*-Modells definiert. Somit entspricht die Ausführung einer *SPARQL*-Query eigentlich der Suche nach einem durch die *BGPs* angegebenen Teilgraphen innerhalb eines *RDF*-Graphen. Neben den *BGPs* kann die *WHERE*-Clause zudem *FILTER*-Clauses beinhalten, welche einfache Vergleiche zwischen den variablen Teilen und angegebenen Konstanten ermöglichen.

<i>SELECT</i> :	Ergebnismengen
<i>CONSTRUCT</i> :	neu definierte <i>RDF</i> -Teilgraphen
<i>DESCRIBE</i> :	die direkte Nachbarschaft des abgefragten Teilgraphen
<i>ASK</i> :	boolesche Ausdrücke, welche angeben ob die Ergebnismenge leer ist

Tabelle 3.1 – Typen von *SPARQL*-Queries

SPARQL kennt im Gegensatz zu *SQL*, welches nur *SELECT* als einzige mögliche Form der Abfrage-Query anbietet, mehrere verschiedene Typen von Queries mit jeweils unterschiedlichen Formen der Ergebnisse. Diese sind in Tabelle 3.1 aufgeführt.

Die im nachfolgenden Beispiel aufgeführte Query beschreibt die Abfrage nach dem Namen (*name*) und der Email-Adresse (*mbox*) aller Personen, welche in Beziehung zur Person mit *name*= "JaneDoe" stehen. Diese Relationen werden mit *BGPs* ausgedrückt. Dabei werden die variablen Teile mit einem führenden „?“ ausgedrückt, während konstante Teile wie beispielsweise die *URL* der *property* in diesem Beispiel das „?“ nicht aufweisen. Die variablen Teile können miteinander verknüpft werden, so dass dadurch beliebige Teilgraphen ausgedrückt werden können.

```

PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
SELECT ?name, ?mbox WHERE {
  ?x foaf:name ?name .
  ?x foaf:knows ?y .
  ?y foaf:name "Jane Doe" .
  ?x foaf:mbox ?mbox .
}

```

Quellcode-Beispiel 3.5 – Beispiel einer *SPARQL*-Query

3.2. Semantic Recommender

Wird der *RDF*-Graph aus Beispiel 3.4 mit dieser Query abgefragt, ist das Ergebnis für die Parameter `name="JohnDoe"` und für `mbox="mailto:JohnDoe@myname.com"`.

3.2 Semantic Recommender

Nach den Definitionen von [Peis et al., 2008], stellen *Semantic Recommender* an sich keine eigenständige Klasse der *Recommender* da. Sie beschreiben vielmehr eine Erweiterung der in Kapitel 2.1 vorgestellten grundlegenden *Recommender*-Typen um eine *Wissensbasis* in Form einer Taxonomie oder Ontologie, anhand der ein *Recommender* seine Vorschläge für Elemente generiert. Dabei können die *Wissensbasen* durch die entsprechenden Technologien des *Semantic Webs* repräsentiert werden. Nach dieser Definition sind die von [Burke, 2000] eingeführten und in Kapitel 2.1.4 vorgestellten *Knowledge-Based Recommender* auch eine gewisse Ausprägung der *Semantic Recommender*, obwohl ihre *Wissensbasis* für ihren speziellen Anwendungsfall festgelegt ist. Vom gegensätzlichen Standpunkt aus betrachtet, können jedoch die *Semantic Recommender* auch als *Hybrid Recommender* aufgefasst werden, welche die von [Adomavicius and Tuzhilin, 2005] genannten Grundsysteme mit einem *Knowledge-Based Recommender* zu einem *Hybrid Recommender* verbinden. Im Gegensatz zu den bisher verwendeten *Knowledge-Based Recommendern*, welche ihr Domänenwissen fest in das Gesamtsystem integriert haben, hat ein *Semantic Recommender* sein Wissen in Form einer austauschbaren Ontologie oder Taxonomie integriert, welche in einer Repräsentationssprache aus dem *Semantic Web* beschrieben ist.

Einer der wichtigsten Gründe für die Verwendung von *Semantic Web*-Technologien ist die Verbesserung der Qualität gemachter Vorhersagen. Die Qualität der Ergebnisse eines Suchsystems wird in der Regel mit den beiden im *Information Retrieval* verwendeten Größen *Precision* und *Recall* gemessen [Cleverdon and Mills, 1966]. Die *Precision* beschreibt das Verhältnis der Anzahl der gefundenen richtigen Ergebnisse gegenüber der Anzahl gefundener Ergebnisse, und der *Recall* beschreibt das Verhältnis der Anzahl der gefundenen richtigen Ergebnisse zur Anzahl aller möglicher richtigen Ergebnisse. Aufgrund ihrer thematischen Nähe zu den klassischen *Information Retrieval*-Verfahren kommen diese beiden Größen auch bei der Evaluierung von *Recommendern* zum Einsatz (siehe Kapitel 6.3).

Mit der Verwendung von *expliziten* Wissensstrukturen lässt sich eine nicht unerhebliche Steigerung der Genauigkeit der Suchergebnisse innerhalb der Suche von Textkorpora erzielen, was diese besonders für die textanalysierenden Verfahren der *Content-Based Recommender* einsetzbar macht. So konnte im System von [Finin et al., 2005] die *Precision* um 40 % gegenüber dem ausschließlich Freitext betrachtenden Baseline-Verfahren gesteigert werden. In einem anderem System erhöhte die Verwendung einer Taxonomie die *Precision* der Vorschläge um über 60 % gegenüber dem dortigen Baseline-Verfahren [Bradley et al., 2000].

Im Folgenden werden exemplarisch einige *Semantic Recommender* aus der Literatur beschrieben sowie einiger der durch sie ermöglichten Vorteile gegenüber den traditionellen Verfahren vorgestellt.

Taxonomy-Driven Recommendation

[Ziegler et al., 2004b] verwendet Taxonomien, um den *Collaboration via Content*-Ansatz von [Pazzani, 1999] unter Benutzung von *explizitem* Wissen zu verbessern. Mit Hilfe dieser Taxonomien wird ein Benutzerprofil basierend auf den Interessensgebieten des Benutzers erzeugt, was durch die Hierarchie der Konzepte aussagekräftiger und genauer ist. Im *Collaboration via Content*-Ansatz wird analog zum *Collaborative Filtering* eine Gruppe von Nutzern bestimmt, die ähnlich zum aktuellen ist. Im Gegensatz zum einfachen *Collaborative Filtering* werden die ähnlichen Benutzer für die Nachbarschaft nicht ausschließlich aufgrund ihrer abgegebenen Bewertungen ausgewählt, sondern es werden auch die von ihnen bewerteten Elemente analysiert. In die Berechnungen fließen somit nur die Bewertungen der Benutzer ein, die auch ähnliche Elemente wie der aktuelle Benutzer bevorzugen. In diesem *Semantischen Recommender* hierarchisiert die verwendete Taxonomie die Elemente anhand ihrer Eigenschaften und bildet die Basis für das Profil des Benutzers. Dieses Profil besteht aus den gewichteten Eigenschaften der Elemente von Interesse sowie deren Oberbegriffen und bildet einen Ausschnitt aus der gesamten Taxonomie. Die Ähnlichkeit zwischen zwei Benutzern wird bestimmt, indem ihre Profiltaxonomien anhand der Pfade der vorhandenen Konzepte in der Taxonomie miteinander verglichen werden. Nachdem die ähnlichsten Personen bestimmt sind, werden aus ihren bewerteten Elementen diejenigen ermittelt, die am Besten zum Profil des aktuellen Benutzers passen. In der Evaluierung konnte neben der Erhöhung der *Precision* und des *Recalls* auch gezeigt werden, dass dieser *Recommender* für die Testbenutzer subjektiv bessere Ergebnisse erzielt, als das rein *Collaborative Filtering* verwendende Vergleichssystem.

Foxtrott und Quickstep

Weiterhin können mit der Verwendung von Taxonomien sowie Ontologien die einzelnen Interessensbereiche eines Benutzers untereinander in Relation gesetzt und somit Zusammenhänge zwischen diesen beschrieben werden. Beispielsweise wird von [Middleton et al., 2002; Middleton et al., 2004] eine Ontologie über die Themengebiete wissenschaftlicher Publikationen verwendet, um ein Benutzerprofil zu erzeugen. Diese Ontologie stellt die einzelnen Themengebiete untereinander in Relation. Durch einen Klassifikationsalgorithmus werden die Themenbereiche einer Publikation bestimmt, indem deren Repräsentationen, bestehend aus ihren gewichteten Termen, mit den Repräsentationen des händisch bestimmten Trainingskorpus verglichen werden. Das Profil eines Benutzers beinhaltet die Themengebiete und ihre Oberbegriffe der von ihm betrachteten Publikationen. Die Vorschläge werden erzeugt, indem die ermittelten Gebiete einer Publikation mit denen des Profils verglichen werden, wobei auch die Oberbegriffe verwendet werden. Dadurch ermöglicht das erweiterte Profil, dem Benutzer auch Publikationen eines Themengebietes vorzuschlagen, welches er noch nicht betrachtet hat, da es auch benachbarte Gebiete von Interesse erfasst. Erst durch die Verknüpfung mittels einer Ontologie lassen sich diese Gebiete ermitteln, durch eine ausschließliche Verwendung von Stichwortvergleichen ist dies nicht realisierbar. Die Verwendung von Ontologien stellt entsprechend eine Lösung für das bei *Content-Based Recommendern* auftretende Problem der *Überspezialisierung* dar.

News@hand

Auch [Cantador et al., 2008] verwendet Ontologien, um ein *semantisches* Profil für die Benutzer zu erzeugen. In diesem System wird das Profil dazu eingesetzt, dem Benutzer aktuelle und zu seinen Interessen passende Nachrichten vorzuschlagen. Die Ontologie kombiniert dabei mehrere einzelne Ontologien, welche verschiedene Themenbereiche wie beispielsweise Politik oder Kultur beschreiben. Weiterhin dienen Wikipedia²⁷-Einträge und ihre zugehörigen Kategorien als einzelne Instanzen für die Klassen der Ontologie. Unter Verwendung des *NLP*-Verfahrens²⁸, der *Named Entity Recognition* sowie dem statistischen *TDF-IDF*-Maß [Salton, 1989], werden die Nachrichtentexte verarbeitet und die aus ihnen extrahierten Terme auf ihre jeweiligen Konzepte der Ontologie gemappt und gewichtet. Somit besteht jeder einzelne Text aus einer Ansammlung von gewichteten Konzepten, welche mit den Benutzerprofilen anhand des Cosinusmaßes verglichen werden. Aufgrund der Ontologie können auch solche Elemente bestimmt werden, die sich im Interessengebiet des Benutzers befinden, deren Eigenschaften aber in den von ihm bisher betrachteten Elementen noch nicht vorgekommen sind. Dazu wird das Profil des Benutzers um weitere Konzepte, welche in Verbindung mit denen seiner Interessengebieten stehen, erweitert. Dies ermöglicht neben der verbesserten *Precision* auch eine Lösung für das *Data Sparsity* Problem. Wenn keine Überlappungen zwischen den Profilen und Elementen existieren, kann auf gegebenenfalls vorhandene Oberkonzepte eines erweiterten Benutzerprofils ausgewichen werden, wodurch immer noch für den Benutzer interessante Elemente bestimmt werden können.

Semantically Enhanced Collaborative Filtering

Die im *Semantic Recommender* von [Mobasher et al., 2003] benutzte Ontologie, bestehend aus Filmen, deren Darsteller und Regisseure wurde nicht *explizit* mittels einer Sprache des *Semantic Webs* definiert. Statt dessen wurde sie als einfaches Datenbankschema modelliert, folgt aber in ihrer Struktur dem Ontologiekonzept des *Semantic Webs*. Die Ontologie wird mit Hilfe von *Meta*-Daten, welche von IMDB²⁹, einer Webseite für Informationen über Filme, erstellt und gewartet werden, erzeugt und dient als *Wissensbasis* des Systems. Nach der Integration der Daten in die Tabellen wurden diese mittels *Latent Semantic Indexing* bearbeitet. Die resultierenden Matrizen dienen dann der Bestimmung der Ähnlichkeiten zwischen den Elementen unter Verwendung des Cosinusmaßes. Im Gegensatz zu den anderen hier genannten *Semantic Recommendern* ist dieser *Recommender* als *Weighted-Hybrid* mit *Item to Item Collaborative Filtering* realisiert. Er ermittelt neben den eben genannten Ähnlichkeiten aufgrund der Eigenschaften auch die Ähnlichkeiten anhand der Elementbewertungen und generiert in Kombination dieser beiden Werte die Vorschläge für die Benutzer. Neben einer Verbesserung der Qualität der Vorschläge konnte auch ein positiver Einfluss auf den *Cold-Start* beobachtet werden. Selbst wenn noch nicht ausreichend Bewertungen auf einzelne Elemente vorliegen, können ihre Ähnlichkeiten dennoch aufgrund ihrer Eigenschaften ermittelt werden. Es konnte zudem mit diesem System

²⁷<http://www.wikipedia.org> Zugriff: 29. Juni 2011

²⁸*NLP*: Natural Language Processing ist das Forschungsgebiet, welches sich mit der algorithmischen Verarbeitung von natürlich sprachlichen Texten beschäftigt

²⁹<http://www.imdb.com> Zugriff: 22. August 2011

nachgewiesen werden, dass bei der Vorschlagsgenerierung nicht alle Relationen zwischen den einzelnen Konzepten der *Wissensbasis* die gleichen Auswirkungen auf die Ergebnisse haben.

CHIP (Cultural Heritage Information Personalization)

Ähnliche Resultate wurden durch die Evaluierung des *Semantic Recommenders* von [Wang et al., 2008; Wang et al., 2009] erzielt. Im Gegensatz zu den bisher vorgestellten Systemen, liegen die Daten hier in Form einer formal definierten Beschreibung der Domäne vor, und nicht nur in einfacher Textform. Es wird eine Ontologie verwendet, um Kunstwerke eines Museums zu beschreiben und diese durch Relationen und Konzepte, wie beispielsweise Künstler, Epoche oder Entstehungsjahr, zu modellieren. Den Benutzern wird die Möglichkeit gegeben, die betrachteten Elemente anhand einer diskreten Skala zu bewerten. Die Vorschläge für sie werden generiert, indem ausgehend von ihren bisher am besten bewerteten Kunstwerken, die ähnlichsten weiteren ermittelt werden. Dabei basiert die Ähnlichkeit zwischen zwei Kunstwerken auf gemeinsam vorhandenen Relationen und deren identischen Werten. Das Ergebnis der Evaluierung war, dass die nach menschlichem Verständnis naheliegendsten Relationen subjektiv die besten Resultate erzielen. So erzeugte die „Entstehungsepochen“-Relation weit bessere Vorhersagen als die „Geburtsort des Künstlers“-Relation. Jedoch blieb die Frage, welche Relation automatisch auszuwählen ist, offen. Trotz dieser Einschränkung konnte gezeigt werden, dass dieses System das *Cold-Start-Problem* für neue Benutzer umgehen kann sowie subjektiv bessere Ergebnisse ermöglicht.

3.3 Zusammenfassung

Mit Hilfe der vorgestellten Techniken des *Semantic Webs* können Elemente und Benutzer besser und vor allem in eine maschinenlesbare Form modelliert werden. Ein großer Vorteil im Hinblick auf *Recommender Systeme* ist die Möglichkeit der eindeutigen Auszeichnung von Personen und Elementen sowie die einfache Verteilbarkeit der Daten. Weiterhin bieten sie den Vorteil einer einfachen Modellierung der *Wissensbasis*, welche die *Knowledge-Based Recommender* zwingend für ihren Betrieb voraussetzen.

So können sich beispielsweise Hersteller auf ein einheitliches Vokabular für eine Klasse von Produkten verständigen, wodurch ein nicht unerheblicher Integrationsaufwand entfällt. Dieser würde ansonsten entstehen, wenn die Produktbeschreibung des Herstellers erst in das eigene verwendete Format überführt werden müsste. Sobald nun ein neues Produkt bereitgestellt wird, braucht dieses nur einmalig mittels eines *RDF*-Graphen durch den Hersteller beschrieben werden und alle *Online-Shops* können dieses Produkt sofort und einfach in ihr System integrieren.

Durch die Vernetzung der Informationen, welche durch das *Semantic Web* realisiert wird, können leicht zusätzliche Informationen über ein Element in den Entscheidungsprozess mit einfließen, welche sich nicht in der eigentlichen Elementbeschreibung befinden, sondern aus externen Quellen stammen. Erst durch eine global eindeutige Identifikation lässt sich dies anwendungs- und systemübergreifend realisieren. Die externen Quellen können auch herangezogen werden, um fehlende Daten eines Elementes aufgrund einer unvollständigen Beschreibung auszugleichen.

3.3. Zusammenfassung

Dabei können diese direkt aus den Quellen wie beispielsweise der *Linked Data Cloud* bezogen werden, oder aber aus diesem Wissen interpoliert werden.

Neben diesen Vorzügen, welche gewisse seiner Applikationen beinhalten, bietet das *Semantic Web* insbesondere für *Recommender* Vorteile. Neben der Erhöhung der Qualität der gemachten Vorhersagen stellt die Verwendung von *semantischen* Techniken auch Lösungsmöglichkeiten für die in Tabelle 2.1 aufgeführten Probleme dar. So konnten die in Kapitel 3.2 vorgestellten Systeme mit diesen Techniken Lösungen für die *Cold-Start*-Problematik, das *Sparsity*-Problem sowie die *Überspezialisierung* aufzeigen. Werden sie als *Hybrid Recommender* betrachtet, welche einen *Knowledge-Based Recommender* beinhalten, so bestätigt dies die von [Burke, 1999] getätigte Aussage über die Architektur eines idealen *Hybrid Recommenders* (siehe Kapitel 2.4). Auch wenn bestimmte Vorzüge sicherlich auch unter Benutzung der klassischen *Knowledge-Based Recommender* realisierbar sind, bieten die *Semantic Recommender* den Vorteil einer modularen und austauschbaren *Wissensbasis*. Wenn diese auf etablierte Standards setzt und aus externen Quellen zu beziehen ist, entfällt beim Einsatz eines *Semantic Recommenders* der fehleranfällige Prozess des Designs und der Wartung seiner *Wissensbasis*. Oftmals sind *Wissensbasen* frei verfügbar, so dass damit das kostenintensive Verfahren (siehe Kapitel 3.1.5) der Erzeugung komplett entfällt. Weiterhin können sie auch unabhängig von den sie einsetzenden Systemen, von Domänenexperten gepflegt und in ihrer Qualität immer weiter verbessert werden.

Obwohl viele der bisherigen Grundtypen der *Recommender Systeme* bereits verschiedene Formen und Grade der *Semantik* beinhalten, wurde diese bisher noch nicht im Genaueren bestimmt. Im folgenden Kapitel wird daher eine Klassifikation für den Grad der *Semantik* vorgestellt, anhand derer die von verschiedenen Architekturen verwendeten Repräsentationen von Benutzern und Elementen analysiert werden.

Kapitel 4

Semantik in Recommender Systemen

Das Benutzermodell und die Elementrepräsentationen sowie Benutzerprofile (im Folgenden unter dem Begriff *Recommender Entitäten* zusammengefasst) spielen eine zentrale Rolle in den *Recommender Systemen*. Dabei ist das Ziel der Benutzermodellierung, die vom Benutzer angegebenen Daten sowie Interaktionen mit dem System in eine erfassbare Beschreibung seiner subjektiven Vorlieben und Interessen zu überführen. Ausgehend von den aufgenommenen Informationen über den Benutzer sowie den in den Repräsentationen enthaltenen Elementeigenschaften, wird eine dem *Recommender* verständliche Abbildung des Benutzers im System erzeugt, das sogenannte Benutzerprofil. Anhand des den Benutzer beschreibenden Profils sowie den Elementrepräsentationen der von ihm noch unbetrachteten und unbewerteten Elemente generieren die verschiedenen Verfahren Vorschläge für neue Elemente, welche dann dem Benutzer präsentiert werden. Von daher ist es entscheidend, eine aussagekräftige Repräsentation des Benutzers zur Verfügung zu haben, welche seine Interessen und Vorlieben präzise erfasst und dem System zugänglich macht.

Neben den Benutzern müssen auch die Elemente im System differenziert und in ihren Eigenschaften aussagekräftig beschrieben sein. Nur so können diejenigen Elemente bestimmt werden, die am Besten zu den Wünschen und Interessen eines Benutzers passen. Wie bereits in Tabelle 2.2 beschrieben, benötigen die Grundsysteme der *Recommender* eine bestimmte Mindestmenge an Informationen über die *Recommender Entitäten*. Werden sie mit einem weiteren System zu einem *Hybrid Recommender* erweitert (siehe Kapitel 2.2), so müssen die verwendeten Repräsentationen der *Recommender Entitäten* den eingesetzten Verfahren angepasst werden. Dabei kann es notwendig sein, Benutzerprofile und Elementrepräsentationen mit einem höheren Gehalt an Informationen zu verwenden, und diese den Verfahren zugänglich zu machen.

Diese Steigerung kann unter der Verwendung der Methoden und Techniken des *Semantic Webs* erreicht werden. Für die in Kapitel 3.2 vorgestellten *Semantic Recommender* konnte gezeigt werden, dass ihre generierten Vorschläge in Bezug auf die Qualitätsmaße denen der Grundsysteme überlegen sind. Dies lässt sich auf die Benutzung von Domänenwissen in den Repräsentationen der *Recommender Entitäten* und ihre dadurch ermöglichte genauere Beschreibung zurück führen. Im Folgenden werden die Modelle zur Erfassung der Benutzerdaten beschrieben und die verschiedenen Repräsentationsformate der *Recommender Entitäten* nach dem Grad ihrer Aussagekraft gestaffelt analysiert.

4.1 Benutzermodelle

Die Benutzermodelle dienen der Erfassung von Daten über den Benutzer, um dann aus ihnen Rückschlüsse auf seine Interessen und Vorlieben ziehen zu können. Dazu werden diese Daten von der *Usermonitoring und -profiling*-Komponente analysiert (siehe Abbildung 2.2) und in das jeweilige Repräsentationsformat des Systems überführt. Es können zwei unterschiedliche Arten von Benutzermodellen identifiziert werden, welche jeweils auf verschiedenen Arten von Benutzerdaten basieren und dadurch Benutzer und ihre Interessen jeweils auf eine bestimmte Weise erfassen [Brusilovsky and Millán, 2007]. Eine Modellierung kann erfolgen anhand:

- vordefinierter *stereotyper* Cluster ähnlicher Benutzer, in denen der Benutzer anhand seiner Daten wie beispielsweise dem Alter, dem ausgeübten Beruf sowie weiteren Daten einsortiert wird;
- einer individuellen Betrachtung eines Benutzers, basierend auf seinen *expliziten* Angaben über gewünschte Elementeigenschaften, oder einer Analyse seiner Interaktion mit dem System.

Während das erste Modell einen Benutzer als Teil einer Gruppe, bestehend aus Individuen mit ähnlichen Eigenschaften, betrachtet (sogenannte *stereotype* Modellierung), ermöglicht das andere Modell eine auf den jeweiligen Benutzer abgestimmte individuelle Anpassung des Systems. Diese basiert auf den gewünschten Eigenschaften der vorzuschlagenden Elemente oder den speziellen Vorlieben des Benutzers (sogenannte *feature*-basierte Modellierung). Beide Modelle können einzeln oder auch in Kombination verwendet werden, um dann ein *hybrides* Benutzermodell zu erzeugen [Ardissono et al., 2001; Ardissono et al., 2004].

Neben den Informationen über den Benutzer, welche direkt durch sein Verhalten oder aus seinen Angaben ermittelt werden, können Informationen über ihn auch mittels eines *Crawlers* automatisch aus externen Quellen ohne sein weiteres Eingreifen bezogen werden [Pazzani, 1999; Middleton et al., 2004]. Die gesammelten Daten werden von der *Datenintegrations*-Komponente eines *Recommenders* aufgearbeitet und der *Usermonitoring und -profiling*-Komponente zur weiteren Verarbeitung und Interpretation zu Verfügung gestellt.

Im Folgenden werden beide Arten von Modellen vorgestellt, wobei die *feature*-basierte Modellierung weiter eingeteilt werden kann in eine Modellierung durch *explizite* Angaben des Benutzers und in *adaptive* Modelle, welche die Vorlieben des Benutzers anhand seiner Interaktion mit dem System erfassen.

4.1.1 Stereotype Modelle

Stereotype Modelle stellen einen der ältesten Ansätze der Benutzermodellierung dar und wurden in den ersten *adaptiven* Systemen genutzt [Rich, 1979]. In diesem Modell wird versucht, alle Benutzer mit ähnlichen Eigenschaften, wie beispielsweise identischen demografischen Daten, in Gruppen einzusortieren. Diese Gruppen werden als *Stereotypen* oder auch als *stereotype* Cluster bezeichnet und beinhalten nur Benutzer beispielsweise einer einzelnen Berufsgruppe oder eines Alters. *Stereotypen* werden im Vorfeld in einem vom eigentlichen *Recommender* unabhängigen

Prozess definiert, was durch verschiedene Methoden erreicht werden kann. So können *Stereotypen* manuell definiert [Rich, 1979] oder aber durch eine Analyse im Vorfeld durchgeführter statischer Erhebungen von Daten, wie beispielsweise dem Kaufverhalten der Benutzer [Krulwich, 1997] oder dem Fernsehverhalten [Kurapati and Gutta, 2002; Ardissono et al., 2004], erhoben werden.

Jedoch funktionieren *stereotype* Modelle nicht ohne Einschränkung. So können für Benutzer, deren subjektive Vorlieben von den vorgegebenen Interessen ihres zugewiesenen *Stereotyps* abweichen, keine hinreichend guten oder gar keine gültigen Vorschläge für passende Elemente erzeugt werden. Weiterhin ist es für einen Benutzer durchaus nicht ungewöhnlich, dass sich im Laufe des Zeitraums der Benutzung seine demografischen Daten ändern, so dass sein neu zugewiesener *Stereotyp* nicht die oftmals gleich gebliebenen Interessen widerspiegelt. Dies ist durch das Verhalten der *stereotypen Modellierung* bedingt, da alle Benutzer eines *Stereotyps* identisch vom System behandelt werden und keine auf den einzelnen Benutzer direkt abgestimmte *Personalisierung* stattfindet. Das System betrachtet dabei die *Stereotypen* als Ganzes, ohne dabei die subjektiven Vorlieben und Interessen des einzelnen Benutzers im Speziellen zu berücksichtigen [Tang and McCalla, 2003].

4.1.2 Feature-basierte Modelle

Feature-basierte Modelle erzeugen ein Benutzerprofil anhand der angegebenen oder ermittelten Vorlieben eines Benutzers. Im Gegensatz zu den *stereotypen* Modellen werden hier die Benutzer als einzelnes Individuum modelliert, so dass eine individuell auf den einzelnen Benutzer abgestimmte Generierung von Vorschlägen ermöglicht wird.

Explizite Modelle

Explizite Modelle stellen die einfachste Ausprägung eines *feature*-basierten Modells dar. In diesem Modell werden *explizit* vom Benutzer angegebene Informationen über seine bevorzugten Themengebiete, speziell gewünschte Inhalte oder Eigenschaften der Elemente erfasst. Diese Daten gibt der Benutzer im Regelfall bei der erstmaligen Verwendung des Systems an. Auch wenn er sie im späteren Verlauf ändern kann, bedeutet dies einen erhöhten Aufwand in seiner Interaktion, so dass Aktualisierungen nur selten durchgeführt werden.

Mit *expliziten* Angaben bezüglich der Interessen eines Benutzers kann sein Profil jedoch bei der Erzeugung initialisiert werden. Daraus kann eine einfache, wenn auch zunächst noch ungenaue, Repräsentation des Benutzers erstellt werden, welche zur Erzeugung der Vorhersagen von den *Recommendern* von Anfang an benutzt werden kann [Moukas, 1997]. Somit existiert ein einfaches Verfahren, um das in Kapitel 2.1.1 beschriebene *New User*-Problem zu lösen oder die Lernphase des Systems zu beschleunigen.

Auch wenn statische Befragungsmodelle, wie beispielsweise der *Analytische Hierarchieprozess (AHP)* [Saaty, 1980], bisher kaum in Kontext der *Recommender Systeme* verwendet werden, lassen sich durch diese Verfahren die Vorlieben eines Benutzers besser erfassen, als wenn dieser seine *expliziten* Angaben isoliert in das System einträgt. Im *Analytischen Hierarchieprozess* werden aus einer Kombination von Antworten die Vorlieben eines Benutzers in einem mehrstufigen Verfahren bestimmt. Dazu werden dem Benutzer, nachdem er seine ersten Angaben selbst

4.1. Benutzermodelle

getätigt hat, mehrere Fragen gestellt. Diese sind in einer vom Benutzer nicht bemerkten Weise redundant und dienen dazu, seine Angaben zu gewichten und in Relation zueinander zu setzen. Anschließend werden sie vom Verfahren analysiert, um so ein genaues Abbild der Vorlieben zu erzeugen. Aufgrund der Redundanzen in der Befragung können bei diesem Ansatz gegebenenfalls auftretende Inkonsistenzen oder Ungenauigkeiten in den Antworten ausgeschlossen werden

Adaptive Modelle

Adaptive Modelle lernen im Laufe ihrer Verwendung, anhand der Interaktionen des Benutzers mit dem System, seine subjektiven Vorlieben und Wünsche kennen und bilden die Grundlage, auf der die meisten *Recommender* Systeme Informationen über die Vorlieben eines Benutzers erfassen. Durch die *explizit* vorgenommenen Bewertungen auf einzelne Elemente oder auch durch die Interpretation des Benutzerverhaltens als *implizite* Bewertung, werden seine subjektiven Interessen abgeleitet. Anhand dieser Informationen werden die passenden Elemente für den Benutzer bestimmt.

Die ersten *Recommender* verwendeten ausschließlich *explizite* Bewertungen. Dabei bewertet ein Benutzer nach der Betrachtung eines Elementes mittels einer diskreten Skala [Pazzani et al., 1996] oder durch eine einfache Angabe des Nutzens mit ja oder nein [Resnick et al., 1994], ob oder inwieweit es seinen subjektiven Interessen entspricht. Im Gegensatz zur *expliziten* Formulierung einer speziellen Interesse stellen *explizite* Bewertungen vielmehr die subjektive Meinung zu einem einzelnen Element als Ganzes dar, ohne dabei bestimmte Eigenschaften im Speziellen zu betrachten. Erst in Kombination mit vielen weiteren Elementbewertungen und deren Interpretationen bildet die Menge von Bewertungen die Vorlieben eines Benutzers ab. Jedoch verlangen die *expliziten* Bewertungen vom Benutzer einen zeitlichen Mehraufwand, den er unter Umständen nicht bereit ist, aufzuwenden. Neben dem zeitlichen Aufwand spielen oftmals auch Bedenken bezüglich des Datenschutzes, bedingt durch die Angabe persönlicher Informationen eine Rolle. Aus diesen Gründen kann es problematisch sein, genug Bewertungen zu erfassen. Erschwerend kommt hinzu dass der einzelne Benutzer seinen persönlichen Vorteil im Vorfeld nicht abschätzen kann [Grudin, 1994; Baudisch and Brueckner, 2002].

Aus der Interaktion eines Benutzers mit dem System werden Rückschlüsse auf seine Vorlieben gezogen, um daraus *implizite* Bewertungen zu erhalten. Einen Überblick über die verschiedenen Aktionen des Benutzers, die dabei in Betracht gezogen werden können, schaffen die Arbeiten von [Oard and Kim, 1998] und von [Nichols, 1998]. So konnte beispielsweise eine Korrelation zwischen der Lesezeit eines Textes und der subjektiven Bewertung des Benutzers beobachtet werden [Morita and Shinoda, 1994; Konstan et al., 1997]. Weiterhin kann auch aus dem Kaufverhalten eines Benutzers in einem *Online-Shop* auf seine Vorlieben geschlossen werden. Während der Einkauf eines Produktes ein Indikator für eine Vorliebe sein kann, ist die Rückgabe eines Produktes ein sicheres Zeichen dafür, dass dieses den Wünschen des Kundens nicht entsprach [Schafer et al., 1999; Schafer et al., 2001]. Eine vereinfachte Version dieses Verfahrens kommt beim *Online-Shop* Amazon.com^{TM30} zum Einsatz, die jedoch keine negativen Bewertungen unterstützt. Durch Betrachtung des Inhaltes des Einkaufswagens eines Benutzers werden

³⁰<http://www.amazon.com> Zugriff: 22. Mai 2011

ihm zusätzliche Produkte angeboten, die oftmals in Kombination mit den aktuellen Produkten gekauft wurden und damit von Interesse sein können. [Burke, 2000] schlägt ein Verfahren vor, welches auf der Analyse des Browsingverhaltens des Benutzers auf einer Webseite basiert. Hierbei wird der Startpunkt der Interaktion durch eine *explizite* Anfrage an ein Element gesetzt. Dies wird als eine positive Bewertung auf das Element aufgefasst, da der Benutzer dieses Element anscheinend kennt und daher für positiv befindet. Alle Elemente, die der Benutzer im Laufe seiner Suche betrachtet, von denen aus er dann aber weiter sucht, werden als negativ bewertet, da diese seine Kriterien anscheinend nicht erfüllen. Erst das letzte betrachtete Element erhält wieder eine positive Bewertung, weil der Benutzer damit sein Suchziel erreicht hat und mit dem Ergebnis zufrieden scheint.

Implizite Bewertungen verhalten sich vollkommen transparent zum Benutzer, da diese nicht wahrnehmbar im Hintergrund gesammelt werden und keinerlei zusätzlichen Aufwand des Benutzers für ihre Erzeugung erfordern. Zwar wird auch von ihnen eine Form der Benutzererkennung benötigt, die jedoch leicht in Form eines einfachen Login-Mechanismus realisiert werden kann. Dieser wird normalerweise von der *Personalisierungstechniken* verwendenden Seiten bereitgestellt. Da der Benutzer sich nur auf dieser einen Seite bewegt, können alle seine Aktionen erfasst werden, ohne dass es erforderlich ist, zusätzliche Software zu installieren oder einen speziellen Proxy-Server³¹ zu verwenden [Gauch et al., 2007].

Explizite Bewertungen generieren ein genaueres Abbild der persönlichen Interessen als die *impliziten* Bewertungen [Quiroga and Mostafa, 1999; Zanker and Jessenitschnig, 2009], verlangen jedoch vom Benutzer einen Mehraufwand bei der Benutzung der Webseite. Allerdings erhöht dieser Aufwand die von den Betreibern von Informationsportalen erwünschte subjektive Loyalität des Benutzers gegenüber dieser Webseite in hohem Maße. Würde der Benutzer den *Online*-Shop wechseln, so müsste er denselben Aufwand erneut betreiben [Schafer et al., 2001].

Neben dem Aufbau der Loyalität aufgrund des Zeitaufwands, den der Kunde beim Training seines Profils investiert hat, bietet *adaptive* Model noch einen weiteren Vorteil. Wie bereits in Kapitel 1.1 beschrieben, fühlt sich ein Benutzer aufgrund der im Verlauf seiner Verwendung besser werdenden Vorschläge des *Recommenders* auch immer besser „beraten“, und scheut daher den Aufwand, ein neues System anzulernen.

4.1.3 Vor- und Nachteile der Modelle

Feature-basierte Modelle weisen bei ihrem Einsatz in einem *Personalisierung* anbietenden System gegenüber den *stereotypen* Modellen den Vorteil auf, dass sie jeden Benutzer im Einzelnen betrachten. Nur ein solches Verfahren ermöglicht eine auf die subjektiven Vorlieben einer einzelnen Person abgestimmte Erzeugung der Vorschläge. Insbesondere *adaptive* Modelle bieten dabei den Vorteil eines lernenden Profils, welches im *stereotypen* Modell nicht in diesem Maße möglich ist. Allerdings weisen die *adaptiven* Modelle ein Problem auf, dass für alle lernenden Systeme gilt. In der Anfangsphase, in der zunächst ein Mangel an Informationen über die Vorlieben eines einzelnen Benutzers besteht, können sie kaum hinreichend „gute“ Vorhersagen für ihn erzeugen.

³¹Ein Proxy-Server ist ein Rechner im Netzwerk, der von der einen Seite Anfragen annimmt und dann über seine eigene Adresse eine Verbindung zur anderen Seite herstellt.

4.2. Repräsentationen von Benutzern und Elementen

Die Kombination eines *stereotypen* Modells mit einem *feature*-basierten Modell und dabei insbesondere dem *adaptiven* Modell, stellt eine mögliche Lösung für das *New User*-Problem dar. So können die wenigen Daten über den Benutzer, die für das *stereotype* Modell benötigt werden, bereits ausreichen, um mit den abgeleiteten Informationen des *stereotypen* Modells das *adaptive* Modell zu initialisieren [Kurapati and Gutta, 2002; Tsiriga and Virvou, 2003; Ardissono et al., 1999]. Somit können erste Vorhersagen erzeugt werden, welche in der Anfangsphase eine bessere Qualität aufweisen, als dies unter alleiniger Verwendung eines uninitialisierten *adaptiven* Modells der Fall wäre.

4.2 Repräsentationen von Benutzern und Elementen

Die einzelnen in Kapitel 2.1 und 3.2 vorgestellten *Recommender Systeme* verwenden in ihrem Betrieb verschiedene Repräsentationen von Benutzerprofilen, welche in ihrer Struktur jeweils eine unterschiedliche Aussagekraft aufweisen und jeweils anders generiert werden. *Recommender Systeme* erzeugen eine auf dem *feature*-basierten Benutzermodell beruhende Repräsentation des Benutzers, welche seine Interessen abbildet. Um diese Interessen zu bestimmen, werden die Elemente zuerst durch die *Datenintegrations*-Komponente in das Format des Systems überführt und dann in der *Datenhaltungs*-Komponente für den späteren Zugriff gespeichert. Die ermittelten Eigenschaften der Elemente werden von der *Usermonitoring und -profiling*-Komponente verwendet, um die Benutzerprofile zu erzeugen und im Laufe ihrer Verwendung zu aktualisieren. Auch die Benutzerprofile werden in der *Datenhaltungs*-Komponente persistent gespeichert (siehe Abbildung 2.2).

Gerade die Repräsentationen, welche Informationen über die Bedeutung der Eigenschaften der Elemente oder deren Relationen zueinander unter Zuhilfenahme einer *Wissensbasis* verwenden, weisen große Vorteile gegenüber den einfacheren Formaten auf (siehe Kapitel 3.2). Mit solchen *semantischeren* Repräsentationen lassen sich die Interessen der Benutzer oftmals besser und genauer erfassen. Die verschiedenen Repräsentationsformen von Profilen und Elementen sowie ihre jeweiligen Stärken und Schwächen werden im folgenden Abschnitt detailliert betrachtet.

4.2.1 Bewertungsvektoren

Die auf Bewertungsvektoren basierenden Repräsentationen der Benutzer stellen die einfachste Form eines Benutzerprofils dar. Sie bestehen aus einfachen, die Elemente beschreibenden Identifikatoren sowie auf sie vom Benutzer abgegebenen Bewertungen. Dabei dienen die Identifikatoren lediglich zur Unterscheidung der Benutzer sowie der Elemente und bieten an sich keinerlei *Semantik*. Es werden also zunächst alle Benutzer beziehungsweise Elemente identisch betrachtet. Erst durch die Erzeugung eines einfachen Profils für den Benutzer können seine Vorlieben dem System verständlich gemacht werden womit eine, wenn auch aus „*semantischer*“ Sicht geringe, Aussagekraft des Profils erreicht wird. Die Bedeutung des Profils entsteht erst durch die Betrachtung eines den Benutzer repräsentierenden Vektors, der die Werte seiner *explizit* abgegebenen oder aus seinem Verhalten abgeleiteten *impliziten* Bewertungen der von ihm betrachteten Ele-

mente beinhaltet. Diese Form der Repräsentation kommt beim Benutzerprofil-zentrierten *User to User Collaborative Filtering* zum Einsatz [Resnick et al., 1994; Herlocker et al., 2000], bei dem die Benutzer anhand ihres repräsentierenden Bewertungsvektors miteinander verglichen werden. Die generierten Vorschläge entsprechen dann den durch die ähnlichsten anderen Benutzer am höchsten bewerteten Elementen.

Im Gegensatz zum Benutzer-zentrierten *User to User Collaborative Filtering* werden beim Element-zentrierten Ansatz, dem sogenannten *Item to Item Collaborative Filtering*, die Vorschläge nicht anhand ähnlicher Benutzer berechnet. Sie basieren stattdessen auf den Ähnlichkeiten der Elemente untereinander. Dabei findet die *Personalisierung* statt, indem die Ähnlichkeiten der Elemente untereinander verwendet werden, um zur Bewertungshistorie des Benutzers passende Elemente von Interesse zu bestimmen [Sarwar et al., 2001; Linden et al., 2003]. Diese Ähnlichkeiten errechnen sich anhand der auf sie abgegebenen Bewertungen. Grundlegend ist hierbei die Annahme, dass Elemente mit annähernd identischen Bewertungen auch annähernd identisch in ihrer Bedeutung sind [Sarwar et al., 2001].

	Aliens	Matrix	Romeo & Julia	The Beach	Terminator	Titanic
Alice	-	-	4	3	1	5
Bob	3	5	1	-	4	1
Eve	2	4	-	4	2	3

Abbildung 4.1 – Beispiel einer Sammlung von Bewertungsvektoren (Bewertungsmatrix)

Abbildung 4.1 zeigt mehrere auf Bewertungsvektoren basierende Profile. Diese Vektoren werden aufgrund ihrer Struktur in einer Matrix zusammengefasst und in den Systemen als Tabelle gespeichert. Im gegebenen Beispiel werden die Namen der Benutzer und Elemente als Identifikatoren verwendet. In realen Systemen sind dies normalerweise automatisch zugewiesene Zahlen, um eine Eindeutigkeit der Identifikatoren innerhalb des Systems zu erreichen. Eine weitere Möglichkeit stellt die Verwendung von *URIs* als Identifikationsmechanismus dar. Gerade ihre globale Eindeutigkeit (siehe Kapitel 3.1.1) ermöglicht eine eindeutige Referenzierung auch zwischen verschiedenen Systemen, das Format hat in den *Recommendern* aber zur Zeit nur eine geringe Verbreitung.

Obwohl die Form der Repräsentation anhand von Bewertungsvektoren aufgrund ihrer einfachen Struktur lediglich eine sehr geringe Aussagekraft besitzt, liegt in genau dieser Einfachheit eben ihre Stärke. Durch die Abbildung jeder beliebigen Art eines Elementes auf einen einfachen Identifikator, ist kein aufwändiger Prozess nötig, um die repräsentierenden Eigenschaften und Informationen aus einem Element zu extrahieren. Dadurch ist dieses Format sehr gut geeignet in Kombination mit den entsprechenden Algorithmen, schwer zu analysierende Daten wie Musik oder Filme zu repräsentieren. Dies konnten die verschiedenen *Collaborative Filtering* verwendenden *Recommender*, welche diese einfache Repräsentation einsetzen, beim Einsatz solch „schwieriger“ Daten zeigen [Herlocker et al., 2000].

4.2.2 Vector Space Modelle

Das *Vector Space Modell* (*VSM*) repräsentiert Elemente und Benutzer in Form von Vektoren bestehend aus Tupeln von Termen und deren Signifikanz ausdrückenden Werten. *Schlüsselworte*, auch *Keywords* genannt, sind den Inhalt repräsentierenden Terme der textuellen Beschreibung eines Elementes. Dies basiert auf der Annahme, dass jedes *Schlüsselwort* einer Eigenschaft des Textes entspricht. Die Werte für die Signifikanz eines *Schlüsselwortes* können dabei in unterschiedlichen Formaten vorliegen. Im einfachsten Format wird die Signifikanz eines Wortes für ein Element durch einen booleschen Wert ausgedrückt, der lediglich die Anwesenheit oder Abwesenheit eines Wortes anzeigen kann, nicht aber auf seine genaue Bedeutung für das Element eingehen kann.

Eine verbesserte Repräsentation kann durch die Verwendung einer Gewichtung erzeugt werden, durch welche diejenigen Terme bestimmt werden, die am Ehesten die Bedeutung des Elementes wiedergeben. Dazu werden aus der Menge der extrahierten Worte die für den Inhalt „unwichtigen“ Worte anhand einer sprachabhängigen Liste sogenannter *Stopwörter* entfernt. *Stopworte* sind Worte wie Artikel, Pronomen, Präpositionen oder häufig verwendete Verben, welche zwar der Satzbildung dienen, jedoch keinerlei Signifikanz für die Bedeutung des Inhalts des gesamten Textes aufweisen. Die restlichen Worte werden anschließend mit einem *Stemming* genannten Verfahren auf ihren Wortstamm reduziert, um durch Flexion oder Konjugation auftretende Redundanzen der extrahierten Worte zu reduzieren [Porter, 1980]. Die übrig gebliebenen und anhand ihrer Signifikanz gewichteten Worte bilden dann einen das Element repräsentierenden Vektor [Salton et al., 1975]. Zur Gewichtung der Terme können verschiedene Metriken verwendet werden, so zum Beispiel die weit verbreitete *TF-IDF*-Metrik [Salton, 1989]. Dabei wird ein in einem Element gehäuft auftretender, im Korpus aller Elemente jedoch selten vorhandener Term als repräsentativer Term für den Inhalt des Elementes höher gewichtet wird als ein global häufig auftretender Term. Neben dieser Metrik wird auch die χ^2 -Metrik [Yang and Pedersen, 1997] eingesetzt, welche einen Term danach gewichtet, inwieweit er repräsentativ für die Kategorie des Elementes ist.

Anhand der aus den Elementen ermittelten *Schlüsselworte* werden die Repräsentationen der Benutzer erzeugt. Deren Profile bestehen, genau wie die der Elemente, aus einem Vektor von Worten und deren Gewichtungen. Die Worte des Vektors entstammen dabei den vom Benutzer bewerteten Elementen, wobei sich die Gewichtung aus dem Grad seines Interesses an einem extrahierten Term ergibt und daraus folgend auch an anderen Elementen, die dieses Wort und damit auch diese Eigenschaft aufweisen. Mit der weiteren Verwendung des Systems und der damit verbundenen Betrachtung und Bewertung von immer mehr Elementen wird das Profil des Benutzers um weitere *Schlüsselwörter* aktualisiert. Auch die bestehenden Daten werden dadurch weiter „gefestigt“, so dass der Benutzer im weiteren Verlauf seiner Benutzung des Systems immer genauer repräsentiert werden kann. Neben einer automatischen Analyse und Bestimmung der für den Benutzer relevanten *Schlüsselworte* aus seinen bewerteten Elementen können diese Terme auch selbstständig und *explizit* in das Profil eingegeben werden [Moukas, 1997]. Somit stellt die Verwendung eines *Vector Space Modells* eine Möglichkeit für den Benutzer dar, bestimmte Interessen gezielt anzugeben. Mit den einfacheren Identifikatoren-basierten Repräsentationen ist dies

nicht in diesem Maße möglich, da sie die Benutzerinteressen von speziellen Elementeigenschaften abstrahieren.

Art			
Portrait	Sculpture	Watercolor	Painting
0.6	0.72	0.45	0.33

Sports			
Soccer	Bat	Touchdown	Score
0.88	0.27	0.79	0.33

Music			
Rock	Symphony	Hip Hop	Orchestra
0.15	0.87	0.31	0.63

Abbildung 4.2 – Beispiel eines Vector Space Modell basierten Benutzerprofils

Abbildung 4.2 zeigt die Struktur eines *Vector Space Modell* basiertes Benutzerprofil sowie eine Erweiterung der Abbildung durch mehrere Vektoren. Dabei wird jedes einzelne seiner unterschiedlichen Interessensgebiete jeweils durch einen Vektor mit den entsprechenden Termen und ihren Gewichtungen beschrieben. Würde nur ein einzelner Vektor für zwei unterschiedliche Interessen verwendet werden, so würde dieser zwischen die beiden Gebiete zeigen und damit ein ungenaues oder sogar falsches Abbild erzeugen. Im Gegensatz dazu können Vektoren jedes der unterschiedlichen Gebiete einzeln für sich durch entsprechenden Terme und ihre Gewichtungen beschreiben, was ein genaueres Abbild der Interessen eines Benutzers erzeugt [Chen and Sycara, 1998; Widyantoro et al., 1999]. Hierbei entsprechen die Gebiete nicht im Vorfeld definierten Domänen, sondern werden anhand der Ähnlichkeiten zwischen den einzelnen bewerteten Elementen ermittelt. So werden sehr ähnliche Elemente zu einem einzelnen Vektor zusammengefasst, während bei sehr unähnlichen Elemente davon ausgegangen wird, dass sie auch unterschiedlichen Interessensgebieten angehören. Diese werden dann jeweils durch einen anderen Vektor im Profil repräsentiert.

Bestehen die Elemente nicht aus Texten, sondern aus schwer zu analysierende Daten, wie beispielsweise Bilder, Musik oder sogar Videos, können Textanalyseverfahren nur beim Vorhandensein von textuellen Beschreibungen verwendet werden. Auch wenn bereits Verfahren existieren, welche den Vergleich von Bildern [Schmid and Mohr, 1997; Lowe, 2004] oder Musik [Orio, 2006] anhand technischer Kriterien ermöglichen, so sind Vergleiche dieser Daten anhand der Bedeutung ihres Inhaltes immer noch schwer durchführbar. Erst die Interpretation der Elemente durch Menschen ermöglicht die maschinelle Verarbeitung von Daten beliebigen Inhalts und beliebigem Formats [Shardanand and Maes, 1995]. Eine Lösung dieses Problems stellt die Verwendung von den Inhalt der Elemente beschreibenden *Meta*-Daten, den sogenannten *Tags*, dar. Diese von Menschen erzeugten Informationen in Form von *Schlüsselwörtern* ermöglichen eine Verarbeitung von Multimediadaten auch mit den traditionellen Techniken, da hierbei keine

4.2. Repräsentationen von Benutzern und Elementen

vorherrschende aufwändige maschinelle Analyse des Inhaltes durchgeführt werden muss. Durch die Erfassung der Elementeigenschaften können diese Elemente beliebigen Formats den Benutzern vorgeschlagen werden [Ghani and Fano, 2002; Szomszor et al., 2007; Musial et al., 2008; Cantador et al., 2010]. Jedoch stellt es einen nicht unerheblichen Aufwand dar, alle Elemente des Systems von Menschen bearbeiten und auszeichnen zu lassen. Dies kann mittels *Collaborative Tagging*, einem Prozess, in welchem eine große Anzahl an Benutzern öffentlich an der Auszeichnung von Daten mit *Meta*-Daten beteiligt ist, erreicht werden [Golder and Huberman, 2006]. Auch wenn diese als *Folksonomien*³² bezeichneten, generierten Daten aufgrund des Fehlens eines gemeinsamen Vokabulars sowie ihrer öffentlichen unkontrollierten Erzeugung chaotisch erscheinen, ermöglichen sie dennoch die Erzeugung und Erfassung einer großen Menge an Informationen über Elemente, welche sonst nur durch Experten in einem kostenintensiven Prozess durchführbar wäre. So stellte Steward Butterfield, der Begründer von Flickr, 2004 die These auf, dass *Folksonomien* 90 % des Wertes einer formalen Taxonomie aufweisen, dabei allerdings 10 mal simpler sind (zitiert von [Mathes, 2004]).

Diese Zahlen sind sicherlich kritisch zu betrachten, zeigen jedoch eine gewisse Tendenz. Nach [Mathes, 2004] würden Benutzer eines öffentlichen Systems wohl niemals zur Auszeichnung der Elemente eine formale und komplexe Taxonomie mit einem eindeutigen schwergewichtigen Vokabular verwenden. Da hierbei die Einstiegshürden bei der Benutzung zu hoch wären, würde ein solches System kaum eine breite Verwendung finden. Die größten *Collaborative Tagging* verwendenden Webseiten (Flickr³³, Last.fm³⁴, IMDB³⁵) haben allerdings mehrere hunderttausende Benutzer, was sich durch die Einfachheit und Offenheit dieser Systeme und damit hohe Akzeptanz bei ihren Benutzern erklären lässt. Gerade die Verwendung durch eine solch große Anzahl an Menschen stellt den hohen Wert des *Collaborative Taggings* dar (vgl. *Metcalfes Law* [Metcalf, 1995]). Denn es ist besser, „irgendwelche“ zusätzlichen Informationen zu haben, als gar keine [Hendler and Golbeck, 2008]. Die Daten des *Collaborative Taggings* haben dahin gehend einen besonderen Wert, dass sie nicht durch Maschinen, sondern durch Menschen erzeugt werden. Diese verstehen den Inhalt eines Elementes durch ihre subjektive Analyse meist besser, als es maschinelle Verfahren normalerweise erreichen können. Jedoch treten auch bei diesem Verfahren, bedingt durch sein offenes und durch Menschen erzeugtes Vokabular, die Probleme auf, welche generell bei der Verarbeitung von natürlicher Sprache aufgrund des Vorhandenseins von Homonymen und Synonymen vorkommen [Rohland and Streibel, 2009].

Homonyme stellen diejenigen Worte dar, welche in einer konkreten natürlichen Sprache mehrere Bedeutungen aufweisen. So beschreibt beispielsweise das Wort „Bank“ verschiedene Konzepte, zum einen das Gebäude, zum anderen das Möbelstück. Synonyme bezeichnen unterschiedliche Worte, welche das gleiche Konzept beschreiben und damit auch die gleiche Bedeutung aufweisen, wie beispielsweise die Worte „Bank“ und „Kreditinstitut“. Während die Homonyme eines Textes gegebenenfalls unter Berücksichtigung der übrigen Terme in der Repräsentation aufge-

³²der Begriff Folksonomie ist ein von Thomas Vander Wal 2007 eingeführtes Kunstwort aus der Kombination von *engl. folk*=Menschenmenge und *engl. taxonomy*=Taxonomie; zitiert von [Angeletou et al., 2007]

³³<http://www.flickr.com> Zugriff: 23. August 2011

³⁴<http://www.lastfm.de> Zugriff: 23. August 2011

³⁵<http://www.imdb.com> Zugriff: 22. August 2011

löst werden können, ist die Erfassung von Synonymen so nicht möglich [Golder and Huberman, 2006].

Eine weitere Einschränkung der Aussagekraft des *Vector Space Modells* ist durch sein „flache“ Struktur bedingt. Durch die Abbildung auf einen einfachen Vektor gehen die *semantischen* Zusammenhänge zwischen den extrahierten Termen verloren, welche normalerweise durch ihr gemeinsames Auftreten in einem Element vorhanden wären. Unterschiedliche Terme stehen aufgrund dieser Zusammenhänge normalerweise nicht orthogonal aufeinander, werden aber beim *Vector Space Modell* aufgrund der Form der Repräsentation so abgebildet [Furnas et al., 1987], auch wenn sie eine gewisse *semantische* Ähnlichkeit aufweisen.

Nichtsdestotrotz kommt das *Vector Space Modell* in vielen Bereichen der informationsverarbeitenden Systeme wie Suchmaschinen und insbesondere den in Kapitel 2.1.2 vorgestellten *Content-Based Recommendern* zum Einsatz [Lang, 1995; Pazzani et al., 1996; Balabanovic and Shoham, 1997]. Sie ermöglichen trotz ihrer einfachen Struktur eine zuverlässige Vorschlagsgenerierung für Elemente, welche die gleichen Terme wie die bereits betrachteten und für interessant befundenen Elemente aufweisen [Brusilovsky and Millán, 2007]. Allerdings ermöglichen sie es nicht, den Benutzern auch solche Elemente vorzuschlagen, deren Eigenschaften noch nicht vom Benutzerprofil erfasst sind. Somit erzeugt das System ausschließlich Vorhersagen, die unter Umständen eine zu große Ähnlichkeit zu den bisherigen und für interessant befundenen Elementen zeigen. Dieses Problem wird im Kontext der *Recommender Systeme* als *Überspezialisierung* bezeichnet (siehe Kapitel 2.1.2) und führte zu Entwicklung neuer Klassen von Repräsentationen, den *taxonomischen* sowie den *semantischen* Repräsentationen. In beiden Repräsentationen werden die Worte nicht bloß als einfache Zeichenketten, sondern ihrem Sinn nach aufgefasst und verarbeitet. Weiterhin werden auch die Relationen zwischen den Worten berücksichtigt sowie die Möglichkeiten gegeben, Generalisierungen und Spezialisierungen der Worte vorzunehmen.

4.2.3 Taxonomische Repräsentationen

Taxonomische Repräsentationen von Elementen und Benutzern werden oftmals analog zu den *Vector Space Modellen* als ein Vektor mit gewichteten Termen dargestellt, welche die Signifikanz einer Eigenschaft im Element oder den Grad des Interesses des Benutzers an diesem Eintrag wiedergeben [Tan et al., 1998; Cantador and Castells, 2009]. Im Gegensatz zu den *Vector Space Modellen* entsprechen die Terme des Vektors bei den *taxonomischen* Repräsentation jedoch nicht den extrahierten Worten, sondern den vordefinierten Konzepten eines externen Wissensmodells.

Konzepte stellen eine abstrakte Darstellung einer Sache dar und sind deshalb nicht auf die einfache Zeichenkette eines Wortes reduzierbar. Sie sind vielmehr die Vorstellung einer Begrifflichkeit, die eine Sache durch eine Sammlung von speziellen Eigenschaften und Sachverhalten als Ganzes ausmacht. Wenn sich zwei Personen unterhalten, benutzen sie zwar einzelne Worte zur Kommunikation, die jedoch als Konzepte zu verstehen sind, so dass diese vom Kommunikationspartner in ihrem Sinn und nicht lediglich als einfaches Wort begriffen werden. Solche Konzepte müssen bei ihrer Verwendung eindeutig voneinander unterschieden werden können, was im *Semantic Web* durch den Einsatz global eindeutiger *URIs* als Identifikationsmechanismus sicher gestellt ist.

4.2. Repräsentationen von Benutzern und Elementen

Es lassen sich unter Verwendung von Konzepten die Mehrdeutigkeiten von Worten in der natürlichen Sprache aufgrund der dort auftretenden Homonyme und Synonyme auflösen, da diese so eindeutig in ihrer Bedeutung interpretierbar sind. Zur richtigen Interpretation der Worte in ihre Konzepte werden *lexikalische* Datenbanken benötigt, welche die Worte in ihren Sinn erfassen und sie damit auf korrekte Konzepte abbilden. Die am häufigsten verwendete *lexikalische* Datenbank für die englische Sprache ist WordNet³⁶ [Miller, 1995; Fellbaum, 1998]. WordNet, in der im September 2011 aktuellen Version 3.0, beinhaltet über 150.000 Worte, die in Nomen, Verben, Adjektive und Adverbien klassifiziert sind. Weiterhin sind die Worte anhand ihrer Bedeutung in 117.000 sogenannte *Synsets* eingeteilt. Ein *Synset* stellt eine Sammlung von unterschiedlichen Worten mit gleicher Bedeutung dar und bildet damit die Menge an Worten, die in der natürlichen Sprache das Konzept einer Sache ausdrücken. Die Vorteile der Verwendung einer *Synset*-basierten Repräsentation sind nach [Magnini and Strapparava, 2004]:

- Homonyme Terme können aufgelöst und in ihrer korrekten *semantischen* Interpretation verwendet werden, wodurch eine höhere *Precision* erreicht werden kann. So hat beispielsweise für einen am Finanzwesen interessierten Benutzer der Eintrag des Wortes „Bank“ in einem Katalog für Möbel keine Relevanz.
- Synonyme, die zum selben *Synset* gehören, können die Aussagekraft der Repräsentationen erhöhen, wodurch der *Recall* gesteigert werden kann (sowohl das Wort „Bank“ als auch das Wort „Kreditinstitut“ beziehen sich auf dasselbe Konzept).
- *Synsets* können bei der Verwendung eines mehrsprachigen Korpus von Elementen die Möglichkeit eines Benutzerprofils schaffen, das für alle verwendeten Sprachen einsetzbar ist.

Werden *Synsets* anstelle von einfachen Worten verwendet, so kann die Qualität der Ergebnisse stark verbessert werden. Untersuchungen, welche den Einfluss von *Synsets* auf die *Precision* der Ergebnisse ermittelten, konnten sowohl bei ihrer Verwendung in Repräsentationen [Gonzalo et al., 1998] als auch bei der Erweiterung von Suchanfragen [Navigli and Velardi, 2003] eine Erhöhung der *Precision* von jeweils fast 30 % ermitteln.

Die Bestimmung des jeweils richtigen, zum Wort passenden *Synsets* kann allerdings schwierig sein. Im einfachen Fall hat das Wort nur eine Bedeutung, und das *Synset* kann sofort bestimmt werden. Falls es jedoch mehreren *Synsets* angehört, müssen zusätzliche im Dokument auftretende Worte in Betracht gezogen werden. Aus dem Kontext, in welchem das mehrdeutige Wort auftritt, kann dann auf seine eindeutige Bedeutung geschlossen werden. Hierbei kommen verschiedene Verfahren zum Einsatz, die sich in statistische und linguistische Verfahren einteilen lassen. Bei den statistischen Verfahren werden zuerst die Worte und deren *Synsets* ermittelt, deren Bedeutung eindeutig mit Hilfe einer *lexikalischen* Datenbank bestimmbar sind. Mit Hilfe eines, auf der Struktur der in der Datenbank enthaltenen linguistischen Taxonomie arbeitenden Ähnlichkeitsmaßes [Wu and Palmer, 1994; Jiang and Conrath, 1997; Leacock and Chodorow, 1998; Yang and Powers, 2005] wird dann das *Synset* bestimmt, welches zu den bereits bekannten die höchste *semantische* Ähnlichkeit aufweist [Leacock and Chodorow, 1998]. Linguistische

³⁶<http://wordnet.princeton.edu/wordnet/> Zugriff 18. September 2011

Verfahren gehen einen gänzlich anderen Weg und basieren beispielsweise auf sogenannten *lexikalischen* Ketten. Eine *lexikalische* Kette ist eine Abfolge zusammengehöriger Worte, welche sich über kurze (ein Satz) oder lange Distanzen (gesamter Text) erstreckt. Werden nun diese Ketten anhand der verschiedenen Bedeutungen der Worte gebildet und nur die längsten berücksichtigt, so kann auf die Bedeutung der einzelnen Worte mit Hilfe einer *lexikalischen* Datenbank geschlossen und diese eindeutig zugewiesen werden [Hirst and St-Onge, 1998].

Neben der Eindeutigkeit der Aussagen und der Qualitätsverbesserung der Ergebnisse, die die Verwendung der *Synsets* gegenüber den einfachen *Vector Space Modellen* ermöglicht, werden Konzepte verwendet, um die Elemente anhand einer Taxonomie in Themengebiete³⁷ einzuteilen [Labrou and Finin, 1999; Castells et al., 2007]. Dies geschieht, indem die in den Elementen enthaltenen *Synsets* auf ihre entsprechenden Konzepte in einer externen Taxonomie abgebildet werden, welche die Themengebiete sowie deren beinhaltete Konzepte beschreibt. Dazu können automatische Verfahren [Daudé et al., 2000; Montoyo et al., 2001] oder manuelle Auszeichnungen [Labrou and Finin, 1999] verwendet und daran die Themengebiete der Elemente ermittelt werden. Untersuchungen bezüglich der Qualität einer solchen Bestimmung haben ergeben, dass die Klassifikation eines Elementes in seine entsprechende Domäne anhand eines *taxonomischen* Modells vergleichbar ist mit der von Experten nach der händischen Betrachtung der Elemente getroffenen Entscheidung [Yang and Powers, 2005].

Die Abbildung eines Benutzers wird auf die gleiche Art und Weise wie die der Elemente gebildet [Gauch et al., 2003; Middleton et al., 2004; Ziegler et al., 2004b; Liao et al., 2006]. Das Benutzerprofil wird aus den Konzepten der von ihm für interessant befundenen Elemente erstellt, wobei deren Gewichtungen den Grad seines Interesses angeben. Die Konzepte des Benutzerprofils werden dabei auf die Konzepte der externen Taxonomie abgebildet. Eine Wissenstruktur, die Informationen über die Domäne zur Beschreibung der Elemente und Benutzer bereitstellt, nennt sich das *Domänen-Modell* [Brusilovsky and Millán, 2007]. Dabei ist sie nicht Bestandteil der Repräsentationen selbst, sondern dient vielmehr dazu, die verschiedenen ermittelten Eigenschaften untereinander in Relation zu setzen und diese zu hierarchisieren. Das Benutzerprofil, auch *Overlay-Modell* genannt, entspricht dann der subjektiven Sicht eines Benutzers auf das *Domänen-Modell*.

Abbildung 4.3 zeigt das Beispiel eines auf der Verwendung von Konzepten basierenden *taxonomischen* Benutzerprofils sowie die Baumstruktur des *Domänen-Modells*, in der die Konzepte angeordnet sind. Durch diese Struktur ist es möglich, Generalisierungen oder auch Spezialisierungen der durch Konzepte repräsentierten Interessen vorzunehmen, was mit reinen *Keyword*-basierten Profilen aufgrund ihrer „flachen“ Struktur nicht durchführbar wäre [Bloedorn et al., 1996]. Weiterhin können mit Hilfe der Taxonomie auch eng verwandte, aber noch nicht in den bereits betrachteten Elementen vorkommende, Konzepte der Interessensgebiete des Benutzers ermittelt, sowie deren Ähnlichkeiten zueinander bestimmt werden. Dies ist auch dann möglich, wenn diese sich nicht in der direkten Hierarchie der bereits erfassten Konzepte befinden, wie es bei Geschwisterkonzepten³⁸ der Fall ist [Schickel-Zuber and Faltings, 2006]. Somit kann die

³⁷dies lässt sich auch auf Produkte in einer Produkthierarchie anwenden

³⁸Geschwisterkonzepte bezeichnen Konzepte auf einer Ebene des Baumes mit gemeinsamen Eltern

4.2. Repräsentationen von Benutzern und Elementen

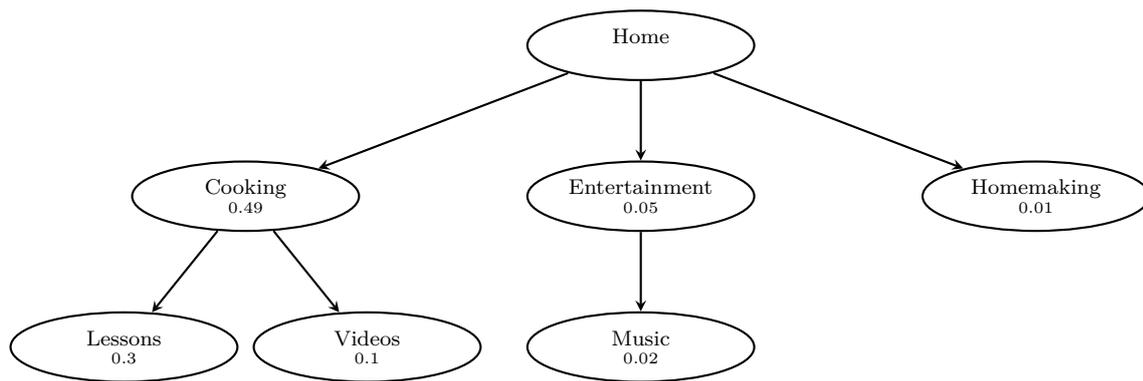


Abbildung 4.3 – Beispiel eines taxonomischen Benutzerprofils

Taxonomie durch die Erweiterungsmöglichkeiten der Profile zur Lösung der *Überspezialisierungs*-Problematik beitragen.

Bei der Modellierung der verwendeten Taxonomie muss besonders sorgfältig vorgegangen werden, da anhand dieser Struktur die Interessen der Benutzer beschrieben und damit die Vorhersagen auf die ihnen vorzuschlagenden Elemente getroffen werden. Je mehr Ebenen in der Taxonomie zur Klassifikation der Konzepte benutzt werden, um so genauer können die Interessen der Benutzer erfasst werden. Jedoch gehen bei der Verwendung der zu vieler Ebenen die Generalisierungen der verschiedenen Interessensgebiete verloren, die aber einen entscheidenden Vorteil der *Konzept*-basierten Repräsentationen darstellen [Gauch et al., 2007]. Der kosten- und zeitintensive Prozess zur Erzeugung einer Taxonomie (siehe Kapitel 3.1.5) kann durch die Verwendung bereits bestehender und etablierter Kataloge und Taxonomien vermieden werden. So nutzen verschiedene Systeme bereits vorhandene Taxonomien wie beispielsweise die Kategoriehierarchien des Open Directory Projects³⁹ [Middleton et al., 2002; Tanudjaja and Mui, 2002] oder der Yahoo! Directories⁴⁰ [Sieg et al., 2004], den Produktkatalog von Amazon.com⁴¹ [Ziegler et al., 2004a; Ziegler et al., 2004b] oder die Kategorien der Wikipedia-Artikel⁴² zur Bestimmung der Domänen von Texten [Achananuparp et al., 2007; Cantador and Castells, 2009].

Weiterhin lässt sich durch die Verwendung von Taxonomien auch eine Steigerung der *Precision* der ermittelten Ergebnisse innerhalb einer Sammlung von Elementen erreichen [Wang et al., 2003]. In einem System für die Arbeitsplatzsuche verbesserte der Einsatz einer Taxonomie die *Precision* um 60 % gegenüber einer einfachen Stichwort-Suche [Bradley et al., 2000]. Ähnliche Ergebnisse konnten auch im Evaluierungssystem von [Finin et al., 2005] gegenüber dem dortigen, lediglich einzelne Worte betrachtenden Baseline-Verfahren ermittelt werden.

Gerade wenn die Elemente und Profile in einem strukturierten Format, wie beispielsweise *RDFS*, beschrieben sind und eine formal definierte Taxonomie zu deren Klassifikation vorliegt, ermöglicht diese Repräsentation in Kombination mit den Techniken des *Semantic Webs* eine starke Verbesserung der Genauigkeit. Durch die Eindeutigkeit der Konzepte und die im Vorfeld

³⁹<http://dmoz.org> Zugriff: 14. August 2011

⁴⁰<http://dir.yahoo.com> Zugriff: 14. August 2011

⁴¹<http://www.amazon.com> Zugriff: 22. Mai 2011

⁴²<http://www.wikipedia.org> Zugriff: 29. Juni 2011

statt gefundene formale Definition der Eigenschaften sowie ihrer Werte entfällt der aufwändige und fehlerbehaftete Prozess, die Elemente zunächst in die Repräsentationsformate zu überführen und deren Eigenschaften zu ermitteln. Weiterhin realisiert der in Kapitel 3.1.4 beschriebene Inferenzmechanismus bei der Verwendung von *RDFS* als Format der Beschreibung eine leichte Umsetzung der erwünschten Generalisierungen und Spezialisierungen.

Der Einsatz von Taxonomien zur Erweiterung von Elementrepräsentationen und Benutzerprofilen stellt eine Lösung für das *Data-Sparsity*-Problem dar, welches auftritt, wenn die im Profil eines Benutzers enthaltenen Konzepte zu wenige Überlappungen mit denen in den übrigen Elementen vorkommenden aufweisen. Mit Hilfe der Baumstruktur kann auf andere, eng verwandte Konzepte geschlossen werden, so dass, auch beim Fehlen gemeinsamer Konzepte, inhaltlich eng verwandte Elemente bestimmt werden können. Zudem können Taxonomien zur Lösung des *New User*-Problems beitragen, da bereits mit den wenigen, im Profil eines neuen Benutzers vorhandenen Konzepten weitere thematisch eng verwandte Konzepte und damit neue Elemente ermittelt werden können.

4.2.4 Semantische Repräsentationen

Semantische Repräsentationen verfolgen einen ähnlichen Ansatz wie die *taxonomischen* Repräsentationen. Es werden ebenfalls Konzepte verwendet, um Elemente und Benutzer zu beschreiben. Im Gegensatz zu den *taxonomischen* Benutzerprofilen wird ein *semantisches* Benutzerprofil jedoch nicht anhand einer vordefinierten Sortierung der Konzepte in ihre Themengebiete gebildet, sondern es wird für jeden Benutzer ein eigenes dynamisches Netzwerk von Konzepten in Form eines *semantischen* Graphen erzeugt. Dabei ermöglicht dieser *semantische* Graph eine dynamische Formulierung der Relationen zwischen den in den Elementen auftretenden Konzepten, wodurch die Interessen eines Benutzers genauer repräsentieren werden. Eine solche Dynamik ist in den *taxonomischen* Profilen nicht möglich, da dort die Konzepte untereinander starr angeordnet sind.

Abbildung 4.4 zeigt ein Beispiel eines *semantischen* Profils. Dabei entsprechen die Knoten des Graphen den Konzepten, welche in den vom Benutzer bewerteten Elementen auftreten. Treten zwei Konzepte gemeinsam in einem Element auf, so wird dies durch eine Kante zwischen diesen beiden Konzepten im Graphen dargestellt. Weiterhin beinhalten die Knoten des Graphen eine Gewichtung, die den Grad des Interesses des Benutzers wieder geben. Aus diesen Gewichtungen leiten sich auch die Gewichte der Kanten ab. Während die ersten Systeme noch einzelne Worte als einen Knoten erfassten [Asnicar and Tasso, 1997], verwenden modernere Systeme die in Kapitel 4.2.3 beschriebenen *Synsets* als Knoten, um die durch sie ermöglichte Verbesserung der Qualität gegenüber der Verwendung von einfachen Worten zu nutzen. Dabei werden diese *Synsets* den Elementen manuell durch Domänenexperten zugewiesen [Magnini and Strapparava, 2004] oder automatisch bestimmt [Micarelli and Sciarrone, 2004].

Semantische Profile werden anhand der extrahierten Konzepte der vom Benutzer betrachteten Elemente erzeugt. Dabei werden die im Element vorkommenden Konzepte mit ihrer Gewichtung sowie der Bewertung des Benutzers als neue Knoten in das Profil hinzugefügt und mit den bereits vorhandenen Knoten der gemeinsam im Element auftretenden Konzepte verbunden,

4.2. Repräsentationen von Benutzern und Elementen

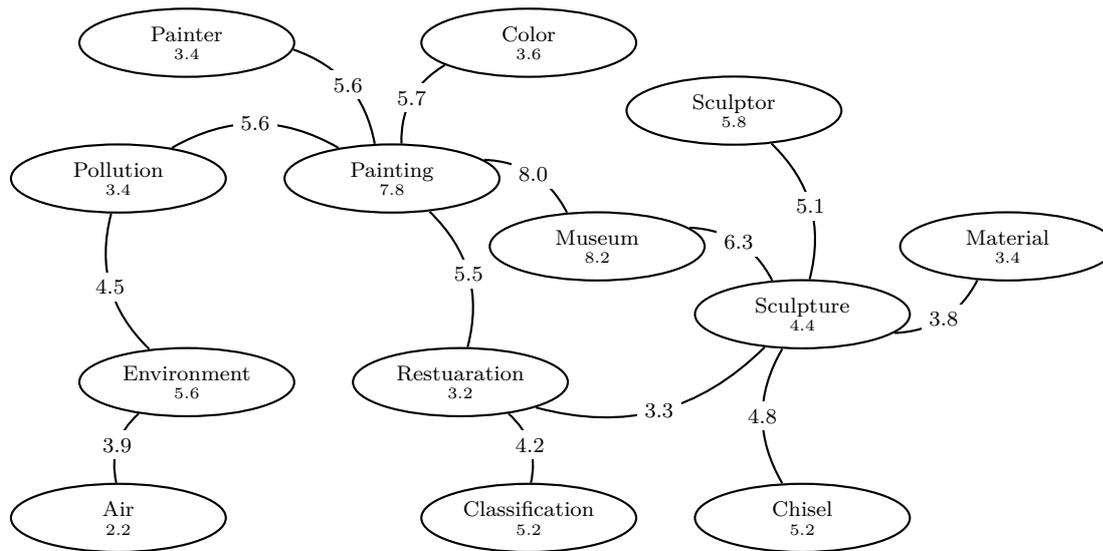


Abbildung 4.4 – Beispiel eines semantischen Benutzerprofils

sofern diese im Graphen noch nicht vorhanden sind. Andernfalls werden die Werte der Knoten entsprechend angepasst und gegebenenfalls neue Kanten in den Graphen eingefügt.

Die *semantische* Repräsentation ermöglicht die Abbildung der Interessen eines Benutzers auf der Ebene der Zusammenhänge einzelner Konzepte, beinhaltet dabei aber ein ähnliches Problem, wie es auch die *Vector Space Modelle* bei der Verwendung eines einzelnen Vektors für alle Interessen aufweisen. Eine solche Repräsentation vermischt verschiedene Interessengebiete zu einem einzigen und verhindert so eine genaue Abbildung des Benutzers. Eine Lösung schafft auch hier der Ansatz, für jedes einzelne Interessensgebiet ein eigenes Netzwerk zu bilden. Im Gegensatz zum *Vector Space Modell* ist es bei den *semantischen* Repräsentationen in diesem Zusammenhang möglich, auch die entsprechenden Domänen der Elemente zu verwenden (siehe Kapitel 4.2.3) und für jede Domäne ein eigenes, die subjektiven Vorlieben des Benutzers repräsentierendes, Netzwerk zu erzeugen.

Neben diesen auf der Bedeutung (*Semantik*) von Worten basierenden Profilen, lassen sich auch *typisierte* Repräsentationen von Elementen aufgrund der eindeutigen Definition ihrer Eigenschaften in die Klasse der *semantischen* Repräsentationen einordnen. Diese Vokabulare sind auf den jeweiligen Typ des Elementes wie beispielsweise Bilder oder Musikstücke angepasst. So ist es mit der Verwendung des Exchangeable Image File Formats (*Exif*)⁴³ möglich, Bild-Dateien im *JPEG*-Format Informationen über den Aufnahmeort, die verwendeten Kamera und vielen weiteren der Aufnahme betreffende technische Angaben zuzuweisen. Ähnliches ist durch die Verwendung von *ID3*-Tags⁴⁴ mit Musik-Dateien im *MP3*-Format möglich. Diese Tags tragen dabei die Informationen über beispielsweise den Interpreten, den Titel, das Albums oder das Genre des Musikstücks. Weiterhin existiert mit Dublin Core (*DC*) [Kunze and Baker, 2007] ein etablierter Standard, um beliebige Dokumente im *World Wide Web* mit zusätzlichen Informationen

⁴³<http://www.exif.org/Exif2-2.PDF> Zugriff: 14. August 2011

⁴⁴<http://www.id3.org/id3v2.4.0-frames.txt> Zugriff: 14. August 2011

wie dem Autor, der verwendeten Sprache oder sogar den Inhalt beschreibenden Stichworten auszuzeichnen.

Durch die im Vorfeld erzeugte und formale Repräsentation entfällt dabei der oftmals mit Fehlern behaftete Prozess der Informationsgewinnung aus den Elementen. Weiterhin können unter Verwendung von Domänenwissen die Elementeigenschaften entsprechend ihrer Bedeutung interpretiert werden, was den Einsatz verbesserter und auf die jeweilige Eigenschaft abgestimmte Methoden beim Vergleich der Elemente ermöglicht [Burke, 2000; Smyth, 2007].

Typisierte Vokabulare lassen sich als Vorgänger der auf den Techniken des *Semantic Webs* basierenden *semantischen* Benutzerprofilen und Elementrepräsentationen auffassen. Diese Art der weiterentwickelten *semantischen* Repräsentationen modelliert die Elemente und Benutzer mittels formaler Beschreibungssprachen wie beispielsweise *OWL* (siehe Kapitel 3.1.5) und erzeugt anhand der Relationen zwischen ihnen ein *semantisches* Netzwerk. Dabei entspricht das Profil eines Benutzers einem subjektiven Ausschnitt des vorhandenen Netzwerkes, bestehend aus seinen bereits bewerteten Elementen [Wang et al., 2008]. Anhand der Relationen der Elemente untereinander können dann, diesen folgend, weitere, zum Profil des Benutzers passende, Elemente im Netzwerk ermittelt werden.

Neben den auf der *Semantik* der Eigenschaften basierenden Verfahren existieren auch solche, die mittels Graph-basierten Algorithmen, anhand der Struktur des durch die Elemente gebildeten Netzwerkes zu den vom Benutzer positiv bewerteten Elemente passende Vorschläge generieren. Die Ähnlichkeit zwischen den Elementen wird dabei entweder durch vorhandene Pfade zwischen diesen [Passant, 2010], durch in den Elementen gemeinsam auftretenden Relationen [Chedrawy and Abidi, 2009] oder sogar durch die Ähnlichkeiten dieser Relationen [Lémdani et al., 2011] ermittelt. Allerdings weisen nicht alle Eigenschaften eines Elementes die gleiche Relevanz bei der Bestimmung ähnlicher Elemente auf [Wang et al., 2009]. Dies berücksichtigen die Graph-basierten Verfahren aufgrund ihrer Arbeitsweise jedoch nicht, da hierbei eine Abstraktion von der Bedeutung der Eigenschaften vorgenommen wird. Dies stellt andererseits aber auch ihren größten Vorteil dar, da diese Verfahren unabhängig von der verwendeten Domäne auf einen beliebigen Datensatz anwendbar sind, während den Verfahren, welche die Bedeutung der Eigenschaften verwenden, diese zuerst im Vorfeld verständlich gemacht werden muss. Auch wenn dies aufgrund der Verwendung von Technologien und Methoden des *Semantic Webs* stark vereinfacht wird, ist damit trotz alledem ein gewisser Mehraufwand verbunden.

4.3 Zusammenfassung

Ähnlich den im *Semantic Web Technology Stack* angeordneten Technologien des *Semantic Webs* (siehe Kapitel 3.1), lassen sich auch die einzelnen Repräsentationen anhand der Mächtigkeit ihrer Aussagekraft hierarchisieren. Diese bauen ebenfalls aufeinander auf, erweitern das jeweils einfachere Modell und lassen sich somit anhand des Grades ihrer vorhandenen *Semantik* klassifizieren. Tabelle 4.1 zeigt die verschiedenen Repräsentationsformate der Benutzerprofile sowie Elemente und führt Beispiele der das entsprechende Format verwendenden Systeme aus der Li-

4.3. Zusammenfassung

teratur auf. Dabei entspricht die Anzahl der Sterne auf der Skala dem Grad der Aussagekraft der Repräsentation, und damit auch dem Grad ihrer *Semantik*.

Grad der <i>Semantik</i>	Repräsentation	Format	Beispielsystem
★	Bewertungsvektoren	Vektor von subjektiven Bewertungen auf Elemente	<ul style="list-style-type: none"> • GroupLens [Resnick et al., 1994] • MovieLens [Herlocker et al., 2000] • Item-To-Item Collaborative Filtering [Sarwar et al., 2001]
★★	Vector Space Modelle	Vektor von extrahierten <i>Schlüsselwörtern</i> mit Gewichtungen	<ul style="list-style-type: none"> • Syskill & Webert [Pazzani et al., 1996] • Amalthea [Moukas, 1997] • Fab [Balabanovic and Shoham, 1997] • Webmate [Chen and Sycara, 1998]
		Inhaltsbeschreibende Tags mit Gewichtungen	<ul style="list-style-type: none"> • Recommendation with Knowledge of Product Semantics [Ghani and Fano, 2002] • Movie Recommendation with Folksonomies [Szomszor et al., 2007] • Multimedia Sharing Systems [Musial et al., 2008]
★★★	Taxonomische Repräsentationen	Taxonomie von Konzepten mit Gewichtungen	<ul style="list-style-type: none"> • Persona [Tan et al., 1998] • OBIWAN [Gauch et al., 2003] • Taxonomic-driven Recommender [Ziegler et al., 2004b] • Quickstep und Foxtrott [Middleton et al., 2004] • News@Hand [Cantador and Castells, 2009]
★★★★	Semantische Repräsentationen	<i>semantisches</i> Netzwerk von Konzepten mit Gewichtungen	<ul style="list-style-type: none"> • ifWeb [Asnicar and Tasso, 1997] • SiteIF [Magnini and Strapparava, 2004] • WIFS [Micarelli and Sciarone, 2004]
		Netzwerk aus typisierten Daten und ihren Bedeutungen	<ul style="list-style-type: none"> • Entree [Burke, 2000] • Semantic Product Recommendation [Farsani and Nematbakhsh, 2006] • CHIP [Wang et al., 2008] • dbrec [Passant, 2010]

Tabelle 4.1 – Semantik in Benutzerprofilen und Elementrepräsentationen

Den geringsten Grad an *Semantik* weisen die Repräsentationen auf, welche auf der Verwendung von Bewertungsvektoren basieren. Diese stellen an sich keinerlei Beschreibungen von Benutzerinteressen oder Elementeigenschaften bereit, sondern abstrahieren vollständig von den

Attributen der Elemente und betrachten einen Benutzer beziehungsweise ein Element als Ganzes, ohne Berücksichtigung der Details.

Vector SpaceModell basierte Repräsentation verwenden aus den Beschreibungen der Elemente extrahierte Worte, um die Interessen eines Benutzers sowie die Elementeigenschaften zu beschreiben. Dabei werden die Eigenschaften der Elemente im Vorfeld mittels Textanalyseverfahren ermittelt und gespeichert. Diese Repräsentation ermöglichen den Benutzern zudem, ihre Vorlieben gezielt anzugeben. Jedoch weisen die Verfahren prinzipbedingt diejenigen Probleme auf, welche bei der Verarbeitung von natürlich sprachlichen Texten auftreten.

Dieses Problem weisen die *taxonomischen* Repräsentationen nicht auf, da die Verwendung des Konzeptmechanismus eine eindeutige Beschreibung und Interpretation der Eigenschaften eines Elementes realisiert. Weiterhin können durch die Taxonomien auch Ober- und Unterkonzepte der in den Elementen auftretenden Konzepte bestimmt werden, wodurch eine Erweiterung der Benutzerprofile realisiert wird und dadurch weitere, thematisch verwandte Elemente von Interesse ermittelt werden können.

Der größte Grad an *Semantik* lässt sich durch *semantische* Repräsentation erreichen. Diese berücksichtigen die Zusammenhänge zwischen den Konzepten, welche von den einfacheren Repräsentationen vernachlässigt werden. Eine verbesserte Art der *semantischen* Repräsentation lässt sich durch Verwendung von Technologien des *Semantic Webs* realisieren. Dabei liegen die Elemente sowie die Profile in einer strukturierten und formalen Beschreibung vor, wodurch sich die Elemente sehr exakt in ihrer Bedeutung beschreiben und verarbeiten lassen. Aufgrund der bekannten Bedeutung einer Eigenschaft und ihres festen Wertebereichs lassen sich Methoden für Vergleiche einsetzen, die auf die direkte Bedeutung einer einzelnen Eigenschaft abgestimmt sind. Weiterhin lässt sich durch die globale Eindeutigkeit der Elemente und ihrer Eigenschaften neben der verwendeten Domänenontologie auch weiteres Wissen aus externen Quellen im Verarbeitungsprozess verwenden. Neben diesen auf der Bedeutung abgestimmten Methoden existieren auch domänenunabhängige Verfahren, welche anhand der Struktur des aus den Relationen zwischen den Elementen gebildeten Netzwerkes, Vergleiche zwischen den Elementen ziehen und so die Vorhersagen für den Benutzer generieren.

Die einzelnen Repräsentationen benötigen jeweils einen bestimmten Grad an Aussagekraft, welcher von den verschiedenen Datenformaten und Konzepten des *Semantic Webs* bereitgestellt wird. Die dabei zumindest benötigte *Semantic Web*-Sprache sowie ihr technisches Konzept, mit welchem sie die von der Repräsentation geforderte Aussagekraft realisiert, ist in Tabelle 4.2 dargestellt. Zwar lässt sich auch mit der Verwendung von *RDF* eine *Taxonomie* realisieren, diese ist aber nicht formal definiert und nur durch die Anwendungslogik der verwendenden Applikation erfassbar. Erst durch den Einsatz von *RDFS* und durch die in dieser Sprache enthaltenen Sprachkonstrukte lassen sich entsprechend anwendungsunabhängige Taxonomien verwenden. Ähnliches gilt für die von den *semantischen* Repräsentationen benötigten Netzwerke. Hierbei werden die in *OWL* enthaltenen Sprachkonstrukte benötigt, um die Graphen formal zu definieren.

Der *semantischen* Klassifikation folgend werden im nächsten Kapitel ausgewählte Algorithmen für die verschiedenen Grade der *Semantik* vorgestellt. Sie generieren Vorschläge anhand

4.3. Zusammenfassung

Repräsentation	Benötigte Aussagekraft	Semantische Technologie	ermöglicht
Bewertungsvektoren	Unterscheidbarkeit von Elementen	<i>URI</i> (Kapitel 3.1.1)	Identifikationsmechanismus
Vector Space Modelle	Eigenschaften der Elemente	<i>XML</i> (Kapitel 3.1.2)	Vereinfachte Extraktion von Elementeigenschaften
Taxonomische Repräsentationen	Hierarchie von Elementeigenschaften	<i>RDFS</i> (Kapitel 3.1.4)	Taxonomie von Konzepten
Semantische Repräsentationen	Netzwerk von Elementeigenschaften	<i>OWL</i> (Kapitel 3.1.5)	Ontologie von Konzepten

Tabelle 4.2 – Äquivalenzen zwischen den Repräsentationen und dem Semantic Web

der in den Repräsentationen von Benutzern und Elementen des entsprechenden *semantischen* Grades enthaltenen Informationen. Zwar kommen oftmals in einfacheren Verfahren Repräsentationen einer höheren *Semantik* zum Einsatz, die dort enthaltenen Informationen dienen dabei jedoch nur für eine ansprechende Darstellung des Elementes im *Frontend* eines *Recommenders*. Zwingend benötigt werden sie hingegen für Verfahren einer höheren Ebene der *Semantik*, die in ihrem Betrieb Elementrepräsentationen und Benutzerprofile verlangen, die mindestens ihren Grad an *Semantik* beinhalten.

Kapitel 5

Algorithmen der Recommender

Die Kernkomponente eines *Recommender Systems* stellt neben den im vorherigen Kapitel vorgestellten Repräsentationsformaten die *Recommendation Engine* dar (siehe Abbildung 2.2). Diese generiert anhand der ihr vorliegenden Repräsentationen sowie einem gegebenenfalls vorhandenen Wissensmodell die Vorschläge für die dem Benutzer anzuzeigenden Elemente. In der Zielsetzung eines *Recommender Systems* muss zwischen Vorhersagen und Empfehlungen differenziert werden.

- **Vorhersagen** $pr(u, e)$ sind die errechneten Werte der möglichen Bewertung r eines Benutzers u auf ein einzelnes von ihm noch nicht bewertetes Element e .
- **Empfehlungen** $RS(u, N)$ sind Mengen von N Elementen, welche für den Benutzer u von wahrscheinlich höchstem Interesse sind. Diese Mengen enthalte keine Elemente, die der Benutzer bereits bewertet hat.

Aufgrund der divergenten Struktur der einzelnen Repräsentationen, welche dabei Informationen unterschiedlichen *semantischen* Grades beinhalten, müssen dabei verschiedene auf die entsprechende Repräsentation angepasste Algorithmen verwendet werden. Alle im Folgenden vorgestellten Verfahren zur Generierung der Vorhersagen basieren auf der Bestimmung einer Ähnlichkeit zwischen den Benutzerprofilen beziehungsweise den Elementrepräsentationen oder sogar beidem.

Neben den anhand von Ähnlichkeiten arbeitenden Verfahren existieren noch viele weitere Methoden zur Vorschlagsgenerierung. Diese basieren beispielsweise auf der Erzeugung eines lernenden Elementklassifikators, welcher mittels eines Neuronalen Netzes [Billsus and Pazzani, 1998] oder mit Bayseschen Netzwerken [Breese et al., 1998] den Korpus der Elemente in zu den Benutzerinteressen passende und nicht passende Elemente einteilt. Weiterhin existieren auch Verfahren, welche das aus dem Datamining stammende *Association Rule Mining* verwenden, um aus gemeinsam betrachteten Elemente mit den Support- und Confidence-Formeln [Agrawal et al., 1993] Regeln zu bestimmen, inwieweit diese zusammengehörig sind [Lin et al., 2002].

Die hier vorgestellten Ähnlichkeits-basierten Verfahren wurden aufgrund ihrer besseren Vergleichbarkeit untereinander gewählt. Weiterhin ermöglichen sie die Verwendung einer beliebigen, aus diskreten Werten bestehenden, Bewertungsskala, welche eine feinere Abstufung des Grades

5.1. Formale Definition der Recommender

an Interesse auf die Elemente bereitstellt, als dies mit der Verwendung von Klassifikatoren und Regeln nicht realisierbar wäre.

5.1 Formale Definition der Recommender

Um die Arbeitsweise eines *Recommender Systems* formal zu beschreiben wird eine Grundmenge von Begriffen benötigt, welche im Folgenden definiert werden:

Sei E die Menge der Mächtigkeit $|E| = n$ aller durch einen Index i unterscheidbarer Elemente $e_i \in E$ mit $0 \leq i < n$ und U die Menge der Mächtigkeit $|U| = m$ aller durch einen Index j unterscheidbarer Benutzer $u_j \in U$ mit $0 \leq j < m$. Dann ist die Bewertungsmatrix \mathfrak{R} als eine Matrix der Dimension $n \times m$ definiert und der Bewertungsvektor $\vec{u}_j = (r_{0,j} \cdots r_{i,j} \cdots r_{n-1,j})$ des Benutzers u_j mit Index j als Zeile der Matrix mit Index j gegeben. Weiterhin entsprechen die einzelnen Werte $r_{i,j} = r$ des Vektors \vec{u}_j den Bewertungen r des Benutzers u_j auf das Element e_i . Dabei liegt der Wert $r \in S$ innerhalb einer entsprechenden Bewertungsskala S . S ist eine Menge von der Domäne entsprechend angepasster Werte und kann beispielsweise boolesche (**true** und **false**), diskrete (zum Beispiel $\{1, 2, 3, 4, 5\}$) oder auch kontinuierliche Werte (zum Beispiel $[0..5]$) annehmen.

Weiterhin sei das Profil $Prof(u)$ eines Benutzers u definiert als das Resultat einer Abbildungsfunktion $f : u \mapsto f(u) := Prof(u)$, welche anhand der Repräsentationen $Rep(e)$ der vom Benutzer u bereits bewerteten Elemente $E(u) := \{e | e \in E \wedge r_{u,e} \neq \emptyset\}$ sein ihn beschreibendes Profil im System erzeugt. Dieses Profil besteht dabei aus einer Menge an Termen, welche die subjektiven Interessen des Benutzers beschreiben. Dazu verwendet die Funktion f die Repräsentation $Rep(e)$ der Elemente, welche das Resultat einer Abbildung $g : e \mapsto g(e) := Rep(e)$ ist, die das Element e in das Repräsentationsformat des Systems überführt. Die Elementrepräsentationen haben in den verschiedenen Verfahren jeweils eine auf deren Arbeitsweise angepasste Struktur und Bedeutung. So können diese aus Nummern, Tupeln, Textfragmente, *RDF*-Klassen oder sogar *RDF*-Teilgraphen bestehen. Auf die konkrete Realisierung der Repräsentationen von Benutzern und Elementen wird in Kapitel 5 bei den entsprechenden Verfahren detaillierter eingegangen, wo diese dann formal beschrieben werden.

Der aktuelle Benutzer, für den die Vorhersagen bestimmt werden soll, wird mit u angegeben und e ist das Element, für welches die Bewertung $Rating(u, e)$ des Benutzers u ermittelt wird. Im allgemeinen Fall lässt sich nun die Funktion $Rating : U \times E \rightarrow S$ wie folgt definieren:

$$Rating(u, e) = \begin{cases} r_{u,e} & \text{wenn } r_{u,e} \neq \emptyset \\ pr(u, e) & \text{sonst} \end{cases} \quad (5.1)$$

Falls der Benutzer u das Element e bereits bewertet hat, ist das Resultat der Funktion $Rating(u, e)$ seine bereits abgegebene Bewertung $r_{u,e}$, ansonsten das Ergebnis der Vorhersagefunktion $pr(u, e)$. Hierbei bestimmt diese anhand der bisher bekannten Bewertungen aus der Bewertungsmatrix \mathfrak{R} , der Menge aller Benutzer U , sowie der Elemente E die mögliche Bewertung $pr(u, e)$ für den Benutzer u für das Element e innerhalb von S . Diese Funktion ist stark vom eingesetzten Verfahren abhängig und stellt hier nur den allgemeinen und für jedes Verfahren

passenden Fall dar und unterscheidet sich in ihrer jeweiligen Realisierung. Die Hilfsfunktionen werden im folgenden Kapitel für die entsprechenden Verfahren detailliert beschrieben. Dabei verwenden sie jeweils die vom System vorgegebenen Repräsentationsformate $Prof(u)$ sowie $Rep(e)$.

Mit Hilfe von $pr(u, e)$ können auch die Empfehlungen $RS(u, N)$ für einen Benutzer u ermittelt werden. $RS(u, N)$ ist eine nach den errechneten Vorhersagen $pr(u, e_l)$ geordnete Menge der Mächtigkeit $|RS(u, N)| = N$ und beinhaltet genau die N Elemente e , für die das System die Vorhersagen $pr(u, e)$ mit den höchsten Werten errechnet und für die der Benutzer u noch keine Bewertungen abgegeben hat. Sie ist formal definiert durch:

$$\begin{aligned} RS(u, N) := \{ e_l \mid & e_l \in E \setminus E(u) \\ & \wedge 0 \leq l < N \\ & \wedge pr(u, e_l) \geq pr(u, e_{l+1}) \} \end{aligned} \quad (5.2)$$

Die Elemente $e_i \in RS(u, N)$ der Menge sind dabei aufsteigend anhand des Wertes der auf sie errechneten Vorhersagen $pr(u, e_i)$ sortiert.

5.2 Algorithmen

Für jedes der in Kapitel 4 vorgestellten Repräsentationsformate existieren unterschiedliche Verfahren, um aus den vorliegenden Daten sowie weiteren Informationen Vorhersagen und Empfehlungen auf Elemente von voraussichtlichem Interesse zu erzeugen. Diese Verfahren werden im folgenden Kapitel vorgestellt und formal definiert. Neben diesen Verfahren wird auch ein einfacher Algorithmus vorgestellt, welcher jedoch keinerlei *Personalisierung* ermöglicht.

5.2.1 Mean-Based Recommender

Das einfachste Verfahren basiert auf der Bestimmung der durchschnittlichen Bewertungen der Benutzer $u \in U$ auf die Elemente $e \in E$. Dabei stellt dieses Verfahren keinerlei *Personalisierung* bereit, da die Interessen eines einzelnen Benutzers nicht berücksichtigt und für alle Benutzer die selben Vorhersagen erzeugt werden. Somit wird kein den Benutzer u repräsentierendes Profil $Prof(u)$ generiert.

Im ersten Schritt wird die Menge der Benutzer $U(e)$ ermittelt, welche das Element e bewertet haben. Diese ist definiert durch $U(e) := \{u \mid u \in U \wedge r_{u,e} \neq \emptyset\}$. Die Vorhersage $pr(u, e)$ für das Element e ergibt sich dann aus dem arithmetischen Mittel aller von den Benutzern $i \in U(e)$ auf das Element abgegebenen Bewertungen $r_{i,e}$:

$$pr(u, e) = \frac{\sum_{i \in U(e)} r_{i,e}}{|U(e)|} \quad (5.3)$$

Diese Gleichung macht ersichtlich, dass die speziellen Interessen eines Benutzer keinerlei Einfluss auf das Ergebnis für ihn aufweisen, sondern jeder Benutzer identisch betrachtet wird. Somit benötigt diese Verfahren keinerlei Benutzerprofile. Weiterhin ist es problematisch, dass

5.2. Algorithmen

wenige, aber sehr gute beziehungsweise schlechte Bewertungen auf ein Element, das für dieses berechnete Ergebnis stark verzerren können, so dass es immer in der Empfehlungsmenge $RS(u)$ enthalten ist, egal welchen objektiven Nutzen das Element für alle Benutzer im Mittel aufweist. Trotzdem kann dieses Verfahren als Ausweidlösung im System verwendet werden, falls die sonstigen im Folgenden vorgestellten Verfahren aufgrund zu weniger Informationen über die Elemente oder Benutzer keinerlei Vorhersagen generieren können.

5.2.2 Collaborative Filtering

Collaborative Filtering stellt das Verfahren mit dem geringsten Grad an verwendeter *Semantik* dar und basiert auf der Benutzung der in Kapitel 4.2.1 vorgestellten Bewertungsvektoren. Hierbei werden die Vorschläge und Empfehlungen ausschließlich anhand der bisherigen Bewertungen der Benutzer erzeugt. Je nach eingesetztem Verfahren werden die zum aktuellen Benutzer ähnlichsten anderen Benutzer oder zur Bewertungshistorie passende Elemente ermittelt. Aufgrund der Einfachheit der Verfahren und der Unabhängigkeit vom Inhalt der Elemente besteht ein Benutzerprofil $Prof(u)$ eines Benutzers u daher aus der Menge an Tupeln $(e, r_{u,e})$ der von ihm bereits bewerteten Elemente $e \in E(u)$ sowie der auf sie von ihm abgegebenen Bewertungen $r_{u,e}$. Diese Menge ist definiert durch:

$$Prof(u) := \{(e, r_{u,e}) \mid e \in E(u)\} \quad (5.4)$$

Die Repräsentation der Elemente im System $Rep(e)$ weist eine ähnlich triviale Struktur auf. Sie besteht sogar nur aus dem Index i des Elementes e_i , welcher das Element im System eindeutig identifiziert.

$$Rep(e) := \{i \mid e \in E \wedge e = e_i\} \quad (5.5)$$

User to User Collaborative Filtering

Im *User to User Collaborative Filtering* werden die Vorhersagen $pr(u, e)$ für einen Benutzer u auf ein Element e anhand seiner Nachbarschaft von Benutzern erzeugt. Diese Nachbarschaft besteht dabei aus den ähnlichsten anderen Benutzern bezüglich des eingesetzten Ähnlichkeitsmaßes. Anhand der von den Benutzern der Nachbarschaft abgegebenen Bewertungen auf das Element e sowie deren Ähnlichkeiten kann die Vorhersage $pr(u, e)$ errechnet werden. Im Folgenden wird das von [Resnick et al., 1994] im GroupLens-System eingesetzte Verfahren des *User To User Collaborative Filtering* vorgestellt.

Um die Nachbarschaft des Benutzers u zu bilden, werden zuerst die Ähnlichkeiten zwischen dem Benutzer u und allen anderen Benutzern $n \in U$ ermittelt. Dazu werden für jeden weiteren Benutzer n die Elemente bestimmt, die sowohl von Benutzer n als auch von Benutzer u bereits bewertet worden sind. Diese Menge ist gegeben durch $E(u, n) := E(u) \cap E(n)$. Nachdem $E(u, n)$ ermittelt wurde, können dann u und n mit dem *Pearson*-Korrelationskoeffizienten [Pearson, 1895] anhand ihrer Bewertungen $r_{u,e}$ beziehungsweise $r_{n,e}$ auf die Elemente e der Menge $E(u, n)$ verglichen werden. Dabei dient der Mittelwert der Bewertungen \bar{r}_u respektive \bar{r}_n der

Benutzer u und n als Anpassungsfaktor für den Koeffizienten, wodurch das unterschiedliche Bewertungsverhalten der Benutzer berücksichtigt wird.

$$\text{corr}(u, n) = \frac{\sum_{e \in E(u, n)} (r_{u,e} - \bar{r}_u) \cdot (r_{n,e} - \bar{r}_n)}{\sqrt{\sum_{e \in E(u, n)} (r_{u,e} - \bar{r}_u)^2} \cdot \sqrt{\sum_{e \in E(u, n)} (r_{n,e} - \bar{r}_n)^2}} \quad (5.6)$$

Es ist jedoch zu beachten, dass die *Pearson*-Korrelation auch negative Werte annehmen kann, was dann einer perfekten Uneinigkeit zwischen den Benutzern u und n entspricht. Negative Werte der Korrelation $\text{corr}(u, n)$ verbessern jedoch nicht die Qualität der gemachten Vorhersagen [Schafer et al., 2007], so dass hier diese Ergebnisse als 0 betrachtet werden.

$$\text{userSim}(u, n) = \begin{cases} \text{corr}(u, n) & \text{wenn } \text{corr}(u, n) > 0 \\ 0 & \text{sonst} \end{cases} \quad (5.7)$$

Unter Verwendung der Ähnlichkeit $\text{userSim}(u, n)$ zwischen zwei Benutzern u und n kann nun die Nachbarschaft $N(u, k)$ des Benutzers u gebildet werden. Die Nachbarschaft des Benutzers u ist definiert als eine geordnete Menge der Mächtigkeit $|N(u, k)| = k$ und beinhaltet die, zum Benutzer u anhand des Ähnlichkeitsmaßes $\text{userSim}(u, n)$ geordneten, k ähnlichsten anderen Benutzer. Die Menge ist definiert durch:

$$\begin{aligned} N(u, k) := \{ u_i \mid & u_i \in U \\ & \wedge 0 \leq i < j < k \\ & \wedge \text{userSim}(u, i) > \text{userSim}(u, j) \} \end{aligned} \quad (5.8)$$

Während die ersten Verfahren noch alle Benutzer des Systems berücksichtigen, dass heißt $k = |U|$, konnte nachgewiesen werden, dass eine Reduzierung der Nachbarschaft auf die $k < |U|$ ähnlichsten Benutzer die Qualität der erzeugten Vorhersagen steigert. Der optimale Wert des Parameters k ist dabei jedoch abhängig vom verwendeten Datensatz und muss im Vorfeld ermittelt werden. Evaluierungen konnten dabei Werte von $k = 50$ bestimmen [Herlocker et al., 1999].

Anhand der Nachbarschaft $N(u, k)$, sowie den von den Benutzern $n \in N(u, k)$ innerhalb der Nachbarschaft abgegebenen Bewertungen $r_{n,e}$ auf das vorzuschlagende Element e , kann nun die Vorhersage $pr(u, e)$ für den Benutzer bestimmt werden. Weiterhin fließen auch der Mittelwert \bar{r}_u der Bewertungen des Benutzers sowie der Mittelwert \bar{r}_n der einzelnen Benutzer $n \in N(u, k)$ der Nachbarschaft in die Berechnung ein, um das unterschiedliche Bewertungsverhalten der Benutzer entsprechend abzubilden. Die Vorhersage $pr(u, e)$ errechnet sich dann anhand der

5.2. Algorithmen

gewichteten Summe der Ähnlichkeiten $userSim(u, n)$ sowie der Bewertungen $r_{n,e}$ der Benutzer der Nachbarschaft:

$$pr(u, e) = \bar{r}_u + \frac{\sum_{n \in N(u, k)} userSim(u, n) \cdot (r_{n,e} - \bar{r}_n)}{\sum_{n \in N(u, k)} userSim(u, n)} \quad (5.9)$$

Bei der Anwendung dieses Verfahrens ist zu beachten, dass für einen Benutzer, zu dem keine Nachbarn ermittelt werden können, keine Vorschläge generiert werden können. Dies kann durch das Fehlen von gemeinsam bewerteten Elementen und dadurch mangelnder Überlappung zwischen den Benutzerprofilen oder auch durch das Fehlen von Bewertung durch ein neues und leeres Benutzerprofil bedingt sein. In diesem Fall muss auf ein anderes Verfahren wie beispielsweise die Bestimmung des Mittelwerts der auf das Element abgegebenen Bewertungen ausgewichen werden. Weiterhin können keine Vorschläge für Elemente generiert werden, welche bisher von keinem Benutzer bewertet wurden.

Item to Item Collaborative Filtering

Item to Item Collaborative Filtering stellt quasi die Umkehrung des *User to User Collaborative Filterings* dar. Während bei letzterem die Vorschläge $pr(u, e)$ anhand der Nachbarschaft der ähnlichsten Benutzer generiert werden, geschieht dies beim *Item to Item Collaborative Filtering* anhand der Ähnlichkeiten der vorzuschlagenden Elemente zu den bereits bewerteten Elementen eines Benutzers. Dies basiert auf der Annahme, dass Elemente, welche von den Benutzern in Kombination ähnlich bewertet wurden, auch in ihrer Bedeutung Ähnlichkeiten aufweisen. Anhand dieser Ähnlichkeiten sowie der subjektiven Bewertungshistorie des aktuellen Benutzers werden dann Vorhersagen auf die potentielle Bewertung getroffen. Im Folgenden wird das von [Sarwar et al., 2001] beschriebene Verfahren des *Item to Item Collaborative Filtering* vorgestellt.

Um die Vorhersage auf ein von einem Benutzer u noch unbewertetes Element e zu erzeugen müssen mehrere Schritte durchgeführt werden. Ausgehend von der Menge der vom Benutzer u bereits bewerteten Elemente $E(u)$, werden für jedes Element $e' \in E(u)$, diejenigen Benutzer $U(e, e')$ ermittelt, welche sowohl das Element e' als auch das Element e bewertet hatten. Diese Menge ist gegeben durch:

$$U(e, e') := U(e) \cap U(e') \quad (5.10)$$

Anhand der von ihnen abgegebenen Bewertungen wird dann die Ähnlichkeit zwischen den einzelnen Elementen berechnet, wobei eine dem *Pearson*-Korrelationskoeffizienten (siehe Gleichung 5.6) ähnelnde Gleichung verwendet wird, welche als *Adjusted Cosinus Measure* bezeichnet wird. Im Gegensatz zum *Pearson*-Korrelationskoeffizient ist der Anpassungsfaktor dabei jedoch nicht der Mittelwert aller Bewertungen auf die jeweiligen Elemente e respektive e' , sondern entspricht dem Mittelwert \bar{r}_u aller Bewertungen des Benutzers u . Dies birgt den Vorteil, dass dadurch auch die unterschiedlichen Bewertungsverhalten der Benutzer in Betracht gezogen und

damit ihre Bewertungen besser untereinander in ihrer subjektiven Bedeutung verglichen werden können. Die Ähnlichkeit ist dabei gegeben durch:

$$\text{corr}(e, e') = \frac{\sum_{u \in U(e, e')} (r_{u, e} - \bar{r}_u) \cdot (r_{u, e'} - \bar{r}_u)}{\sqrt{\sum_{u \in U(e, e')} (r_{u, e} - \bar{r}_u)^2} \cdot \sqrt{\sum_{u \in U(e, e')} (r_{u, e'} - \bar{r}_u)^2}} \quad (5.11)$$

Da die Korrelation $\text{corr}(e, e')$ auch negative Werte annehmen kann, die zwar einen negativen Bezug zwischen den beiden Elementen e und e' angeben, jedoch nicht zu einer zuverlässigen Bestimmung der Ähnlichkeit verwendet werden können [Schafer et al., 2007], muss dieser Wert entsprechend angepasst werden. Somit ergibt sich für die Ähnlichkeit $\text{itemSim}(e, e')$ folgender Wert:

$$\text{itemSim}(e, e') = \begin{cases} \text{corr}(e, e') & \text{wenn } \text{corr}(e, e') > 0 \\ 0 & \text{sonst} \end{cases} \quad (5.12)$$

Sind die Ähnlichkeiten des Elementes e zu allen vom Benutzer u in seinem Profil $\text{Prof}(u)$ enthaltenen und damit bereits bewerteten Elementen $e' \in E(u)$ ermittelt kann damit die Vorhersage $pr(u, e)$ des Benutzers u auf das Element e errechnet werden. Diese ergibt sich aus der gewichteten Summe der Ähnlichkeiten $\text{itemSim}(e, e')$ sowie der im Profil des Benutzers u enthaltenen Elemente und ihren Bewertungen $(e', r_{e'})$:

$$pr(u, e) = \frac{\sum_{(e', r_{e'}) \in \text{Prof}(u)} \text{itemSim}(e, e') \cdot r_{e'}}{\sum_{(e', r_{e'}) \in \text{Prof}(u)} \text{itemSim}(e, e')} \quad (5.13)$$

Weiterhin kann bei der Verarbeitung der Menge $E(u)$ die Bedingung gestellt werden, dass nur die k ähnlichsten Elemente zum Element e betrachtet werden oder nur die Elemente, deren Ähnlichkeiten über einem gewissen Grenzwert $\text{itemSim}(e, e') \geq t$ liegen. Beide Kriterien können auch kombiniert werden, so dass nur die Elemente verwendet werden, deren Ähnlichkeit über dem Grenzwert liegt, höchstens aber k viele. Der Grenzwert t muss jedoch im Vorfeld aufwändig bestimmt werden und ist in starkem Maße abhängig von den verwendeten Daten [Herlocker et al., 1999], so dass die meisten Verfahren nur das erste Kriterium aufgrund seiner relativen Unabhängigkeit von den Eingabedaten verwenden. Im Allgemeinen erhöht diese Reduktion der Vergleichsmenge die Skalierbarkeit des Verfahrens aufgrund der oftmals viel geringeren Anzahl der durchgeführten Operationen. In der von [Sarwar et al., 2001] durchgeführten Evaluierung des Verfahrens konnte eine Reduzierung der Menge auf die 40 ähnlichsten Elemente sogar zu einer Verbesserung der Genauigkeit der generierten Vorhersagen bezüglich der verwendeten Metrik führen.

Auch das *Item to Item Collaborative Filtering* weist die bereits beschriebene Einschränkung auf, dass der Benutzer u mindestens ein Element bewertet haben und das vorzuschlagende Element e bereits von mindestens einem anderen Benutzer u' bewertet worden sein muss .

5.2. Algorithmen

Zusätzlich muss dieser Benutzer u' neben dem Element e auch mindestens ein weiteres Element $e' \in E(u)$ von u bewertet haben, da sonst keine Ähnlichkeit zwischen den Elementen e und e' bestimmt werden kann.

5.2.3 Content-Based Recommender

Im Gegensatz zu dem auf der Verwendung von Bewertungsvektoren basierenden *Collaborative Filtering* werden bei den *Content-Based Recommendern* die Vorhersagen anhand der Eigenschaften der Elemente sowie eines die Eigenschaften der bevorzugten Elemente beinhaltenden Benutzerprofil gebildet. Benutzerprofile bestehen hierbei aus einem Vektor von aus den Elementen extrahierten Termen sowie deren Gewichtungen, welche den Grad des Interesses des Benutzers auf diesen Term angeben (siehe Kapitel 4.2.2). In einer Elementrepräsentation sind die Terme nach ihrer Signifikanz für die Bedeutung des Elementes anhand der *TF-IDF*-Metrik [Salton, 1989] gewichtet. Diese ist definiert durch:

$$TF-IDF = tf_{k,e} \cdot \log \frac{N}{N_k} \quad (5.14)$$

Dabei stellt $tf_{k,e}$ die Anzahl des Auftretens des Terms k im Element e dar, N die Anzahl aller Elemente des Systems und N_k die Anzahl der Elemente, welche den Term k enthalten. Ausgehend von dieser Gewichtung werden die als Vektoren aufgefassten Elemente \vec{e} anhand des Skalarproduktes miteinander verglichen. Aus diesem ergibt sich der Cosinus des Winkels zwischen den beiden Vektoren, der als Ähnlichkeitsmaß zwischen einem Benutzerprofil und einem Element dient:

$$\cos Sim(\vec{e}, \vec{e}') = \frac{\vec{e} \cdot \vec{e}'}{|\vec{e}| \cdot |\vec{e}'|} \quad (5.15)$$

Im folgend beschriebenen einfachen Verfahren werden Vorhersagen $pr(u, e)$ auf ein Element e berechnet, indem dieses mit jedem einzelnen der vom Benutzer e bereits bewerteten Elemente $e' \in E(u)$ verglichen wird. Die Vorhersage $pr(u, e)$ ergibt sich dann aus der gewichteten Summe der Ähnlichkeiten $\cos Sim(\vec{e}, \vec{e}')$ zwischen den Elementen e und e' sowie ihrer Bewertungen $r_{u,e'}$:

$$pr(u, e) = \frac{\sum_{e' \in E(u)} \cos Sim(\vec{e}, \vec{e}') \cdot r_{u,e'}}{\sum_{e' \in E(u)} \cos Sim(\vec{e}, \vec{e}')} \quad (5.16)$$

Tag-Based Recommender

Content-Based Recommender lassen sich nicht ohne weiteres auf Elemente anwenden, welche keine textuelle Beschreibung anbieten, wie beispielsweise Musik oder Filme. Hierbei muss auf Techniken und Datenstrukturen zurückgegriffen werden, die eine textuelle Beschreibung eines Elementes ermöglichen. Eine solche Technik stellt die Verwendung von den Inhalt beschreibenden *Schlüsselworten*, den sogenannten *Tags* dar. Eine Sammlung solcher *Tags* wird dabei als eine *Tag-Cloud* bezeichnet, die in dem hier vorgestellten Verfahren den Termvektoren des *Vector Space Modells* entsprechen. Im Folgenden wird ein solches von [Szomszor et al., 2007] vorgeschlagenes

Verfahren, welches auf der Verwendung von *Tag-Clouds* zur Beschreibung von Filmen basiert, vorgestellt. Dieses Verfahren lässt sich auch auf beliebige andere Dateiformate anwenden, sofern diese eine Beschreibung durch *Tags* ermöglichen.

Vor ihrer Verwendung im System werden die Repräsentationen $Rep(e)$ aller Elemente $e \in E$ zum späteren Zugriff generiert und gespeichert. Dazu wird für jedes einzelne Element e die Menge K_e seiner in ihm auftretenden Schlüsselworte k extrahiert. Etwaige auftretende *Stopworte*⁴⁵, welche keine Relevanz bei der Beschreibung des Inhaltes aufweisen, werden dabei nicht in die Menge mit aufgenommen. Die Menge K_e aller verarbeiteter Worte k eines Elementes stellt dann die Repräsentation $Rep(e) \equiv K_e$ des Elementes e im System dar.

Anhand der Elementrepräsentationen $Rep(e)$ der vom Benutzer bewerteten Elemente $e \in U(u)$ wird dann das Benutzerprofil gebildet. Dazu wird für jeden Wert der Bewertungsskala $r \in S$ eine entsprechende *Tag-Cloud* $T(u, r)$ in diesem Profil erzeugt. Somit ist das Profil $Prof(u)$ eines Benutzers u definiert als eine Menge von *Tag-Clouds* für alle möglichen Bewertungswerte r der Skala S :

$$Prof(u) := \{T(u, r) \mid \forall r \in S \exists T(u, r) \in Prof(u)\} \quad (5.17)$$

Eine *Tag-Cloud* $T(u, r)$ eines Benutzers u für den Bewertungswert r besteht ferner aus allen Tupeln (k, n_k) gegeben durch die Schlüsselworte $k \in Rep(e)$ der Elemente $E(u, r) := \{e \mid e \in E(u) \wedge r_{u,e} = r\}$, welche der Benutzer mit dem Wert r bewertet hat, sowie der Anzahl $n_k(u, r)$ der Elemente mit der Bewertung r , die das Schlüsselwort k beinhalten. Die *Tag-Cloud* $T(u, r)$ ist definiert durch:

$$T(u, r) := \{(k, n_k(u, r)) \mid e \in E(u, r) \wedge k \in Rep(e)\} \quad (5.18)$$

$$\text{mit } n_k(u, r) = |\{e \mid e \in E(u, r) \wedge k \in Rep(e)\}| \quad (5.19)$$

Aufgrund der Aufteilung des Profils in einzelne diskrete *Tag-Clouds* weist dieses Verfahren allerdings die Einschränkung auf, dass es nur in Systemen einsetzbar ist, welche eine Bewertungsskala bestehend aus diskreten Werten verwenden, wobei die Anzahl der Werte der Skala jedoch beliebig ist. Das System von [Szomszor et al., 2007] benutzt beispielsweise eine Bewertungsskala mit den Werten $S := \{1, 2, 3, 4, 5\}$, was sich aus dem dort verwendeten Datensatz ergibt.

Um für ein neues, das heißt noch unbewertetes Element $e \notin E(u)$ die Vorhersagen für den Benutzer u zu ermitteln, werden im ersten Schritt die Ähnlichkeiten des Elementes e zu den einzelnen *Tag-Clouds* seines Profils bestimmt. Dies geschieht, indem die Signifikanz aller Worte k , die sowohl in der *Tag-Cloud* $T(u, r)$ als auch im Element e vorkommen, aufsummiert wird. Dabei wird ein Schlüsselwort k proportional zu seinem Auftreten in der *Tag-Cloud* gewichtet. Neben dieser lokalen Gewichtung wird auch sein globales Auftreten N_k in allen Elementen, ähnlich der *TF-IDF*-Metrik, berücksichtigt und fließt invers proportional in die Signifikanz ein. Somit wird beachtet, dass global seltener auftretende Worte eine höhere Signifikanz für die Bedeutung

⁴⁵sehr allgemeine „Bindewörter“ wie beispielsweise „und“, „oder“, „der“

5.2. Algorithmen

eines Elementes aufweisen, während häufig auftretende Worte kaum zur Bedeutung des Inhaltes beitragen. Der errechnete Wert $tagSim(T(u, r), e)$ der Summe ergibt dann die Ähnlichkeit des Elementes e zu der *Tag-Cloud* $T(u, r)$ des Benutzers u für den Bewertungswert r :

$$tagSim(T(u, r), e) = \sum_{\{(k, n_k) \in T(u, r) \wedge k \in Rep(e)\}} \frac{n_k(u, r)}{\log(N_k)} \quad (5.20)$$

Da hierbei nur Worte betrachtet werden, die sowohl in der *Tag-Cloud* als auch im Element selber auftreten, ist sichergestellt, dass deren globale Anzahl N_k niemals 0 entspricht. Weiterhin kann auch $\log(N_k)$ niemals 0 sein. Ein global nur einmalig vorhandenes Schlüsselwort wird nicht berücksichtigt, da es nur im vorzuschlagenden Element und nicht in der *Tag-Cloud* vorkommt oder bereits Teil der *Tag-Cloud* ist und somit nicht im Element enthalten sein kann. Der einzige Fall, in welchem ein einmalig existierendes *Schlüsselwort* sowohl im Element als auch in der *Tag-Cloud* vorkommt, entsteht, wenn eine Vorhersage auf ein vom Benutzer bereits bewertetes Element generiert werden soll. Gleichung 5.1 schließt diesen Fall aus, da dann das Ergebnis der bereits durchgeführten Bewertung entspricht und Gleichung 5.20 nicht verwendet wird.

Sind die Ähnlichkeiten zwischen dem Element e und den *Tag-Clouds* des Benutzers u ermittelt, lässt sich die Vorhersage $pr(u, e)$ auf seine Bewertung errechnen. Diese entspricht dann der gewichteten Summe der Ähnlichkeiten $tagSim(T(u, r), e)$ des Elementes e zu der jeweiligen *Tag-Cloud* $T(u, r)$ und dem entsprechenden Bewertungswert r :

$$pr(u, e) = \frac{\sum_{T(u, r) \in Prof(u)} tagSim(T(u, r), e) \cdot r}{\sum_{T(u, r) \in Prof(u)} tagSim(T(u, r), e)} \quad (5.21)$$

Ein Verbesserung des Verfahrens kann dadurch erzielt werden, dass nicht nur die einfachen extrahierten *Schlüsselworte* der Elemente in ihren jeweiligen Zeichenketten auf ihre Identität überprüft werden, sondern stattdessen die durch sie dargestellten Konzepte für die Vergleiche verwendet werden. Somit können neben den vorhandenen Worten auch ihre Synonyme berücksichtigt und gegebenenfalls vorhandene Mehrdeutigkeiten durch auftretende Homonyme aufgelöst werden. Diese Eindeutigkeit der Konzepte und der daraus abgeleiteten Beschreibungen der Elemente kann die Genauigkeit und Qualität der ermittelten Ähnlichkeiten verbessern (siehe Kapitel 4.2.3).

5.2.4 Taxonomic Recommender

Taxonomic Recommender erweitern die Verfahren der *Content-Based Recommender*, indem sie zusätzliches Wissen über die Elemente sowie den eigentlichen Prozess der Vorschlagsgenerierung in Form einer Taxonomie einsetzen. Diese besteht dabei aus in ihrer Bedeutung eindeutigen Konzepten und hierarchisiert sie anhand von Ober- und Unterkategorien. Somit können im Gegensatz zur Verwendung einfacher *Schlüsselworte* die Elementeeigenschaften sowie Benutzerinteressen eindeutig interpretiert werden (siehe Kapitel 4.2.3). Weiterhin können durch die in der

Taxonomie enthaltenen Informationen die Elemente untereinander in eine thematische Relation gesetzt werden.

Eine solche Taxonomie \mathcal{T} ist als ein zyklenfreier, gerichteter Graph $\mathcal{G} = (V, E)$ bestehend aus einer Menge an Knoten V und einer Menge an Kanten E definiert. Dabei gilt $E \subseteq V \times V$. Weiterhin weist der Graph \mathcal{G} genau einen Knoten c_{\perp} mit Eingangsgrad 0, die sogenannte Wurzel des Baumes, auf und es gilt ferner die Bedingung $|V| = |E| - 1$. In einer Konzepttaxonomie \mathcal{C} werden die in ihr enthaltenen Konzepte $c \in \mathcal{C}$ durch Knoten des Graphen dargestellt, das heißt $V \equiv \mathcal{C}$, und die gerichteten Kanten beschreiben die Hyperonym-Relation zwischen zwei Konzepten c und c' . Somit bedeutet eine in der Konzepttaxonomie \mathcal{C} vorhandene Kante $v = (c, c')$, dass c der Oberbegriff von c' ist.

Liegen jedoch die Elementbeschreibungen nicht bereits in Form von Konzepten der Taxonomie vor, müssen diese zuerst in einem *Preprocessing* genannten Schritt analysiert und in das verwendete Repräsentationsformat des Verfahrens überführt werden. Liegen die Elemente beispielsweise in Form einfacher Texten vor, so werden die enthaltenen Worte auf ihre entsprechenden *Synsets* abgebildet, welche den Konzepten der Taxonomie entsprechen. Dabei kommen Distanzfunktionen zum Einsatz, welche die Ähnlichkeit zwischen zwei Konzepten in der Taxonomie bestimmen. Neben den Metriken, welche lediglich die Anzahl der Kanten zwischen zwei Konzepten zählen [Leacock and Chodorow, 1998], kann bei der Bestimmung der Distanz auch zusätzlich ihre Position in der Taxonomie berücksichtigt werden [Oldakowski and Bizer, 2005].

Diese Distanzfunktionen können, neben der Ermittlung der zu den *Schlüsselworten* gehörenden *Synsets*, auch verwendet werden, um Elemente miteinander in einem verbesserten Maße anhand der sie beschreibenden Konzepte zu vergleichen. Durch die Verbindungen zwischen den einzelnen Konzepten und ihrer Hierarchisierung können thematisch ähnliche Elemente ermittelt werden, auch wenn diese keine gemeinsamen Konzepte beinhalten.

Im Folgenden wird exemplarisch ein Verfahren vorgestellt, welches anhand einer vorgegebenen Taxonomie zuerst die aus *Schlüsselworten* bestehenden Elemente in eine Konzept-basierte Repräsentation überführt und dann durch die Verwendung derselben Taxonomie die Vorschläge für den Benutzer generiert.

Preprocessing der Elemente

Im ersten Schritt werden die einfachen *Schlüsselworte* $k \in K_e$ der Elemente auf ihr entsprechendes und eindeutiges Konzept c abgebildet. Dies geschieht mit der Funktion $m' : K \rightarrow \mathcal{P}(\mathcal{C}), \mapsto m'(k_e) = C(k_e)$, welche ein *Schlüsselwort* $k_e \in K_e \subseteq K$ eines Elementes e auf die zu ihm gehörende Teilmenge der Konzepte $C(k) \in \mathcal{P}(\mathcal{C})$ abbildet. Dabei ist $\mathcal{P}(\mathcal{C})$ als Potenzmenge der Menge der Konzepte \mathcal{C} aus der Konzepttaxonomie \mathcal{C} definiert. Wenn es genau ein einziges passendes Konzept $c \in C(k_e)$ für das Schlüsselwort gibt, das heißt $|m'(k_e)| = 1$, so ist das eindeutige, das *Schlüsselwort* k_e repräsentierende Konzept gefunden worden. Besteht $C(k_e)$ jedoch aus mehreren möglichen Konzepten, so muss aus dem Kontext $C(e)$, in dem das *Schlüsselwort* im Element e steht, auf sein repräsentierendes Konzept c geschlossen werden. Dabei besteht der

5.2. Algorithmen

Kontext $C(e)$ aus allen bereits eindeutig den *Schlüsselworten* K_e des Elementes e zugewiesenen Konzepten des Elementes e :

$$C(e) := \{c \mid k \in K_e \wedge c \in m'(k) \wedge |m'(k)| = 1\} \quad (5.22)$$

Um einem mehrdeutigen *Schlüsselwort* k' sein eindeutiges Konzept c zuweisen zu können, wird die Bedeutung jedes seiner möglichen Konzepte $c' \in m(k')$ mit den Bedeutungen der bereits bekannten Konzepte $C(e)$ des Elementes e anhand des Ähnlichkeitsmaßes $conceptSim(c, c')$ verglichen. Als Formel für $conceptSim(c, c')$ kann beispielsweise das von [Leacock and Chodorow, 1998] vorgeschlagene Verfahren verwendet werden, welches die Ähnlichkeiten zwischen zwei Konzepten anhand ihrer Positionen in der Konzepttaxonomie \mathcal{C} ermittelt. Die Summe der Ähnlichkeiten $conceptSim(c, c')$ mit den Konzepten c' des Kontextes $C(e)$ ergibt dann die Ähnlichkeit der Bedeutung $senseSim(c, C(e))$ des Konzeptes c .

$$senseSim(c, C(e)) = \sum_{c' \in C(e)} conceptSim(c, c') \quad (5.23)$$

$$conceptSim(c, c') = \log \frac{2 \cdot D}{Dist(c, c')} \quad (5.24)$$

D entspricht dabei der im Vorfeld zu ermittelnden Höhe der Taxonomie und $Dist(c, c')$ einer im Folgenden beschriebenen Distanzfunktion. Um diese Funktion formal zu beschreiben, muss zuerst der Pfad $p(c)$ eines Konzeptes c zur Wurzel c_\perp der Konzepttaxonomie \mathcal{C} definiert werden. Dieser ist definiert als ein n -Tupel mit:

$$\begin{aligned} p(c) &:= (p_0, \dots, p_i, \dots, p_q) & (5.25) \\ \text{mit} & (p_i, p_{i+1}) \in V \\ & \wedge p_0 \equiv c_\perp \wedge p_q = c \end{aligned}$$

Dabei entspricht c_\perp der Wurzel von \mathcal{C} und p_q dem Konzept c , dessen Pfad durch $p(c)$ gegeben ist. Unter Verwendung des Pfades $p(c)$ kann nun der Abstand $Dist(c, c')$ zweier Konzepte c und c' in der Konzepttaxonomie \mathcal{C} definiert werden. Gegeben seien die Pfade $p(c)$ sowie $p(c')$ der Konzepte c und c' . Dann ist der kleinste gemeinsame Vorfahre $lca(c, c')$ das in den beiden Pfaden enthaltene Konzept p_a mit dem Index a , ab dem sich die Pfade zu unterscheiden beginnen. Der Abstand $Dist(c, c')$ zweier Konzepte c und c' ergibt sich dann aus der Summe der jeweiligen

Anzahl der traversierten Kanten zwischen c respektive c' und ihrem kleinsten gemeinsamen Vorfahren beider Konzepte $lca(c, c')$:

$$Dist(c, c') = q + p - 2 * a \quad (5.26)$$

$$lca(c, c') = p_a \quad (5.27)$$

$$\begin{aligned} \text{mit} \quad p(c) &= (p_0, \dots, p_a, \dots, p_p) \wedge p(c') = (p'_0, \dots, p'_a, \dots, p'_q) \\ &\wedge \forall 0 \leq i \leq a : p_i = p'_i \\ &\wedge \forall j > a : p_j \neq p'_j \end{aligned}$$

Wurden nun alle Ähnlichkeiten zwischen den möglichen Bedeutungen $c' \in m'(k)$ des *Schlüsselwortes* k und den bereits bekannten Konzepten des Elementes $C(e)$ errechnet, kann auf die Bedeutung $m(k, e) = c$ des *Schlüsselwortes* k im Element e geschlossen werden. Diese entspricht dann dem Konzept $c \in m(k, e)$ für das die höchste Ähnlichkeit zum Kontext $C(e)$ ermittelt wurde. Somit lässt sich die Abbildungsfunktion $m : K \times E \rightarrow C$ wie folgt beschreiben:

$$m(k, e) = \begin{cases} m'(k) & \text{wenn } |m'(k)| = 1 \\ c & \text{sonst} \end{cases} \quad (5.28)$$

$$\begin{aligned} \text{mit} \quad c &\in m'(k) \\ &\wedge \text{senseSim}(c, C(e)) = \max(\{\text{senseSim}(c', C(e)) \mid c' \in m'(k)\}) \end{aligned}$$

Die Funktion wird auf alle *Schlüsselworte* $k \in K_e$ des Elementes angewendet, so dass allen ihr eindeutiges Konzept c zugewiesen werden kann.

Generierung der Vorhersagen

Der im Folgenden beschriebene Algorithmus zur Berechnung der Vorhersagen $pr(u, e)$ für einen Benutzer u auf ein Element e basiert auf einem von [Ziegler et al., 2004b] vorgeschlagenen Verfahren. Dabei folgt das Verfahren dem *Collaboration via Content*-Ansatz von [Pazzani, 1999], bei dem die Nachbarschaft eines Benutzers nicht anhand der von den übrigen Benutzern abgegebenen Bewertungen auf die Elemente (siehe Kapitel 5.2.2) gebildet wird, sondern mittels einer Analyse der Eigenschaften der von ihnen bewerteten Elemente. So werden diejenigen Benutzer bestimmt, welche in ihren Eigenschaften ähnliche Elemente wie der aktuelle Benutzer bevorzugen.

In diesem Verfahren werden die Elemente ähnlich den *Content-Based Recommendern* in Form eines *Vector Space Modells* (siehe Kapitel 4.2.2) beschrieben. Im Gegensatz zu diesen basieren die erweiterten Repräsentationen auf der Verwendung von Konzepten, welche im Vorfeld durch das *Preprocessing* ermittelt werden. Dort werden die *Schlüsselworte* $k \in K_e$ des Elementes e auf ihre entsprechenden Konzepte abgebildet und bilden so die Menge der „realen“ Konzepte $C(e)$ eines Elementes e . Neben dieser Menge wird noch eine weitere Menge von Konzepten benötigt, um die Repräsentation $Rep(e)$ eines Elementes zu generieren, die sogenannten „virtuellen“ Konzepte $V(e)$. Diese kommen nicht im Element vor, sondern werden aus den in der Repräsentation enthaltenen Konzepten anhand der Konzepttaxonomie \mathcal{C} abgeleitet. Dabei entspricht die Menge $V(e)$ der

5.2. Algorithmen

Vereinigung aller in den Pfaden $p(c)$ der „realen“ Konzepte $c \in C(e)$ auftretenden Konzepte c' ohne die bereits im Element enthaltenen Konzepte $c \in C(e)$. Hierbei werden die Pfade als ungeordnete Mengen von Konzepten betrachtet.

$$V(e) := \bigcup_{c \in C(e)} p(c) \setminus C(e) \quad (5.29)$$

Im Gegensatz zu den „realen“ Konzepten $C(e)$ entspricht die Gewichtung $sco(c')$ eines „virtuellen“ Konzeptes c' nicht dem Wert 1, was seine Existenz im Element wiedergeben würde, sondern einem errechneten Wert $sco(c')$. Dieser Wert berücksichtigt die Position des Konzeptes c' in der Taxonomie \mathcal{T} . So weisen in der Taxonomie weiter „oben“ liegende Konzepte aufgrund ihrer Generizität ein kleineres Gewicht auf, als die weiter unten liegenden Konzepte, welche ein Element e spezifischer beschreiben können. Dabei „vererben“ die Konzepte ihr Gewicht entlang des Pfades $p(c)$ nach oben. Die Funktion $sib(c)$ bestimmt die Anzahl der Kinder des Konzeptes c in der Taxonomie \mathcal{C} .

$$sco(p_m) = \kappa \cdot \frac{sco(p_{m+1})}{sib(p_{m+1}) + 1} \quad (5.30)$$

Ferner gilt, dass die Summe der vererbten Gewichte entlang des Pfades $p(c)$ dem Gewicht $sc(c)$ des ursprünglichen Konzeptes c entspricht. Diese Summe muss dann entsprechend normiert werden.

$$\sum_{m=0}^q sco(p_m) = sc(c) \quad (5.31)$$

Ist ein Konzept sowohl „real“ als auch „virtuell“, muss dies entsprechend berücksichtigt werden, so dass sich sein Gewicht $w(c)$ aus seinem „realen“ Gewicht $w(c) = 1$ sowie aus dem „virtuellen“ Gewicht $sco(c)$ errechnet. Gleiches gilt auch, wenn das Konzept Oberbegriff von mehreren anderen Konzepten ist. So errechnet sich das Gewicht $sc(c)$ eines Konzeptes c wie folgt:

$$sc(c) = w(c) + \sum_{c \in p(c') \wedge c' \in C(e)} sco(c) \quad (5.32)$$

mit $w(c) = \begin{cases} 1 & \text{wenn } c \in C(e) \\ 0 & \text{sonst} \end{cases}$

Mit den eindeutig bestimmten Konzepten eines Elementes e lässt sich nun die Repräsentation $Rep(e)$ eines Elementes generieren. Diese besteht aus Tupeln $(c, w(c))$ von Konzepten c aus der Menge der „realen“ sowie der „virtuellen“ Konzepte und ihren errechneten Gewichtungen $sc(c)$:

$$Rep(e) := \{(c, sc(c)) \mid c \in C(e) \cup V(e)\} \quad (5.33)$$

Anhand der Elementrepräsentationen $Rep(e)$ der vom Benutzer u bereits bewerteten Elemente $e \in E(u)$ kann dann sein ihn beschreibendes Profil $Prof(u)$ gebildet werden. Analog zu

dem in Kapitel 5.2.3 vorgestellten Verfahren besteht auch das hier verwendete Benutzerprofil aus mehreren Konzeptvektoren $Cv(u, r)$, jeweils einem für jeden Wert $r \in S$ der Bewertungsskala des Systems. Diese Einteilung ermöglicht eine genaue Repräsentation der Vorlieben für jeden Grad des Interesses. Somit können die Abneigungen entsprechend getrennt von den Vorlieben repräsentiert werden.

$$Prof(u) := \{Cv(u, r) \mid \forall r \in S \exists Cv(u, r) \in Prof(u)\} \quad (5.34)$$

Der Konzeptvektor $Cv(u, r)$ besteht wiederum selbst aus Tupeln von Konzepten $c \in Rep(e)$, welche in den Repräsentationen der vom Benutzer mit dem Wert r bereits bewerteten Elemente $E(u, r)$ vorkommen, sowie deren jeweiligem kombiniertem Gewichtung $w(c, r)$. Diese Gewichtung entspricht dabei der Summe der in den Repräsentationen enthaltenen Gewichte $sc(c)$ des Konzeptes c .

$$\begin{aligned} Cv(u, r) &:= \{(c, w(c, u, r)) \mid c \in Rep(e) \wedge e \in E(u, r)\} \\ \text{mit} \quad w(c, u, r) &= \sum_{e \in E(u, r)} sc'(c, e) \\ sc'(c, e) &= \begin{cases} sc(c) & \text{wenn } (c, sc(c)) \in Rep(e) \\ 0 & \text{sonst} \end{cases} \end{aligned} \quad (5.35)$$

Anhand der durch die Konzeptvektoren beschriebenen Profile kann nun die Ähnlichkeit $userConceptSim(u, n)$ zwischen zwei Benutzern n und u ermittelt werden. Da sich das originale Verfahren nur für binäre Bewertungsskalen eignet, wurde es entsprechend erweitert, so dass diskrete Bewertungsskalen S mit beliebig vielen Abstufungen unterstützt werden. Dazu werden für jeden Wert $r \in S$ der Bewertungsskala die entsprechenden Konzeptvektoren $Cv(u, r)$ respektive $Cv(n, r)$ beider Benutzer miteinander durch den *Pearson*-Korrelationskoeffizienten anhand der Gewichte $w(c, u, r)$ der in ihnen enthaltenen Konzepte c miteinander verglichen. Dabei entspricht $\overline{w(n, r)}$ dem Mittelwert aller Konzeptgewichte im Konzeptvektor $Cv(u, r)$. Dabei entspricht $c \in Cv(u, r)$ der Existenz des Tupels $(c, w(c, u, r))$ im Konzeptvektor $Cv(u, r)$.

$$corr_r(u, n) = \frac{\sum_{c \in (Cv(u, r) \cap Cv(n, r))} (w(c, u, r) - \overline{w(u, r)}) \cdot (w(c, n, r) - \overline{w(n, r)})}{\sqrt{\sum_{c \in Cv(u, r)} (w(c, u, r) - \overline{w(u, r)})^2} \cdot \sqrt{\sum_{c \in Cv(n, r)} (w(c, n, r) - \overline{w(n, r)})^2}} \quad (5.36)$$

Da die Korrelation $corr(u, n)$ auch negative Werte annehmen kann, muss das Ergebnis entsprechend angepasst werden (siehe Kapitel 5.2.2). Somit ergibt sich dann für die Ähnlichkeit $vectorSim_r(u, n)$ der Konzeptvektoren zweier Benutzer u und n :

$$vectorSim_r(u, n) = \begin{cases} corr_r(u, n) & \text{wenn } corr_r(u, n) > 0 \\ 0 & \text{sonst} \end{cases} \quad (5.37)$$

5.2. Algorithmen

Indem der Mittelwert aus den Ähnlichkeiten $vectorSim_r(u, n)$ für alle Konzeptvektoren errechnet wird, kann nun die Ähnlichkeit $userConceptSim(u, n)$ zwischen den beiden Benutzern u und n gebildet werden.

$$userConceptSim(u, n) = \frac{\sum_{r \in S} vectorSim_r(u, n)}{|S|} \quad (5.38)$$

Sind nun die Ähnlichkeiten zwischen dem aktuellen Benutzer u und den Benutzern seiner Nachbarschaft ermittelt, kann die Vorhersage $pr(u, e)$ auf das Element errechnet werden. Analog dem in Kapitel 5.2.2 beschriebenen Algorithmus wird zuerst die Nachbarschaft $N(u, k)$ des Benutzers u gebildet (siehe Gleichung 5.8). Dabei wird jedoch das in diesem Kapitel vorgestellte Ähnlichkeitsmaß $userConceptSim(u, n)$ zum Vergleich zweier Benutzerprofile verwendet. Die Bestimmung der Vorhersage $pr(u, e)$ erfolgt dann mittels einer zur Gleichung 5.9 nahezu identischen Formel, bei der jedoch $userConceptSim(u, n)$ zur Gewichtung der Bewertungen von den Benutzern der Nachbarschaft verwendet wird:

$$pr(u, e) = \bar{r}_u + \frac{\sum_{n \in N(u, k)} userConceptSim(u, n) \cdot (r_{n, e} - \bar{r}_n)}{\sum_{n \in N(u, k)} userConceptSim(u, n)} \quad (5.39)$$

Dieses Verfahren lässt sich auch auf Elemente anwenden, welche anhand von Themengebieten klassifiziert sind, sofern diese anhand einer Taxonomie hierarchisiert sind. Diese Themengebiete bilden dann Eigenschaften, welche die Elemente beschreiben und den Interessen eines Benutzers entsprechen. Da diese Auszeichnung im Vorfeld geschehen ist, entfällt hierbei der aufwändige Schritt des *Preprocessing*, bei dem zunächst die *Schlüsselworte* der Elemente ihren entsprechenden und eindeutigen Konzepten zugewiesen werden müssen.

Insbesondere im *E-Commerce* sind die Elemente oftmals bereits in einer Taxonomie angeordnet. Der dort vorhandene Produktkatalog hierarchisiert die Produkte des *Online-Shops*, wodurch sich dieses Verfahren gerade in einer solchen Domäne einfach und ohne großen Zusatzaufwand einsetzen ließe.

5.2.5 Semantic Recommender

Im Gegensatz zu den bisher vorgestellten Verfahren verwenden „echte“ *Semantic Recommender* keine Vektoren zur Beschreibung von Elementeneigenschaften und Benutzerinteressen. In diesen bilden die Elemente und ihre Eigenschaften ein Netzwerk von Informationen, welches durch einen gerichteten Graph \mathcal{G} , einer sogenannten Ontologie \mathcal{O} , beschreiben ist. Diese Ontologie dient der Modellierung der Domäne und bildet dabei die *Wissensbasis* des *Recommenders*. In einer Ontologie entsprechen die Konzepte \mathcal{C} den Knoten \mathcal{V} eines Graphen \mathcal{G} , welche über die gerichteten Kanten $\mathcal{R} \equiv \mathcal{V}$ miteinander verbunden sind. Dabei bilden die Kanten nicht nur eine einfache Verbindung zwischen zwei Konzepten, sondern tragen eine zusätzliche Information in Form einer Benennung. Diese Benennung ermöglicht es, einer Kante des Graphen und damit der durch sie repräsentierten Eigenschaft eine Bedeutung (*Semantik*) zuzuweisen.

In den *Semantic Recommendern* bilden die Menge der Repräsentationen $Rep(e)$ der Elemente $e \in E$ sowie zusätzliche Daten über die Domäne die verwendete Ontologie \mathcal{O} des Systems. Dabei stellt $R \in \mathcal{R}$ die benannten Eigenschaften des Elementes e dar und $C \in \mathcal{C}$ der Wert dieser Eigenschaft. Dabei kann C auch die Repräsentation eines Elementes referenzieren, wodurch beispielsweise eine Hierarchie von Elemente realisiert werden kann.

$$Rep(e) := \{(e, R, C) \mid R \in \mathcal{R} \wedge C \in \mathcal{C}\} \quad (5.40)$$

Durch eine Analyse der Eigenschaften sowie ihrer Werte werden dann einem Benutzer u diejenigen Elemente vorgeschlagen, welche die höchste Ähnlichkeit zu den im Profil $Prof(u)$ des Benutzers u enthaltenen und gewünschten Elementeigenschaften aufweisen. Dieses Benutzerprofil, bestehend aus den Tupeln der Elementrepräsentationen $Rep(e)$ und der auf sie abgegebenen Bewertungen r , bildet seinen subjektiven Ausschnitt auf die Ontologie \mathcal{O} und ist definiert durch:

$$Prof(u) := \{(e', r_{e'}) \mid e' \in E(u) \wedge r_{e'} = r_{u,e'}\} \quad (5.41)$$

Um nun dem Benutzer ein Element vorzuschlagen, wird dieses anhand seiner Eigenschaften mit dem Profil des Benutzers verglichen, wobei die Signifikanz einer Eigenschaft anhand von Regeln definiert wird. Diese werden im Vorfeld bestimmt und eignen sich somit nur für diesen einen festgelegten Anwendungsfall. Im Gegensatz dazu arbeiten domänenunabhängige Algorithmen auf der Struktur des durch die Ontologie \mathcal{O} gebildeten Graphen \mathcal{G} , indem diese beispielsweise die Verbindungen zwischen den Konzepten analysieren.

Im Folgenden wird exemplarisch ein solches domänenunabhängiges Verfahren für einen *Semantic Recommender* vorgestellt, welches von [Lémdani et al., 2011] vorgeschlagen wurde. Das Verfahren basiert auf der Idee der *Property-Propagation*, bei der sich die Ähnlichkeiten zwischen den Konzepten nach und nach iterativ durch eine lokale Abbildung des Graphen ausbreiten. Durch die Iteration können auch entfernt liegende Eigenschaften einen Einfluss auf die Ähnlichkeiten haben, so dass auch solche Zusammenhänge erfasst werden können, die mit einer einfachen Analyse der direkten Nachbarschaft oder einem einfachen Vergleich zweier Elemente unerkannt geblieben wären.

Um die *semantische* Ähnlichkeit $semanticSim(i, j)$ zwischen zwei Konzepten i und j der Ontologie zu ermitteln, werden mehrere Schritte durchgeführt. Im ersten Schritt werden die Ähnlichkeiten von identischen Konzepten i und j mit 1 initialisiert, während die Ähnlichkeit zweier unterschiedlicher Konzepten 0 entspricht.

$$semanticSim_0(i, j) = \begin{cases} 1 & \text{wenn } i = j \\ 0 & \text{sonst} \end{cases} \quad (5.42)$$

Im nächsten Schritt werden dann die Ähnlichkeiten $semanticSim(i, j)$ zwischen zwei Konzepten i und j iterativ berechnet. Auch hier ist ihre Ähnlichkeit $semanticSim(i, j) = 1$, falls i und j das selbe Konzept darstellen, ansonsten errechnet sich die Ähnlichkeit als durchschnittliche *semantische* Ähnlichkeit der zu i und j gehörenden Paare $(i', j') \in P_{i,j}$. Die Menge $P_{i,j}$ von

5.2. Algorithmen

Konzeptpaaren entspricht allen möglichen Kombinationen der mit den Konzepten i respektive j über die selbe Relation R verbundenen Konzepte i' und j' ,

$$\begin{aligned}
 \text{semanticSim}_{it+1}(i, j) &= \begin{cases} 1 & \text{wenn } i = j \\ \sum_{(i', j') \in P_{i,j}} \frac{\text{semanticSim}_{it}(i', j')}{|P_{i,j}|} & \text{sonst} \end{cases} \quad (5.43) \\
 \text{mit } P_{i,j} &= \{(i', j') \mid \exists R \in \mathcal{R} : R(i, i') \wedge R(j, j')\}
 \end{aligned}$$

Für $k = 0$ betrachtet die *semantische* Ähnlichkeit $\text{semanticSim}_1 i, j$ nur die Eigenschaften, welche ausschließlich i und j beschreiben, oder anders ausgedrückt, genau die Konzepte, welche in der durch einen Graphen beschriebenen Ontologie \mathcal{O} genau den Abstand 1 zu den beiden Konzepten i und j aufweisen.

Für ein $k > 0$ werden für die Bestimmung der Ähnlichkeit $\text{semanticSim}_{it+1}(i, j)$ auch diejenigen Ähnlichkeiten der Konzepte verwendet, die im vorhergehenden Schritt it ermittelt wurden. Intuitiv bedeutet dies, dass nun auch die Ähnlichkeiten von Konzepten in die Berechnung einfließen, welche im Graphen den Abstand k zu den beiden Konzepten i und j aufweisen.

Die Bestimmung der Konzeptähnlichkeiten erstreckt sich dabei über alle Konzepte der Ontologie \mathcal{O} , auch wenn für die eigentliche Berechnung der Vorhersagen $pr(u, e)$ nur die Ähnlichkeiten zwischen den Repräsentationen der beiden Elemente e und e' von Interesse sind. Um nun die Vorhersage $pr(u, e)$ für einen Benutzer u auf das Element e zu ermitteln, werden die Ähnlichkeiten $\text{semanticSim}_{it+1}(e, e')$ zwischen dem vorzuschlagenden Element e und allen der bereits vom Benutzer u bewerteten Elementen $(e, r_{e'}) \in Prof(u)$ bestimmt. Diese bilden dann mit ihren entsprechenden Bewertungen $r_{e'}$ eine gewichtete Summe, welche den Wert der Vorhersage $pr(u, e)$ ergibt.

$$pr(u, e) = \frac{\sum_{(e', r_{e'}) \in Prof(u)} \text{semanticSim}_{it+1}(e, e') \cdot r_{e'}}{\sum_{(e', r_{e'}) \in Prof(u)} \text{semanticSim}_{it+1}(e, e')} \quad (5.44)$$

Dabei ist der beste Wert für die Iterationen it abhängig vom gewählten Datensatz und muss im Vorfeld ermittelt werden. In der Evaluierung von [Lémdani et al., 2011] wurde ein Wert von $it = 10$ vorgeschlagen. Dieser ermöglichte in dem dort verwendeten Datensatz eine höhere Konsistenz zwischen den Ähnlichkeiten von sehr ähnlichen Elementen als eine geringere Anzahl an Iterationen. Dabei wurde jedoch nicht auf den Einfluss des Wertes auf die Genauigkeit der Vorschläge eingegangen.

Kapitel 6

Evaluierung der Verfahren

Recommender Systeme weisen verschiedene Kriterien auf, anhand derer sie sich in ihrem Betrieb miteinander vergleichen lassen. Um eine objektive Vergleichbarkeit der in Kapitel 5 vorgestellten Algorithmen zu gewährleisten, wurden diese in einem gemeinsamen System implementiert und konnten so unter gleichen Bedingungen evaluiert werden. Es wurden für die Evaluierung aller Algorithmen identische Testdaten bestehend aus den öffentlich verfügbaren und anonymisierten Bewertungen des MovieLens⁴⁶ *Recommenders* verwendet. Diese Daten finden in der Literatur der *Recommender Systeme* eine weite Verbreitung bei der Evaluierung der dort beschriebenen Verfahren [Herlocker et al., 2000; Sarwar et al., 2001; Mobasher et al., 2003; Candillier et al., 2007; Gunawardana and Meek, 2008; Park and Tuzhilin, 2008; Ge et al., 2010].

Ausgehend von diesen Daten wurde ein System realisiert, welches unter der Verwendung von Informationen aus weiteren Quellen seinen Benutzern auf Basis ihrer persönlichen Vorlieben Filme von potentiellem Interesse vorschlägt. Während in Domänen wie beispielsweise dem Einkauf von Consumer-Electronics normalerweise keine persistente *Personalisierung* verwendet wird⁴⁷, bietet die Domäne der Filme den Vorteil der Existenz von subjektiven und in der Regel dauerhaft bestehenden Vorlieben der Benutzer, wodurch die Erzeugung lernender Benutzerprofile ermöglicht wird.

Anhand dieser Benutzerprofile generieren die in Kapitel 5 vorgestellten Verfahren ihre Vorhersagen und Empfehlungen. Sie eignen sich auch für weitere Domänen, in denen objektiv sehr ähnliche Elemente gleichen Typs mit jedoch subjektiv unterschiedlich wahrgenommenen Eigenschaften immer wieder bezogen werden und diese subjektiven Kriterien entscheidend für den persönlichen Nutzen eines Elementes sind. Solche Domänen umfassen unterschiedliche Bereiche wie beispielsweise Kleidungsstücke [Ghani and Fano, 2002], Zigarren [Zanker et al., 2006] oder Musikstücke [Chedrawy and Abidi, 2009].

Die implementierten Algorithmen werden im Folgenden anhand verschiedener Kriterien miteinander verglichen. So lässt sich ermitteln, inwieweit die Verfahren höheren *semantischen* Grades die traditionellen Verfahren durch die Verwendung *semantische* Technologien verbessern

⁴⁶<http://www.movielens.org/> Zugriff: 15.Juli 2011

⁴⁷aufgrund der dort einmalig stattfindenden Interaktion werden eher *User Feedback*-Schleifen zur Erfassung der Benutzerinteressen eingesetzt

6.1. Evaluierungsdaten

können. Dabei wird ein besonderes Augenmerk auf die Genauigkeit der erzeugten Vorhersagen und Empfehlungen gelegt.

6.1 Evaluierungsdaten

MovieLens ist eine Webseite, welche ein System bereitstellt, in dem Benutzer auf einer Skala von eins bis fünf subjektive Bewertungen auf Filme abgeben können. Dabei entspricht die Eins dem geringsten Grad und die Fünf dem höchsten Grad an Interesse. Die Bewertungen werden verwendet, um den angemeldeten Benutzern Vorschläge auf Filme von potentiellm Interesse anzuzeigen. Seit einigen Jahren werden Auszüge aus der Bewertungsdatenbank von MovieLens in einem anonymisierten Format der Öffentlichkeit zur Verfügung gestellt. Der Datensatz ist einer der wenigen einer gewissen Größe, über den frei verfügt werden kann und der auf realen, dass heißt von Menschen erzeugten Daten basiert. Dies erklärt seine breite Verwendung im Forschungsumfeld der *Recommender Systeme* zur Evaluierung von Algorithmen.

Zur Evaluierung der hier vorgestellten Systeme wird der MovieLens Datensatz⁴⁸ mit einer Million Bewertungen verwendet. Konkret enthält dieser Datensatz 1.000.209 Bewertungen auf einer Skala von eins bis fünf mit dem Zeitpunkt ihrer Erfassung, die von 6.040 Benutzern auf 3.953 Filme abgegeben wurden. Zusätzlich sind demographische Daten über die Benutzer wie Geschlecht, ausgeübter Beruf und Alter beinhaltet.

Schlüsselwörter

Anhand der im MovieLens Datensatz vorhandenen Filmtitel sowie deren Erscheinungsjahren wurde für jeden Eintrag eine Suchanfrage an die IMDB-Webseite⁴⁹ gestellt und dann die in Form einer den Film beschreibenden *HTML*-Webseite gegebene Antwort verarbeitet. IMDB ist ein im *World Wide Web* frei verfügbares, für die Benutzer offenes Verzeichnis über Filme, Serien sowie deren Darsteller welches im August 2011 über 2 Millionen Filmeinträge und mehr als 4 Millionen Personen⁵⁰ beinhaltet. Dabei stellte die Ermittlung der zu den Filmen gehörenden Webseiten aufgrund der zur IMDB inkompatiblen Benennung im MovieLens Datensatz ein Problem dar. Von Vorteil hatte sich die von IMDB angebotene Analyse der Namen bei der Suche erwiesen, da hierbei viele der im Datensatz vorhandenen Schreibfehler sowie fehlerhaft angegebenen Erscheinungsjahre automatisch aufgelöst wurden. Allerdings konnten erst nach manuell durchgeführter Korrektur der Daten die Filme auch ihren Webseiten korrekt zugewiesen werden. Aus den entsprechenden Webseiten wurden dann die beschreibenden Daten wie Regisseur, Darsteller und Erscheinungsjahr des Films extrahiert. Neben diesen Angaben enthält die Webseite eines jeden Films auch von den Benutzern generierte *Tags* in englischer Sprache, welche stichwortartig die Handlung des Films beschreiben. Diese *Tags* wurden ebenfalls extrahiert und die in ihnen auftretenden *Stopworte* anhand einer Liste der häufigsten englischen Worte⁵¹ entfernt. Die so

⁴⁸<http://www.grouplens.org/node/73> Zugriff: 15. Juli 2011

⁴⁹<http://www.imdb.com> Zugriff: 22. August 2011

⁵⁰<http://www.imdb.com/stats> Zugriff: 22. August 2011

⁵¹<http://snowball.tartarus.org/algorithms/english/stop.txt> Zugriff: 15. September 2011

bearbeiteten *Tags* wurden dann für den späteren Zugriff in einer Datenbank mit den übrigen ermittelten Informationen über den Film gespeichert.

Konzepte und ihre Taxonomie

Nach der Extraktion der *Tags* eines Films wurden die zu den Worten gehörenden *Synsets* mit Hilfe einer *lexikalischen* Datenbank bestimmt, um so die den Inhalt repräsentierenden Konzepte zu erhalten. Als *lexikalische* Datenbank kam eine lokale Kopie von WordNet⁵² in der aktuellen Version 3.0 zum Einsatz. WordNet stellt für die englische Sprache den de facto Standard als Wörterbuch dar und findet eine weite Verbreitung in der maschinellen Verarbeitung von Texten. Neben der Bestimmung der Wortart eines angefragten Wortes und seiner Grundform, falls es aufgrund von Flexion oder Deklination von dieser abweicht, können für Nomen auch deren Ober- und Unterbegriffe ermittelt werden.

Aufgrund der Taxonomie der Nomen wurden bei der Ermittlung der die Filme beschreibenden *Synsets* nur diejenigen verwendet, welche auch aus Nomen bestehen. Verben und Adverbien wurden folglich ignoriert. Unter Verwendung von WordNet wurden den aus den *Tags* extrahierten Nomen ihre *Synsets* anhand des in Kapitel 5.2.4 beschriebenen Verfahrens zugewiesen. Eine manuelle Untersuchung der Ergebnisse in Form von Stichproben konnte dabei einen Anteil von 10 % fehlerhaft zugewiesener *Synsets* bestimmen, der bedingt durch die automatische Zuordnung und dem naivem Algorithmus auftrat. Eine höhere Qualität könnte nur erreicht werden, wenn die *Synsets* in einem zeitintensiven Prozess manuell bestimmt oder fortschrittlichere Verfahren aus dem Umfeld der Computerlinguistik verwendet würden, welche eine geringere Fehlerquote aufweisen.

Formal definierte Daten

Ausgehend von den *URLs* der die jeweiligen Filme beschreibenden IMDB-Webseiten, wurden *SPARQL*-Queries (siehe Kapitel 3.1.6) an die *SPARQL*-Schnittstelle des Linked Movie Database-Projekts⁵³ gestellt. Neben einer Abbildung der in der IMDB enthaltenen Daten über die Filme in das *RDF*-Format [Hassanzadeh and Consens, 2009], enthält die Datenbank auch ihre jeweiligen global eindeutigen Identifikatoren. Durch die gemeinsame Verwendung dieser Identifikatoren können zusätzliche Informationen über die Filme in der Datenbank des dbpedia-Projekts⁵⁴ bestimmt werden.

Ziel des dbpedia-Projekts ist die Erzeugung einer formal definierten Sicht auf die Einträge der Wikipedia⁵⁵ im *RDF*-Format [Auer et al., 2007]. Dazu werden die Wikipedia-Artikel analysiert, die dort vorhandenen Informationen extrahiert und in das *RDF*-Format (siehe Kapitel 3.1.3) überführt. Insbesondere die zu den Artikeln gehörenden „Boxen“ bieten eine Möglichkeit, die die Artikel beschreibenden und in ihrem Sinn eindeutig begreifbaren Daten zu erhalten. Jede dieser Boxen enthält dabei unterschiedliche Informationen auf das durch den Artikel beschriebene Objekt. So beinhaltet beispielsweise die eine Stadt beschreibende Box das Land, in welchem

⁵²<http://wordnet.princeton.edu/wordnet/> Zugriff 18. September 2011

⁵³<http://www.linkedmdb.org/> Zugriff 4. Oktober 2011

⁵⁴<http://dbpedia.org> Zugriff 12. Oktober 2011

⁵⁵<http://www.wikipedia.org> Zugriff: 29. Juni 2011

6.2. Implementierung

sich diese befindet, sowie deren aktuelle Einwohnerzahl. Im Falle eines Filmes sind Darsteller, Erscheinungsjahr oder Regisseur dort angegeben. Um die Daten aus den Boxen zu extrahieren, existieren unterschiedliche und an deren jeweilige Struktur angepasste Templates, welche die Daten der Boxen auf vordefinierte *RDFS*-Klassen abbilden.

Zudem sind Wikipedia-Artikel auch einer oder mehreren Kategorien zugeordnet, welche in der Datenbank des dbpedia-Projekts in einer formalen Modellierungssprache beschrieben und ferner untereinander hierarchisiert sind. Diese Taxonomie der Kategorien bildet die Grundlage für die verwendeten formal definierten Daten über die Filme. Ausgehend von der Kategorie der Filme als Wurzel der Taxonomie, wurden sie unter Verwendung von *SPARQL*-Queries aus der Datenbank des dbpedia-Projekts extrahiert. Diese Daten wurden dann in einer lokalen Kopie gespeichert und lagen somit zum lokalen Zugriff bereit.

Konnten die Daten eines Films weder in der dbpedia Datenbank noch in der Linked Movie Database automatisch ermittelt werden, so wurde eine manuelle Suche in der Wikipedia durchgeführt. Mit Hilfe der so erhaltenen *URL* des Wikipedia-Artikels konnte dann auf den entsprechenden Eintrag in der dbpedia geschlossen werden. Trotz alledem konnten die Informationen für 9 % der Filme des Datensatzes nicht ermittelt werden, da für sie zum aktuellen Zeitpunkt (November 2011) noch keine Wikipedia-Artikel existierten und diese somit nicht in der Datenbank der dbpedia enthalten waren.

6.2 Implementierung

Die in Kapitel 5 vorgestellten Algorithmen wurden in der Programmiersprache *JAVA* prototypisch implementiert, so dass eine Evaluierung der Verfahren durchgeführt werden konnte. Dabei folgte die Architektur des Systems der in Kapitel 2 vorgestellten allgemeinen Struktur für *Recommender Systeme* (siehe Abbildung 2.2). Abbildung 6.1 zeigt die Paket-Struktur des implementierten Systems. In der Abbildung ist das Paket, welche die Klassen für die Evaluierung beinhaltet, nicht dargestellt. Dies ist darin begründet, dass diese Klassen im lauffähigen System keinerlei Verwendung finden, sondern nur dazu genutzt werden, die verschiedenen implementierten Verfahren miteinander anhand unterschiedlicher Kriterien quantitativ zu vergleichen. Auch wenn in der Implementierung ein gewisser Grad der Effizienz durch hinreichend schnelle Bibliotheken im Austausch zu den Standard-Bibliotheken der verwendeten Programmiersprache sowie eine Optimierung in der Implementierung der entsprechenden Algorithmen erzielt wurde, war die *Skalierbarkeit* aufgrund des *Proof of Concept*-Charakters des Systems zu keiner Zeit eine Zielsetzung.

core-Paket

Die Klassen des *core*-Pakets dienen dazu, die verschiedenen, zum Betrieb notwendigen Klassen, wie beispielsweise das Speichersystem oder die *Wissensbasis*, zu instantiiieren und insbesondere für das *Frontend* zugreifbar zu machen. Sie bieten jedoch keine darüber hinausgehende Funktionalität an, so dass dieses Paket von daher nicht in Abbildung 6.1 dargestellt ist.

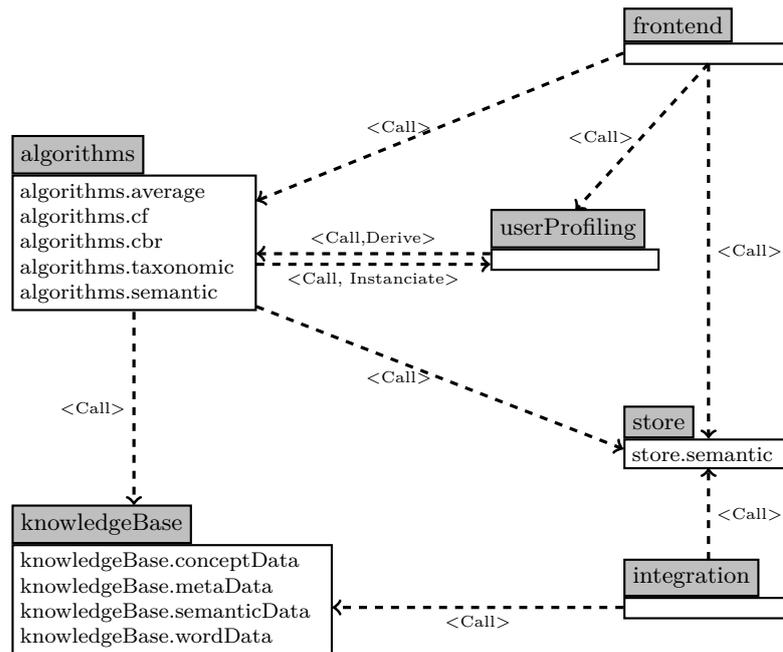


Abbildung 6.1 – Paket-Übersicht des Systems in UML-Notation

integration-Paket

Das *integration*-Paket beinhaltet alle Klassen, welche die in Kapitel 6.1 beschriebenen Evaluierungsdaten aus externen Quellen abrufen und in das System integrieren. Hierzu existieren Klassen, welche die in der Struktur des Movie Lens Datensatzes vorliegenden Text-Dateien für die Benutzer und Bewertungen einlesen, parsieren und diese Daten dann in das Repräsentationsformat des Systems zur persistenten Speicherung überführen.

Neben dieser, auf Dateiebene stattfindenden Integration, existieren auch Klassen, welche zu den Elementen gehörenden Informationen aus dem *World Wide Web* beziehen. Dazu greifen diese per *HTTP*-Protokoll auf die entsprechenden Webseiten zu. Aufgrund des verwendeten Use-Cases, also in diesem Fall Multimedia-Daten in Form von Filmen, handelt es sich hierbei um die *IMDB*-Webseite, welche die größte öffentliche Datenbank für Informationen über Filme und deren Darsteller ist. Um Informationen über einen Film zu erhalten, wird eine Anfrage an das Suchformular der Webseite gestellt und anschließend die durch eine *HTML*-Webseite repräsentierte Antwort parsiert. Dazu wurde ein Parser händisch durch eine manuelle Analyse der Webseite und deren Quellcode implementiert. Aufgrund dieser Herangehensweise ist zu beachten, dass die Parsierung im System fest codiert ist und sich somit nur für die aktuelle Struktur der Webseiten (Stand: August 2011) eignet. Falls sich diese in Zukunft ändern sollte, müsste der Parser entsprechend angepasst werden.

Eine solche Problematik existiert für die Integration der formal definierten Daten nicht. Hierbei werden an die *SPARQL*-Endpoints der dbpedia⁵⁶ und des Linked Movie Database-Projekts⁵⁷ Anfragen in Form von *SPARQL*-Queries (siehe Kapitel 3.1.6) gestellt. Auch wenn dabei ebenfalls *HTTP* als Transportprotokoll verwendet wird, liegen hier die Antworten in Form

⁵⁶<http://dbpedia.org/sparql> Zugriff: 28. Oktober

⁵⁷<http://data.linkedmdb.org/sparql> Zugriff: 28. Oktober

6.2. Implementierung

von strukturierten und eindeutigen Daten im *SPARQL-XML*-Format vor. Diese können aufgrund des standardisierten Formats sofort verarbeitet werden, ohne dass die Struktur der Antworten im Vorfeld manuell und aufwändig analysiert werden muss. Die über die Elemente erhaltene Informationen werden dann in das intern verwendete *RDFS*-Format überführt und im *Triple-Store* gespeichert.

userprofiling-Paket

Das *userprofiling*-Paket dient als Schnittstelle des *Frontends* zum System, um aus den von den Benutzern abgegebenen Elementbewertungen ihr jeweiliges, ihre Vorlieben abbildendes Profil zu erzeugen. Dazu leitet es die Bewertungen und das Profil des jeweiligen Benutzers an die Klassen des aktuell eingesetzten Verfahrens weiter, so dass jenes entsprechend seiner Arbeitsweise das Benutzerprofil aktualisieren kann. Neben dieser Aufgabe verfügt es über keine weitergehende Funktionalität.

algorithms-Paket

Das *algorithms*-Paket beinhaltet die Implementierungen der verschiedenen Verfahren. Neben den eigentlichen, für die Generierung der Vorschläge und Empfehlungen zuständigen Methoden, befinden sich auch solche zur Erzeugung der Benutzerprofile und Elementrepräsentationen in den entsprechenden Klassen.

Für die Verfahren eines jeden *semantischen* Grades existiert dabei jeweils ein eigenes Paket (siehe Abbildung 6.1). Zusätzlich zur Generierung der Repräsentationen und Vorschläge haben die Implementierungen der Algorithmen die Aufgabe des *Cachings* der Zwischenergebnisse sowie der bereits erzeugten Repräsentationen. Dadurch kann der Zugriff auf das Speichersystem reduziert werden, und auch die kostenintensive Überführung der Elementdaten in die entsprechende Repräsentation des verwendeten Verfahrens muss nur einmalig durchgeführt werden. Somit kann auf Kosten des Speicherverbrauchs eine große Steigerung der Performanz erzielt werden.

store-Paket

Die im *storage*-Paket befindliche *Datenhaltungs*-Komponente ist zweigeteilt in eine *H2-SQL*⁵⁸ in der Version 1.3.159 verwendende Relationale Datenbank und in einen *Sesame*⁵⁹ verwendenden *Triple-Store* der Version 2.6.1. Beide Bibliotheken sind in *JAVA* implementiert und erlauben einen nativen Zugriff des Systems auf die von ihnen bereitgestellten Funktionen.

Während die Bewertungen der Benutzer in der relationalen Datenbank gespeichert werden, befinden sich die Beschreibungen der Elemente im *Triple-Store*. Diese Zweiteilung wurde aufgrund der großen Anzahl der Bewertungen gewählt, da ihre Speicherung im *Triple-Store* eine zusätzliche Anzahl von mehr als einer Million gespeicherter Statements bedeuten würde. Bedingt durch die Graphen-Struktur des *RDFS*-Modells kann auf diese Daten nur ineffizient zugegriffen werden, so dass der Einsatz einer Tabelle für die Bewertungen in einem getrennten Speichersystem die Performanz des Systems erheblich steigert.

⁵⁸<http://www.h2database.com/html/main.html> Zugriff 14. Juli 2011

⁵⁹<http://www.openrdf.org/> Zugriff 31. Juli 2011

Das *Domänen*-Modell der Elemente und die weiteren zu ihnen gehörenden Daten ist im Gegensatz dazu im *RDFS*-Format beschreiben. Abbildung 6.2 zeigt in schematischer Darstellung die Klassen und Relationen des Modells. Auch wenn die Verwendung von *URIs* als eindeutiger Kennzeichner für die Elemente ausreicht, besitzen diese noch die zusätzliche *Property* `hasNumericalId`, um sie ihren Bewertungen in der relationalen Datenbank zuweisen zu können. Diese *Property* ist nicht in Abbildung 6.2 dargestellt, da sie an sich kein Teil der Modellierung ist, sondern vielmehr der Verbindung zwischen dem *Triple*-Store und der relationalen Datenbank in Form eines Fremdschlüssels dient. Zwar unterstützen relationale Datenbanken auch Zeichenketten als Schlüssel, dennoch ist es effizienter, Zahlen zur Unterscheidung der Einträge zu verwenden. In der Abbildung sind die Klassen der Ontologie als Ellipsen und ihre Eigenschaften mit Literalen als Datentyp durch Rechtecke dargestellt. Dabei stellt `sr` den Namespace des Systems `http://semantic-recommender.org` dar, sowie `rdfs` und `skos` die Namespaces der verwendeten Vokabulare *RDFS* respektive *SKOS*. Ferner stellen die Relationen vom Typ `rdfs:label` den für Menschen verständlichen Bezeichner von Personen, Filmtiteln oder Kategorien dar.

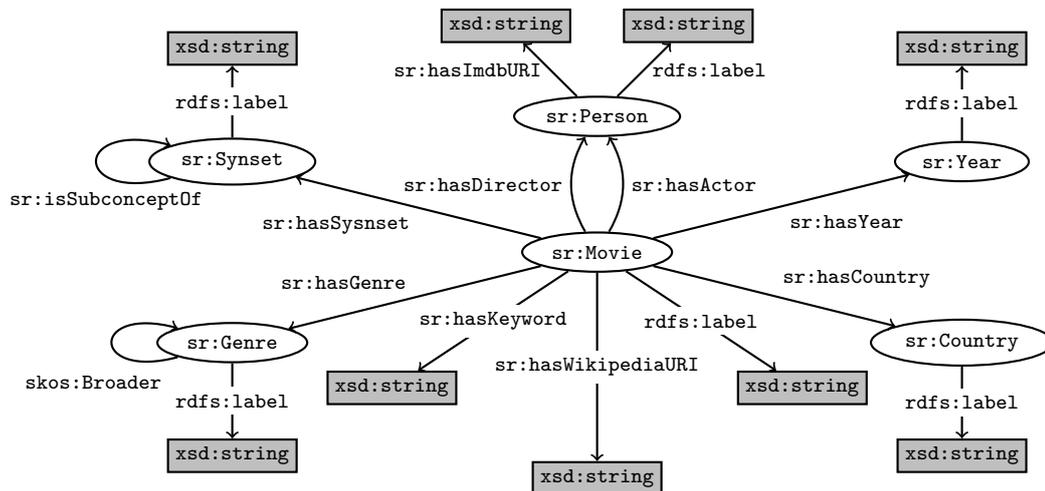


Abbildung 6.2 – Schematische Darstellung der verwendeten RDFS-Modellierung

Auch wenn für die Verfahren des entsprechenden *semantischen* Grades in einem „realen“ System die Verwendung einer formalen Modellierungssprache wie beispielsweise *OWL* notwendig wäre (siehe Kapitel 4.3), wird im hier eingesetzten Evaluierungs-System *RDFS* zur Beschreibung der Elemente verwendet. In einem solchen *Proof of Concept* System ist eine „einfache“ Sprache wie *RDFS* ausreichend, da eine große Aussagekraft, wie *OWL* sie bereitstellen würde, schlichtweg nicht zwingend benötigt wird. Die beim Einsatz eines „realen“ Systems auftretenden Probleme der Datenintegration und Austauschbarkeit der *Wissensbasis* spielen hier nur eine untergeordnete Rolle.

knowledgeBase-Paket

Das *knowledgeBase*-Paket enthält unter anderem die Klassen und Methoden, die einer *linguistischen* Analyse der die Elemente beschreibenden *Schlüsselworte* dienen. Neben den Satzzeichen werden auch die *Stopworte* aus der Menge der *Schlüsselworte* eines Elementes entfernt,

6.2. Implementierung

da diese nur über eine geringe Signifikanz für die Bedeutung der Elementbeschreibungen verfügen. Dazu wird eine Liste der häufigsten Worte der englischen Sprache verwendet. Weiterhin existieren Methoden, um das globale Auftreten eines bestimmten Wortes zu ermitteln. Die so verarbeiteten *Schlüsselworte* bilden dann einen Teil der Elementrepräsentationen.

Im *knowledgeBase*-Paket befinden sich zudem die Klassen, welche den Zugriff auf die WordNet API bereitstellen sowie der Ermittlung der zu den *Schlüsselworten* der Elemente gehörigen *Synsets* dienen. Als problematisch hat sich dabei die für den Zugriff auf die linguistische Datenbank eingesetzte JAVA-Bibliothek *JAWS*⁶⁰ erwiesen. Auch wenn diese den größten Funktionsumfang der verschiedenen im Zuge der Implementierung betrachteten JAVA-Bibliotheken für WordNet beinhaltet, bietet sie keinen Zugriff auf den WordNet-Identifikator für die *Synsets*. Deshalb musste ein eigener Identifikationsmechanismus verwendet werden, der jedoch bei der persistenten Speicherung der zu den Elementen gehörenden *Synsets* keine bijektive Abbildung auf die der *linguistischen* Datenbank zulässt. Werden neue Elemente in das System integriert, so muss das Verfahren der *Synset*-Bestimmung erneut für alle Elemente durchgeführt werden. Eine persistente Übersetzung aller *Synsets* aus der Datenbank in das Format des Systems würde eine Lösung für dieses Problem darstellen. Dafür müssten aber zunächst alle Deklinationen von Nomen manuell erzeugt werden, da die Bibliothek selbst keine direkte Ermittlung der Deklinationen eines gegebenen Wortes zulässt, sondern nur die Grundform der Worte eines *Synset* liefern kann. Dies erscheint jedoch aufgrund der großen Anzahl an Worten nicht praktikabel.

Die ermittelten *Synsets* sowie ihre Hyperonym-Relationen in Form von *RDF*-Statements werden persistent im *Triple*-Store gespeichert. Für einen beschleunigten Zugriff wird diese Taxonomie durch die entsprechende Klasse im Arbeitsspeicher gehalten.

Neben den Klassen für die *linguistische* Verarbeitung der Elementdaten befinden sich im *knowledgeBase*-Paket auch solche, die einen vereinfachten Zugriff auf die durch *RDFS* modellierten Elemente sowie ihre Eigenschaften ermöglichen. Weiterhin werden dort die Elemente sowie ihre Relationen untereinander in das interne Repräsentationsformat des Systems überführt und im Arbeitsspeicher bereitgehalten. So wird der zeitintensive Zugriff auf den Hintergrundspeicher minimiert.

frontend-Paket

Das *Frontend* des Systems dient der Visualisierung der Elemente für die Benutzer sowie der Anzeige der von den Algorithmen generierten Empfehlungen. Weiterhin stellt es eine Möglichkeit zur Bewertung von Elementen durch den Benutzer bereit, so dass im Hintergrund ein ihn repräsentierendes Profil erzeugt werden kann.

Für die hier vorliegende Arbeit wurde das in Abbildung 6.3 gezeigte *Frontend* als eine *HTML*-basierte Web-Oberfläche realisiert, welche durch einen *Tomcat*-Webserver in der Version 6.0.33⁶¹ bereitgestellt wurde. Dieser Webserver ermöglicht die Verwendung von Webseiten mit dynamischem Inhalt mittels *Java Server Pages (JSPs)*. *JSPs* erlauben die Einbettung von *JAVA*-Code in *HTML*-Webseiten, was im Hinblick auf die verwendeten Bibliotheken sinnvoll ist. Insbesondere die Bibliotheken, welche dem Einsatz der *Semantic Web* Technologien dienen, sind in *JAVA*

⁶⁰<http://lyle.smu.edu/~tspell/jaws/index.html> Zugriff 12. September 2011

⁶¹<http://tomcat.apache.org/> Zugriff 28. August 2011

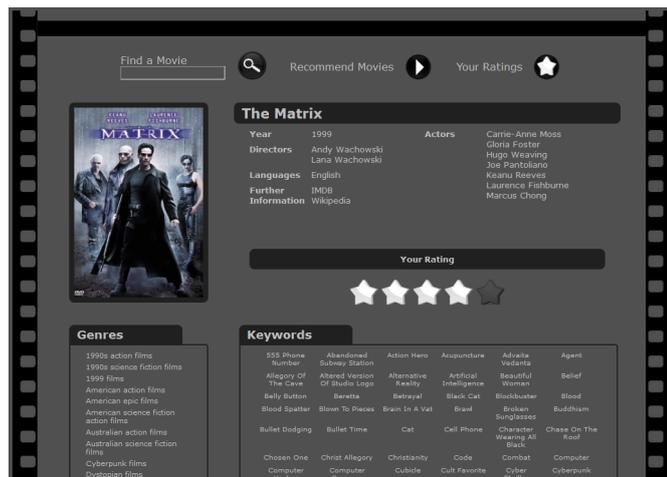


Abbildung 6.3 – Screenshot der Web-Oberfläche des Frontends

implementiert. Somit kann durch die Verwendung von *JSPs* eine nahtlose Integration aller einzelnen Algorithmen, Bibliotheken sowie des Speichersystems in das *Frontend* erreicht werden.

meta-Paket

Das *meta*-Paket ist nicht in Abbildung 6.1 dargestellt, da es lediglich die verschiedenen verwendeten Konstanten enthält sowie die Klassen, welche dem Austausch von Zwischenergebnissen zwischen den einzelnen Komponenten dienen. Dabei weisen die Instanzen der Klassen keinerlei, über die einfache Datenhaltung hinausgehende, Funktionalität auf und dienen ausschließlich als Container für Werte.

6.3 Evaluierungsmetriken

Recommender Systeme weisen viele Kriterien auf, anhand derer sich unterschiedliche Systeme quantitativ miteinander vergleichen lassen. Die verwendeten verschiedenen Metriken dabei basieren auf anerkannten statistischen Verfahren, die in der Literatur der *Recommender Systeme* eine weite Verbreitung finden. Diese Metriken eignen sich insbesondere für die Durchführung einer *Offline*-Evaluierung anhand von vorgegebenen Testdaten, wodurch diese unabhängig von den Benutzern erfolgen kann. Weiterhin wird so eine objektive Vergleichbarkeit unterschiedlicher *Recommender* anhand eindeutiger Kriterien ermöglicht.

Im Folgenden werden die verschiedenen Kriterien und ihre entsprechenden Metriken vorgestellt, anhand derer die implementierten Algorithmen des Systems evaluiert und miteinander verglichen werden. Auch wenn die Genauigkeit der gemachten Vorschläge dabei sicherlich das entscheidendste Kriterium für die Güte eines Verfahren ist, so spielen weitere Faktoren wie beispielsweise die Performanz und die Menge der vorschlagbaren Elemente (*Coverage*) im Betrieb eines *Recommenders* ebenfalls eine Rolle.

6.3. Evaluierungsmetriken

6.3.1 Genauigkeit

Die Genauigkeit der von *Recommender Systemen* ermittelten Ergebnisse hat einen starken Einfluss auf die Zufriedenheit der Benutzer, die Systeme mit passenden Vorhersagen und Empfehlungen gegenüber Systemen mit ungenauen Ergebnissen klar bevorzugen. Die Genauigkeit der Vorhersagen und Empfehlungen ist unabhängig vom Benutzer-Interface und kann daher in einem *Offline*-Verfahren bestimmt werden [Shani and Gunawardana, 2011].

Dabei müssen unterschiedliche, auf die Art der Ergebnisse des Systems angepasste, Metriken verwendet werden, um deren Genauigkeit zu erfassen. So muss bei der Evaluierung von Vorhersagen auf die Abweichung der vorhergesagten von den tatsächlichen Bewertungen eingegangen werden, während bei Empfehlungen betrachtet werden muss, inwieweit die ermittelte Menge in der Gesamtheit ihrer enthaltenen Elemente den Vorlieben des Benutzers entspricht.

Durchschnittlicher absoluter Fehler

Der durchschnittliche absolute Fehler, abgekürzt als *MAE* (*Mean Absolute Error*) entspricht der durchschnittlichen Abweichung einer errechneten Vorhersage $pr(u, e)$ von der tatsächlich vom Benutzer abgegebenen Bewertung $r_{u,e}$. Er findet eine breite Verwendung in der Evaluierung verschiedener *Recommender Systeme* [Shardanand and Maes, 1995; Breese et al., 1998; Herlocker et al., 1999; Sarwar et al., 2001; Mobasher et al., 2003]. Für jedes Vorhersagen-Bewertungs-Paar $(pr(u, e), r_{u,e})$ wird bei der Berechnung des *MAE* der absolute Fehler gleichwertig behandelt. Es werden zunächst die Fehler aller Paare aufsummiert und dann aus dem erhaltenen Wert das arithmetische Mittel gebildet. Dabei entsprechen kleinere Werte des *MAE*s einer höheren Genauigkeit der Vorhersagen. Formal ist der *MAE*(u) für einen Benutzer u definiert durch:

$$MAE(u) = \frac{1}{N} \cdot \sum_{e \in E(u)} |pr(u, e) - r_{u,e}| \quad (6.1)$$

Neben der Erfassung der Genauigkeit der errechneten Vorhersagen für jeden Bereich einer beliebigen Bewertungsskala S bietet diese Evaluierungsmetrik noch weitere Vorteile. Zum einen ist die Berechnung der Werte simpel, weshalb auch die Ergebnisse leicht nachzuvollziehen und zu verstehen sind. Zum anderen basiert der *MAE* auf etablierten statistischen Verfahren und ermöglicht es so, zwei unterschiedliche Systeme anhand der Genauigkeit ihrer gemachten Vorhersagen zu vergleichen [Herlocker et al., 2004].

Fehlerklassen von Vorhersagen

Auch wenn der *MAE* eine gute Abbildung der Genauigkeit eines Verfahrens über alle Werte der verwendeten Bewertungsskala ermöglicht, berücksichtigt er dennoch nicht die subjektive Bedeutung einzelner Werte für die Benutzer. So ist es für einen Benutzer von eher geringem Interesse, ob ein Element als „gut“ anstelle von „sehr gut“ klassifiziert wurde. Viel entscheidender ist es für ihn, dass „gute“ beziehungsweise „schlechte“ Elemente auch als solche erkannt werden.

Dabei lassen sich die generierten Vorhersagen anhand Tabelle 6.1 anhand ihrer Relevanz für den Benutzer klassifizieren. Insbesondere für die Betreiber von *Recommender Systemen* in

	empfohlen	nicht empfohlen
relevant	True-Positive (tp)	False-Negative(fn)
nicht relevant	False-Positive(fp)	True-Negative(tn)

Tabelle 6.1 – Klassifikation eines einzelnen Vorschlages für einen Benutzer

Online-Shops ist die Vermeidung fehlerhaft generierter Vorhersagen von großer Bedeutung. Die Fehlerklassen lassen sich, analog zu denen anderer Suchsysteme, wie folgt bestimmen:

- *False Negatives*: Elemente, die nicht vorgeschlagen werden, obwohl der Benutzer sie gesucht hat
- *False Positives*: Elemente, die vorgeschlagen werden, obwohl sie nicht den Vorlieben des Benutzers entsprechen

Während im *E-Commerce* Umfeld die erste Art von Fehlern gegenüber der zweiten von geringerer Bedeutung sein kann, vorausgesetzt es werden immer noch ausreichend zu den Vorlieben des Benutzers passende Elemente im großen Suchraum des Produktkataloges gefunden, sollten *False Positives* unbedingt vermieden werden. Diese Fehler führen zur Unzufriedenheit der Kunden, was sich für den Betreiber dadurch bemerkbar macht, dass diese den *Online-Shop* nicht mehr besuchen [Sarwar et al., 2000a].

Da sich die erzeugten Ergebnisse der *Recommender Systeme* von denen der klassischen Suchsysteme unterscheiden, muss die Berechnung der *False Negatives* und der *False Positives* entsprechend an deren Eigenarten angepasst werden. Besteht die Bewertungsskala des Systems nicht bereits aus einem binären Format, so muss diese zuerst in ein solches transferiert werden. Dies geschieht durch eine Einteilung in „relevante“ und „nicht relevante“ Elemente, anhand eines vorgegebenen und an den Wertebereich des jeweiligen *Recommenders* angepassten Grenzwertes.

Precision / Recall

Um die Genauigkeit der gemachten Empfehlungen $RS(u, N)$ zu bestimmen, eignet sich der durchschnittliche absolute Fehler *MAE* jedoch nicht. Hier ist nicht die Abweichung von der tatsächlichen Bewertung von Bedeutung, sondern es muss vielmehr betrachtet werden, inwieweit die ermittelte Menge der Elemente zu den Vorlieben des Benutzers passt. Dabei kommen die zwei im *Information Retrieval* weit verbreiteten Evaluierungsmetriken, die sogenannte *Precision* und der *Recall*, zur Anwendung [Cleverdon and Mills, 1966]. Die Berechnungen der *Precision* und des *Recalls* basieren auf der in Tabelle 6.1 dargestellten Klassifikation der Ergebnisse einer erzeugten Empfehlung.

Die *Precision* stellt das Verhältnis der ermittelten relevanten Elemente zu der Menge der empfohlenen Elemente dar und beschreibt damit die Wahrscheinlichkeit, inwieweit ein durch den *Recommender* empfohlenes Element auch für den Benutzer von Interesse ist. Sie ist definiert durch:

$$Precision = \frac{|True-Positives|}{|True-Positives| + |False-Positives|} \quad (6.2)$$

6.3. Evaluierungsmetriken

Der *Recall* stellt das Verhältnis der ermittelten relevanten Elemente zu der Menge aller relevanten Elemente dar und beschreibt die Wahrscheinlichkeit, dass ein Element von Interesse auch ausgewählt wird. Er ist definiert durch:

$$Recall = \frac{|True-Positives|}{|True-Positives| + |False-Negatives|} \quad (6.3)$$

Es konnte beobachtet werden, dass die *Precision* und der *Recall* sich invers zueinander verhalten und stark von der Mächtigkeit N der Ergebnismenge abhängen [Cleverdon and Mills, 1966]. Wenn die Ergebnismenge eine größere Anzahl an Elementen enthält, steigt der *Recall* während die *Precision* des Ergebnisses sinkt. Somit müssen beide Größen immer in Abhängigkeit von N und vor allem in Kombination miteinander betrachtet werden, um unterschiedliche Systeme miteinander vergleichen zu können. Die $F1$ -Metrik [Yang and Liu, 1999], welche auf dem Effektivitätsmaß E von [van Rijsbergen, 1979] basiert, stellt dabei das am stärksten verbreitete Verfahren dar, beide Werte miteinander zu kombinieren. Hierbei fließen sowohl die *Precision* als auch der *Recall* mit identischer Gewichtung in das Ergebnis ein und werden so auf einen einzigen vergleichbaren Wert abgebildet. Die $F1$ -Metrik ist definiert durch:

$$F1_N = \frac{2 \cdot Precision_N \cdot Recall_N}{Precision_N + Recall_N} \quad (6.4)$$

Auch für die Berechnung der *Precision* und des *Recalls* muss die Bewertungsskala des Systems in ein binäres Format überführt werden, falls es nicht bereits aus einer solchen besteht. Dies geschieht analog zu der bereits beschriebenen Methode durch eine anhand eines gegebenen Grenzwertes vollzogene Klassifikation der Elemente in „relevant“ und „nicht relevant“.

Der „echte“ *Recall* lässt sich jedoch nur schwer bestimmen, da dazu von den Benutzern verlangt wäre, dass diese alle Elemente des Systems betrachten und bewerten, was in der Realität nicht durchführbar ist. Insbesondere für die unvollständigen Daten, die bei einer *Offline*-Evaluierung vorliegen, müssen dabei Annahmen getroffen werden, um den *Recall* approximieren zu können. So muss davon ausgegangen werden, dass noch unbewertete Elemente auch irrelevant für den Benutzer sind. Über ihre tatsächliche Relevanz kann aufgrund fehlender Informationen keine Aussage getroffen werden, auch wenn die Elemente von subjektivem Interesse sein könnten.

Eine oftmals eingesetzte Technik ist die Aufteilung der Elemente $E(U)$, auf die der Benutzer u eine Bewertung abgegeben hat, in eine Trainingsmenge $TR(u)$, anhand der das Profil des Benutzers initialisiert wird, und in eine Testmenge $TS_k(u)$ bestehend aus k relevanten Elementen [Sarwar et al., 2000a; Sarwar et al., 2000b; Ziegler et al., 2004a; Farsani and Nematbakhsh, 2006; Zanker and Jessenitschnig, 2009]. Dabei gilt $TR(u) \cap TS_k(u) = \emptyset$. Ausgehend von diesen beiden Mengen können dann die *Precision* und der *Recall* für eine Empfehlungsmenge der Mächtigkeit N ermittelt werden. Dazu werden zunächst für das mit den Trainingsdaten $TR(u)$ initialisierte Profil die Empfehlungen $RS(u, N)$ bestimmt. Anschließend wird überprüft, inwieweit die ermittelte Menge empfohlener Elemente $RS(u, N)$ mit der Testmenge $TS_k(u)$ übereinstimmt. Es errechnen sich dann wie folgt die *Precision* und der *Recall*:

$$Precision_N = \frac{|TS_k(u) \cap RS(u, N)|}{|RS(u, N)|} \quad (6.5)$$

$$Recall_N = \frac{|TS_k(u) \cap RS(u, N)|}{|TS_k(u)|} \quad (6.6)$$

Zu beachten ist dabei, dass je nach Wahl der Parameter k und N entweder die *Precision* oder der *Recall* niemals 100 % erreichen werden. Dies gilt insbesondere für Benutzer, die weniger Bewertungen abgegeben haben, als die Empfehlungsmenge Elemente aufweist.

Wenn davon ausgegangen wird, dass die Verteilung von relevanten und nicht relevanten Elementen der Testmenge für einen einzelnen Benutzer auch der realen Verteilung aller Elemente für diesen Benutzer entspricht, so kann mit dieser Technik eine gute Approximation für die „echte“ *Precision* und den „echten“ *Recall* erreicht werden [Basu et al., 1998].

6.3.2 Performanz

Ein wichtiges Kriterium beim Einsatz eines *Recommender Systems* ist seine Performanz. Dabei ist entscheidend, dass das verwendete System Vorhersagen und Empfehlungen für die Benutzer hinreichend schnell generieren kann, da zu lange Wartezeiten von diesen als störend empfunden werden. Falls eine gewisse zeitliche Dauer überschritten wird, kann der Benutzer von seinem eigentlichen Ziel abgelenkt werden [Miller, 1968; Nielsen, 1994] und unter Umständen in der Annahme des Ausbleibens eines Ergebnisses die Webseite verlassen [Nielsen and Loranger, 2006].

Der Zeitraum, der benötigt wird, um ein Ergebnis zu bestimmen wird die *Response Time* RT genannt. Diese ergibt sich aus der Zeitspanne zwischen der Eingabe und dem Erhalt des Ergebnisses. Zur Bestimmung der *Response Time* eines *Recommender Systems*, wird der Mittelwert über die Zeiten $rt(u)$ gebildet, welche notwendig sind, um eine Vorhersage oder Empfehlung für einen Benutzer $u \in U$ des Systems zu erzeugen [George, 2005].

$$RT = \frac{\sum_{u \in U} rt(u)}{|U|} \quad (6.7)$$

Der Mittelwert über die *Response Time* berücksichtigt den Einfluss, den die unterschiedliche Größe von Profilen verschiedener Benutzer auf die Berechnungszeit aufweisen kann. Zudem wird der Einfluss weiterer Faktoren, wie beispielsweise zusätzliche auf der Maschine laufende Prozesse, minimiert werden. Neben der *Response Time* existiert eine weitere Metrik, um die Geschwindigkeit eines *Recommender Systems* zu bestimmen, der *Throughput* Tr . Dieser Wert gibt die Anzahl an Empfehlungen pro Sekunde an, die ein System generieren kann [Sarwar et al., 2001; Herlocker et al., 2002].

$$Tr = \text{Anzahl Empfehlungen/sec} \quad (6.8)$$

Beide Werte sind eng miteinander verbunden. So kann eine geringe *Response Time* zu einem hohen *Throughput* führen, der Umkehrschluss gilt jedoch nicht. Da der *Throughput* nicht

6.3. Evaluierungsmetriken

die Dauer einer einzelnen Berechnung berücksichtigt, können auch Systeme mit einem hohen *Throughput* eine lange *Response Time* aufweisen. Weiterhin kann der Durchsatz eines Systems durch die Verwendung von nebenläufig arbeitenden Systemen gesteigert werden, welche die Vorteile der aktuellen Multi-Prozessor-Computer besser ausnutzen. Auch dies hat keinen Einfluss auf die *Response Time*.

6.3.3 Coverage

Die *Coverage* bezeichnet den Grad der Elementabdeckung der von einem *Recommender* erzeugten Empfehlungen. Dabei bezeichnet die *Prediction Coverage* die Menge an Elementen, auf die ein *Recommender* überhaupt Vorhersagen und Empfehlungen generieren kann. Eine andere Form der *Coverage* ist die *Catalog Coverage*. Sie bezeichnet die Anzahl der Elemente, die den Benutzern, basierend auf ihrem subjektiven Nutzen, vorgeschlagen werden können. Allerdings ist dabei zu beachten, dass der Wert einer wirklich praktikablen *Catalog Coverage* nur schwer zu ermitteln ist [Herlocker et al., 2004]. So bedeutet es beispielsweise für einen einzelnen Benutzer keinen Nachteil, wenn ihm das System keine Elemente vorschlagen kann, welche nicht seinen Interessen entsprechen. Eine Abwesenheit solcher Elemente würde ein Benutzer nicht bemerken.

Ein System könnte eine möglichst hohe *Prediction Coverage* erzielen, indem es Empfehlungen mit einer größeren Anzahl an Elementen generiert. Dabei sinkt jedoch die Genauigkeit der gemachten Vorhersagen für den einzelnen Benutzer. Somit müssen sowohl die *Coverage* des Systems als auch die Genauigkeit der Empfehlungen gemeinsam betrachtet werden. Hierbei wird das von [Herlocker et al., 2004; Ge et al., 2010] beschriebene Verfahren zur Bestimmung der *Catalog Coverage*, basierend auf der Vereinigung aller generierter Empfehlungen für alle Benutzer des Systems, verwendet. Für N kann dann beispielsweise der Wert verwendet werden, bei dem ein System die höchste Genauigkeit der gemachten Empfehlungen bezüglich der *Precision* aufweist. Somit fließt auch die Genauigkeit des Verfahrens in gewissem Maße in die Bestimmung der *Coverage* ein. Jedoch sollte beachtet werden, dass praktikable Werte für N verwendet werden. Selbst wenn das System bei einer großen Empfehlungsmenge seine höchste Genauigkeit erzielt, bringt es für einen einzelnen Benutzer nur einen geringen Sinn, trotz alledem mehrere hundert Elemente zu betrachten. Die *Prediction Coverage* ist definiert durch:

$$Cover(N) = \frac{|\bigcup_{u \in U} RS(u, N)|}{|E|} \quad (6.9)$$

Auch wenn sich der *Recall* und die *Coverage* ähneln, beschreiben beide Metriken ein jeweils unterschiedliches Konzept. Während der *Recall* den Anteil der Elemente angibt, die ein *Recommender* aus der Menge für einen einzelnen Benutzer ermitteln kann, beschreibt die *Coverage* dies aus der Sicht des Systems. Sie gibt den Anteil der Elemente an, die der *Recommender* allen Benutzern überhaupt empfehlen kann und dient somit beispielsweise der Optimierung der Menge der im System enthaltenen Elemente.

6.4 Ermittlung der Umgebungsvariablen

Im Folgenden werden für die in Kapitel 5.2 vorgestellten Verfahren die Parameter ermittelt, welche für den Benutzer Vorhersagen mit der höchsten Genauigkeit bezüglich der in Kapitel 6.3.1 vorgestellten MAE -Metrik (siehe Gleichung 6.1) ermöglichen. Dazu wurde der Datensatz der Benutzer zufällig in eine Trainings- sowie in eine Testmenge aufgeteilt, wobei die Testmenge aus 250 zufällig aus dem Datensatz entnommenen Benutzern bestand. Die übrigen Benutzer bildeten die Trainingsmenge.

Die Bewertungen eines jeden Benutzers der Testmenge dienten der Erzeugung seines Profils. In einem Testlauf wurde für jeden einzelnen Benutzer zuerst das Element, für das die Vorhersage ermittelt werden sollte, aus seinem Profil entfernt. Dann wurde die Abweichung zwischen der Vorhersage $pr(u, e)$ und der tatsächlich abgegebenen Bewertung $r_{u,e}$ berechnet und anschließend das Element wieder im Profil abgelegt. Aus den errechneten und tatsächlichen Bewertungen aller Benutzer der Testmenge wurde dann die Genauigkeit anhand der MAE -Metrik für die zu testenden Parameter ermittelt. Die Parameter, bei denen geringste Wert für die MAE -Metrik beobachtet werden konnte, bildeten dann das Optimum für das jeweils untersuchte Verfahren und die Grundlage für die Evaluierung in Kapitel 6.

6.4.1 Collaborative Filtering

User to User Collaborative Filtering

Beim *User to User Collaborative Filtering* hat die Größe k der ermittelten Nachbarschaft des Benutzers einen großen Einfluss auf die Genauigkeit der für den Benutzer generierten Vorhersagen. Abhängig von der Verteilung der im System vorhandenen Bewertungen, muss dieser Wert im Vorfeld ermittelt werden. Dabei lässt sich für das in Kapitel 5.2.2 vorgestellte Verfahren aus der Literatur ein optimaler Wert für die Größe der Nachbarschaft von $k = 50$ entnehmen [Herlocker et al., 1999].

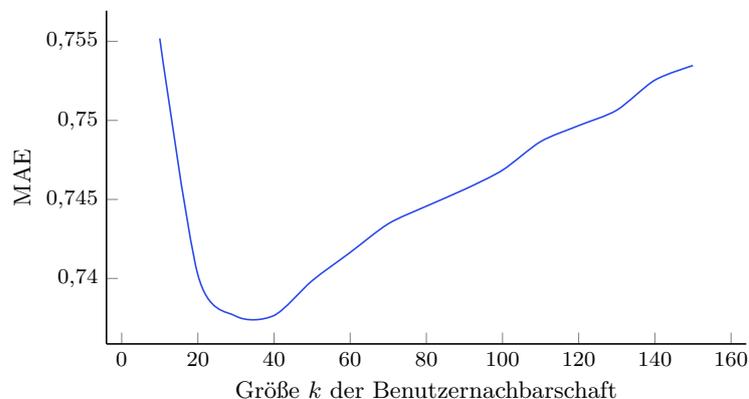


Abbildung 6.4 – Genauigkeit der Vorhersagen des User to User Collaborative Filterings in Abhängigkeit der Größe k der Benutzernachbarschaft

6.4. Ermittlung der Umgebungsvariablen

Abbildung 6.4 zeigt die Genauigkeit der generierten Vorhersagen der durchgeführten Evaluierung bezüglich der *MAE*-Metrik in Abhängigkeit von der Größe k der Nachbarschaft. Hierbei beschreibt die Nachbarschaft die zum aktuellen Benutzer u ähnlichsten Benutzer $n \in N(u, k)$. Als optimale Größe konnte dabei ein Wert von $k = 30$ ermittelt werden, welcher in der Größenordnung des aus der Literatur entnommenen Wertes liegt.

Sowohl kleinere als auch größere Werte für die Größe der Nachbarschaft führten zu einer Verschlechterung der Genauigkeit der generierten Vorhersagen. Bei der Verwendung einer zu kleinen Nachbarschaft liegen unter Umständen zu wenige oder gar keine Bewertungen auf das vorzuschlagende Element vor. Wurden die verwendeten Bewertungen von einer zu geringen Anzahl an Personen getätigt, so beeinflussen diese wenigen Bewertungen stark die Vorhersagen. Haben die Benutzer der Nachbarschaft das Element überhaupt noch nicht bewertet, kann keine Vorhersage ermittelt werden. Beides führt zu einer starken Abweichung der berechneten von der tatsächlichen Vorhersage.

Befinden sich andererseits zu viele Benutzer in der Nachbarschaft, werden bei der Berechnung auch Bewertungen von Benutzern einbezogen, deren Interessen nur geringfügig mit denen des aktuellen Benutzers übereinstimmen. Solche abweichenden Vorlieben verzerren ebenfalls das Ergebnis und verschlechtern die Genauigkeit der Vorhersagen.

Item to Item Collaborative Filtering

Auch bei dem in Kapitel 5.2.2 beschriebenen Algorithmus des *Item to Item Collaborative Filtering* hat die Größe k der Nachbarschaft einen Einfluss auf die Genauigkeit der Vorhersagen. Hierbei bezieht sich die Nachbarschaft jedoch nicht auf die ähnlichsten Benutzer, sondern auf die, zu denen vom Benutzer bereits bewerteten ähnlichsten Elemente. Zusätzlich wird bei der Ermittlung der Nachbarschaft die Bedingung an die Elemente gestellt, dass ihre Ähnlichkeit zum vorzuschlagenden Element einen Grenzwert t überschreitet. In der von [Sarwar et al., 2001] durchgeführten Evaluierung konnte für die Größe k der Nachbarschaft ein optimaler Wert von $k = 30$ ermittelt werden. Dabei wurde jedoch kein Grenzwert t berücksichtigt, es wurden immer die k ähnlichsten Elemente verwendet.

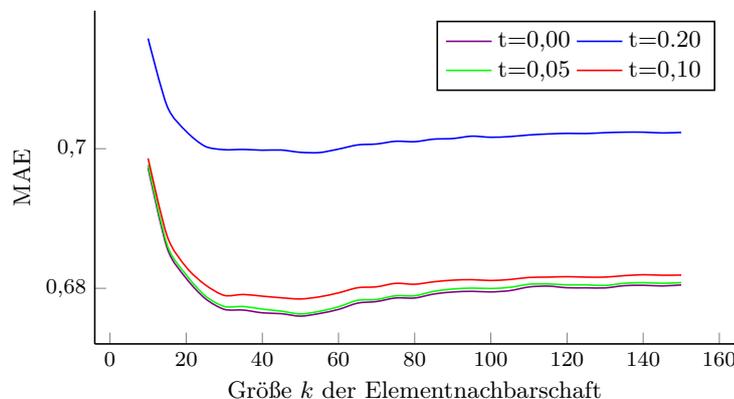


Abbildung 6.5 – Genauigkeit der Vorhersagen des *Item to Item Collaborative Filtering* in Abhängigkeit der Größe k der Elementnachbarschaft und dem Grenzwert t der Ähnlichkeit

Abbildung 6.5 zeigt die Genauigkeit der generierten Vorhersagen bezüglich der verwendeten *MAE*-Metrik in Abhängigkeit zur Größe k der Nachbarschaft. Weiterhin sind die Genauigkeiten für verschiedene Grenzwerte t dargestellt. Als optimaler Wert für die Größe der Nachbarschaft wurde der Wert $k = 30$ bestimmt, welcher damit in der Größenordnung des aus der Literatur entnommenen Wertes liegt.

Wird die Nachbarschaft zu klein gewählt, werden nur die wenigen Bewertungen der ähnlichsten Elemente berücksichtigt, die das Ergebnis möglicherweise stark in eine Richtung beeinflussen. Erst wenn eine ausreichende Menge an „unähnlicheren“ Elementen betrachtet wird, kann ein entsprechendes Abbild der Benutzerinteressen geschaffen werden. Im Gegensatz zum *User to User Collaborative Filtering* fällt die Genauigkeit der Vorhersagen jedoch nicht, wenn eine immer größer werdende Nachbarschaft an Elementen verwendet wird. Da die immer kleiner werdenden Ähnlichkeiten einzelner Elemente bei einer größer werdenden Nachbarschaft eine immer geringere Signifikanz für das Ergebnis aufweisen, wirken sich diese Elemente nur geringfügig auf den Wert der Vorhersage aus. Wenn die Größe k der Nachbarschaft unter Berücksichtigung der Genauigkeit eher klein gewählt wird, kann die Performanz des Systems aufgrund der geringeren Anzahl an Operationen gesteigert werden [Sarwar et al., 2001].

Die Verwendung eines Grenzwertes t für die Ähnlichkeit der in der Nachbarschaft enthaltenen Elemente verschlechterte die Genauigkeit der Vorhersagen. Da die meisten Elemente untereinander bezüglich des Ähnlichkeitsmaßes des Verfahrens nur einen sehr geringen Wert aufweisen, wird beim Einsatz eines Grenzwertes eine zu kleine Menge an Elementen in die Nachbarschaft aufgenommen. Somit steht keine ausreichende Anzahl für eine Generierung von Vorhersagen hoher Genauigkeit zur Verfügung.

6.4.2 Content-Based Recommender

Im verwendeten Verfahren des *Content-Based Recommenders* hat die Größe der für die Repräsentation der Elemente verwendeten Termvektoren einen Einfluss sowohl auf die Genauigkeit der generierten Vorhersagen, als auch auf die Performanz des Systems. Während die Verwendung einer höheren Anzahl an Einträgen in den Repräsentationen oftmals die Elemente genauer in ihren Eigenschaften beschreiben lässt, sinkt die Performanz, da eine größere Anzahl an Operationen für die Vergleiche zwischen den Repräsentationen und den Profilen durchgeführt werden muss. Eine Reduktion der Anzahl der verwendeten Terme kann daher eine Steigerung der Verarbeitungsgeschwindigkeit ermöglichen.

Es muss jedoch im Vorfeld entschieden werden, ob dabei die jeweils global häufigsten *Schlüsselworte* respektive *Synsets* betrachtet werden, oder die seltensten. Während bei der Verwendung der häufigsten Terme eine größere Anzahl an Übereinstimmungen zwischen den Profilen und den Elementrepräsentationen gefunden werden kann, weisen die selteneren Terme eine höhere Signifikanz für die Bedeutung eines Elementes auf und beschreiben es daher genauer. Im Folgenden wird die Größe der Termvektoren ermittelt, bei der das *Recommender System* die höchste Genauigkeit bezüglich der *MAE*-Metrik aufweist.

Abbildung 6.6 zeigt einen Ausschnitt aus der Verteilung der Termhäufigkeiten der jeweils 50 seltensten Worte respektive *Synsets*. Dabei wird die Anzahl der Terme in Abhängigkeit ihres

6.4. Ermittlung der Umgebungsvariablen

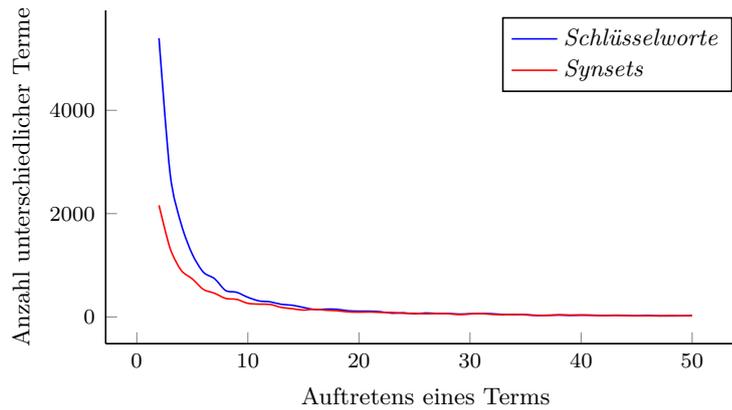


Abbildung 6.6 – Ausschnitt aus der Verteilung der Termhäufigkeiten

jeweiligen Auftretens, das heißt, in wie vielen Elementrepräsentationen sie jeweils enthalten sind, dargestellt. Die jeweilige Verteilung folgt erwartungsgemäß dem *Pareto-Prinzip* [Pareto, 1896], welches in der Natürlichen Sprache auftritt. Dieses beschreibt den Umstand, dass sehr viele unterschiedliche Terme global nur selten auftreten, einige wenige hingegen sehr oft.

In der Evaluierung der Systeme wurden nur Terme verwendet, die in mindestens zwei unterschiedlichen Elementen enthalten sind. Global einmalig auftretende Terme weisen keinerlei Nutzen bei der Ermittlung weiterer Elemente auf, sondern können sogar die Genauigkeit der Ergebnisse stark verschlechtern kann. Beinhaltet die Repräsentation eines Elementes ausschließlich diese Terme, kann niemals eine Übereinstimmung mit einem Benutzerprofil ermittelt werden, und es kann folglich auch nicht vorgeschlagen werden.

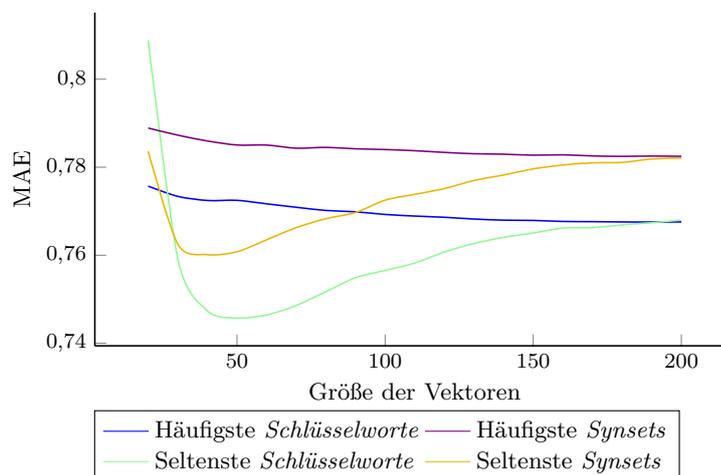


Abbildung 6.7 – Genauigkeit der Vorhersagen des Schlüsselwort- und des Synset-basierten Content-Based Recommenders unter Berücksichtigung der Größe der Elementvektoren

Abbildung 6.7 zeigt neben der Genauigkeit der errechneten Vorhersagen anhand der *MAE*-Metrik für die jeweils global häufigsten sowie seltensten Terme, auch die Genauigkeit des *Recommenders* bei der Verwendung von *Schlüsselworten* beziehungsweise *Synsets*. Dabei wird die jeweilige Genauigkeit in Abhängigkeit von der Größe der Termvektoren dargestellt. Sowohl bei

der Verwendung von *Schlüsselworten* als auch von *Synsets* kann jeweils ein identisches Verhalten des Verfahrens beobachtet werden. So werden bessere Ergebnisse durch den Einsatz der jeweils global seltensten Terme im Vergleich zu den häufigsten erzielt. Dies bestätigt die bereits erwähnte Annahme, dass die seltensten Terme die Elemente besser und genauer beschreiben.

Wird dabei jedoch eine zu geringe Anzahl verwendet, so fällt die Genauigkeit, da keine Übereinstimmungen zwischen den Profilen und den Repräsentationen gefunden werden können. Eine zu große Anzahl hingegen verschlechtert das Ergebnis, da sich dann die Repräsentationen aufgrund zu vieler gemeinsam enthaltener Einträge ähneln und somit der Vorteil der genaueren Beschreibung verloren geht. Dieses Verhalten kann aus der Darstellung aus den sich angleichenden Kurven entnommen werden. So konnte bei der Anwendung von *Schlüsselworten* in den Repräsentationen ein optimales Ergebnis bei einer Vektorgröße von 50 der seltensten Worte erzielt werden, während beim Einsatz von *Synsets* das Optimum bei der Verwendung der 40 seltensten *Synsets* lag.

Interessanterweise konnte die Verwendung von *Synsets* anstelle einfacher *Schlüsselworte* keine Verbesserung der Genauigkeit erzielen (siehe Abbildung 6.7). Sie verschlechterte diese sogar geringfügig. Dies lässt sich durch die Verteilung der Größen der Einträge in den einzelnen *Synsets* erklären. So bestanden 81 % der *Synsets* aus genau einem Eintrag (siehe Tabelle 6.2), wodurch sie vom System genau wie *Schlüsselworte* behandelt wurden.

Synset-Größe	Anzahl der Synsets
1	12728
2	2361
3	469
4	129
5	33
6	15
7	5
8	1
11	1

Tabelle 6.2 – Verteilung der Größen der *Synsets*

Dies ist darin zu begründen, dass der Korpus der zur Beschreibung der Elemente verwendeten Worte dieses Datensatzes verhältnismässig klein war und gleiche Eigenschaften von Benutzern nahezu immer identisch ausgezeichnet wurden. So stellten die *Schlüsselworte* in diesem Fall bereits eindeutig interpretierbare Eigenschaften dar.

6.4.3 Taxonomic Recommender

Das verwendete Verfahren des *Taxonomic Recommenders* basiert, analog zum *User to User Collaborative Filtering*, auf der Ermittlung der Nachbarschaft der zum aktuellen Benutzer ähnlichsten anderen Benutzer. Somit muss auch hier die optimale Größe k für die Nachbarschaft bestimmt werden. Neben dieser hat auch der Vererbungs-faktors κ eine Auswirkung auf die Genauigkeit. Aus diesem Grund muss der Wert an die Höhe sowie die Verteilung der inneren Knoten und Blätter der verwendeten Taxonomie angepasst werden.

6.4. Ermittlung der Umgebungsvariablen

Im Folgenden wird das Verfahren für zwei unterschiedliche Taxonomien untersucht. Anhand der aus der *lexikalischen* Datenbank entnommenen Hyperonym-Hierarchie der für die *Schlüsselworte* bestimmten *Synsets*, wurde die Taxonomie des *Synset*-basierten *Taxonomic Recommenders* gebildet. Die zweite Taxonomie ist durch die Hierarchie der Kategorien der die Elemente beschreibenden Wikipedia-Artikel gegeben. Diese Hierarchie wird durch entsprechende *Properties* in der Datenbank des dbpedia-Projektes im *RDFS*-Format beschrieben und bildet die Grundlage für den *RDFS*-basierten *Taxonomic Recommender*.

In der von [Ziegler et al., 2004b] durchgeführten Evaluierung des Verfahrens wurde für die dort verwendete Taxonomie sowie die Benutzerdaten eine Nachbarschaftsgröße von $k = 20$ und ein Vererbungsfaktor von $\kappa = 0,75$ verwendet. Diese Werte waren willkürlich gewählt, ohne auf deren Auswirkung im Genaueren einzugehen.

Abbildung 6.8 zeigt die Genauigkeit des auf der Verwendung von *Synsets* basierenden *Taxonomic Recommenders* in Abhängigkeit der Größe k der Nachbarschaft. Dabei kann bei diesem Verfahren ein ähnliches Verhalten wie beim *User to User Collaborative Filtering* beobachtet werden (siehe Kapitel 6.4.1), was sich dadurch erklärt, dass beide auf der Ermittlung einer Nachbarschaft der zum aktuellen Benutzer ähnlichsten anderen Benutzer basieren. Im Gegensatz zum *User to User Collaborative Filtering* wird dabei die Ähnlichkeit jedoch nicht anhand der einfachen Bewertungsvektoren der Benutzer ermittelt, sondern basiert auf den aus den bewerteten Elementen entnommenen Eigenschaften sowie der Taxonomie, in der diese hierarchisiert sind. Somit können auch ähnliche Benutzer ermittelt werden, selbst wenn keine gemeinsam bewerteten Elemente vorliegen.

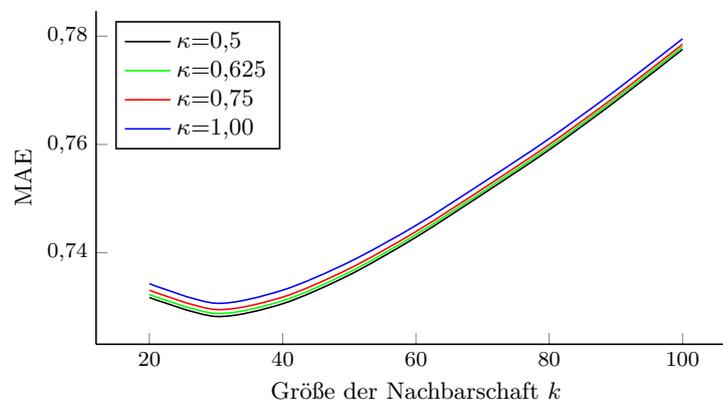


Abbildung 6.8 – Genauigkeit der Vorhersagen des *Synset*-basierten *Taxonomic Recommenders* anhand der MAE-Metrik in Abhängigkeit der Nachbarschaftsgröße k und des Vererbungsfaktors κ

Wird bei der Generierung der Vorhersagen eine zu geringe oder eine zu große Anzahl ähnlicher Benutzer betrachtet, so sinkt jeweils die Genauigkeit der Empfehlungen. Während bei einer zu kleinen Menge an Benutzern, einige wenige Bewertungen einen zu starken Einfluss auf die errechnete Vorhersage aufweisen, verschlechtert sich das Ergebnis aufgrund der Berücksichtigung von Benutzern mit teils abweichenden Interessen. Da bei den *Taxonomic Recommendern* bei einem zu großem k auch Benutzer in die Nachbarschaft aufgenommen werden, die nur über weit entfernte Oberkonzepte mit dem aktuellen Benutzer verbunden sind, zeigen sie sich aufgrund

der zu starken Generalisierung sensibler gegenüber einer zu großen Nachbarschaft. Dies erklärt den im Vergleich zum *User to User Collaborative Filtering* stärkeren Anstieg der Kurve.

Bei der hier verwendeten Taxonomie wurde ein optimaler Wert für die Größe der Nachbarschaft von $k = 30$ und optimaler Wert für den Vererbungsfaktor von $\kappa = 0,5$ ermittelt. Wurden größere Werte verwendet, so verschlechterte sich die Genauigkeit. In diesem Fall erhalten Oberkonzepte eine zu hohe Gewichtung, was auf Kosten der exakten Beschreibung der Benutzerinteressen geschieht.

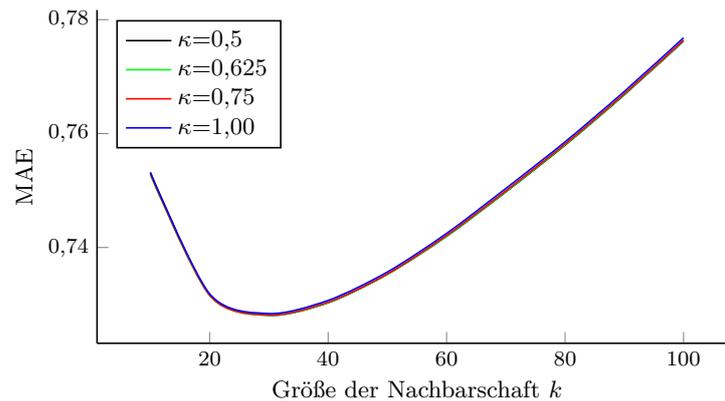


Abbildung 6.9 – Genauigkeit der Vorhersagen des RDF-basierten Taxonomic Recommenders anhand der MAE-Metrik in Abhängigkeit zur Nachbarschaftsgröße k und des Vererbungsfaktors κ

In Abbildung 6.9 ist die Genauigkeit des Verfahrens bei der Verwendung der in *RDFS* modellierten Taxonomie dargestellt. Dabei konnte ein ähnliches Verhalten wie beim Einsatz der anhand von *Synsets* gebildeten Taxonomie beobachtet werden. Hier wurde ein Wert für die optimale Größe der Nachbarschaft von $k = 30$ ermittelt. Interessanterweise hat die Verwendung unterschiedlicher Werte für den Vererbungsfaktor κ nur eine geringe, kaum messbare, Auswirkung auf die Genauigkeit der Vorhersagen, was in der geringen Höhe der in *RDFS* modellierten Taxonomie begründet liegt. Dabei liegt die Abweichung zwischen den einzelnen Kurven im Bereich von Rundungsfehlern, auch wenn eine Tendenz erkennbar ist. Trotzdem wurde ein optimaler Wert für $\kappa = 0,625$ ermittelt.

6.4.4 Semantic Recommender

In dem verwendeten Verfahren des *Semantic Recommenders* ist die Anzahl der Iterationen it das Kriterium, welches die Genauigkeit des Verfahrens beeinflusst. Dabei gibt der Wert für it die Reichweite der *Property-Propagation* an, das heißt die Entfernung im durch die Ontologie gebildeten Graphen, in der die enthaltene Konzepte bei der Bestimmung der Ähnlichkeit zweier Konzepte noch einen Einfluss aufweisen. Dieser Wert ist, ähnlich wie der in den *Taxonomic Recommendern* vorhandene Vererbungsfaktor κ , in großem Maße abhängig von der Struktur der verwendeten *Wissensbasis*.

In der Evaluierung von [Lémdani et al., 2011] wurde jedoch nur auf die Auswirkung der Anzahl der Iterationen auf die Ähnlichkeit der Elemente des Systems eingegangen und nicht

6.5. Evaluierung der Verfahren

ihr Einfluss auf die Genauigkeit der generierten Vorhersagen ermittelt. Somit lässt sich kein praktikabler Vergleichswert für it aus der Literatur entnehmen.

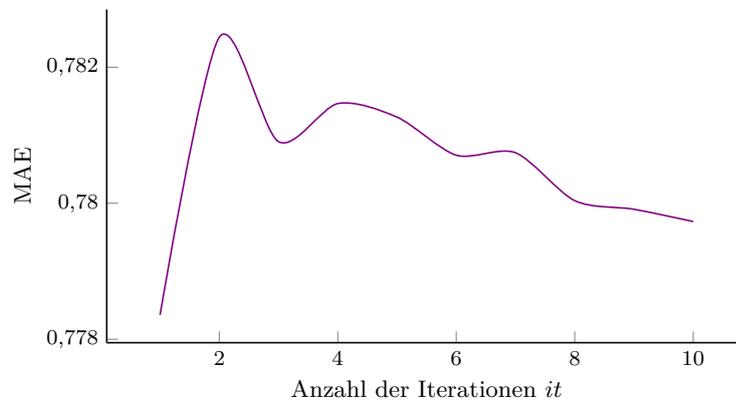


Abbildung 6.10 – Genauigkeit der Vorhersagen des *Semantic Recommenders* in Abhängigkeit zur Anzahl it der Iterationen

Abbildung 6.10 zeigt die Genauigkeit der Vorhersagen des *Semantic Recommenders* in Abhängigkeit zur Anzahl der durchgeführten Iterationen it . Dabei wurde eine optimale Anzahl von $it = 1$ ermittelt. Die Bestimmung dieses Wertes lag in der Struktur der verwendeten Ontologie begründet (siehe Abbildung 6.2). So sind alle vorzuschlagenden Elemente vom gleichen Typ und alle ihre beschreibenden Eigenschaften haben den Abstand eins von ihnen. Somit befinden sich alle Elemente im Abstand zwei zueinander. Entsprechend nun it mindestens ihrem Abstand, überlagerten sich die Daten der einzelnen Elemente, so dass bei der Ermittlung der Ähnlichkeiten Informationen verwendet wurden, die dafür keinen Nutzen aufwiesen oder sogar kontraproduktiv waren.

6.5 Evaluierung der Verfahren

Nach Bestimmung der optimalen Parameter, bei denen die einzelnen Algorithmen die genauesten Vorhersagen bezüglich der *MAE*-Metrik genießen, können diese anhand weiterer, in Kapitel 6.3 vorgestellter Kriterien unter gleichen Bedingungen miteinander verglichen werden. Die jeweils verwendeten Parameter der einzelnen Verfahren sind in Tabelle 6.3 aufgeführt. Dabei ist der *Mean-Based Recommender* als Baseline-Verfahren nicht in der Tabelle enthalten, da dieser keinerlei Parameter verwendet, die seine Arbeitsweise beeinflussen.

Im Folgenden werden die in Tabelle 6.3 aufgeführten Verfahren sowie der *Mean-Based Recommender* bezüglich der Genauigkeit ihrer Vorhersagen sowie ihren Empfehlungen analysiert. Dabei wird zudem betrachtet, inwieweit die Verfahren die Elemente fehlerhaft klassifizieren, da diese Fehler insbesondere im *E-Commerce* von großer Bedeutung sind (siehe Kapitel 6.3.1). Ferner werden auch weitere, im Betrieb eines *Recommenders* wichtige, Kriterien des Lernverhalten der Benutzerprofile beziehungsweise des Systems, der Skalierbarkeit sowie der Coverage der Empfehlungen für jedes Verfahren analysiert.

Verfahren	Parameter	Wert
User to User Collaborative Filtering	Größe der Benutzernachbarschaft (k)	30
Item to Item Collaborative Filtering	Größe der Elementnachbarschaft (k) Grenzwert der Elementähnlichkeit (t)	50 0,0
Content-Based Recommender (Tags)	Größe der Repräsentationsvektoren Seltenste <i>Schlüsselworte</i>	50
Content-Based Recommender (<i>Synsets</i>)	Größe der Repräsentationsvektoren Seltenste <i>Synsets</i>	40
Taxonomic Recommender (<i>Synsets</i>)	Größe der Benutzer Nachbarschaft (k) Vererbungsfaktor (κ)	30 0,5
Taxonomic Recommender (RDFS)	Größe Benutzer Nachbarschaft (k) Vererbungsfaktor (κ)	30 0,625
Semantic Recommender	Anzahl der Iterationen (it)	1

Tabelle 6.3 – Parameter der Verfahren für ihre jeweils höchste Genauigkeit der Vorhersagen

6.5.1 Genauigkeit

Die Genauigkeit der ermittelten Vorhersagen und Empfehlungen eines *Recommender Systems* stellt sowohl für die Benutzer als auch für die Betreiber das wichtigste Kriterium für die Güte eines Verfahrens dar. Werden für die Benutzer zu viele fehlerhafte Empfehlungen ermittelt, so führt dies zu einer Unzufriedenheit und sie verlassen endgültig die Webseite, da sie das Gesuchte nicht finden konnten oder ihnen gänzlich falsche Elemente vorgeschlagen wurden (siehe Kapitel 6.3.1).

Für die Evaluierung der Genauigkeit wurden zunächst 250 Benutzer zufällig aus der Menge der im Datensatz enthaltenen Benutzer gezogen, welche die Testmenge darstellten. Die restlichen Benutzer bildeten die Trainingsmenge, mit welcher das Verfahren initialisiert wurde. Dabei waren die beiden Mengen für die verschiedenen Verfahren identisch, um jene auch unter identischen Bedingungen vergleichen zu können. Für jeden Benutzer der Testmenge wurde dann anhand der in Kapitel 6.3.1 beschriebenen Metriken die Genauigkeit der für ihn ermittelten Vorhersagen und Empfehlungen gemessen. Der Mittelwert aller ermittelten Werte bildete dann das Ergebnis der jeweiligen Metrik und wurde in den entsprechenden Diagrammen dargestellt.

Durchschnittlicher Absoluter Fehler

Abbildung 6.11 zeigt eine Zusammenfassung der in Kapitel 6.4 ermittelten Resultate für die *MAE*-Metrik (siehe Gleichung 6.1). Dabei wurden für die einzelnen Verfahren die Parameter verwendet, bei denen in der zuvor durchgeführten Evaluierung (siehe Kapitel 6.4) die höchste Genauigkeit bezüglich dieser erzielt wurde. Dies erlaubt einen Vergleich der verschiedenen Algorithmen unter jeweils gleichen Bedingungen.

In der Evaluierung erzielte das *Item to Item Collaborative Filtering* die höchste Genauigkeit für seine Vorhersagen. Auch das *User to User Collaborative Filtering* konnte Vorhersagen guter Genauigkeit generieren, erreichte aber nicht die Präzision des erstgenannten Verfahrens. Trotz alledem überstieg seine Genauigkeit die der *Content-Based Recommender* und des *Semantic Re-*

6.5. Evaluierung der Verfahren

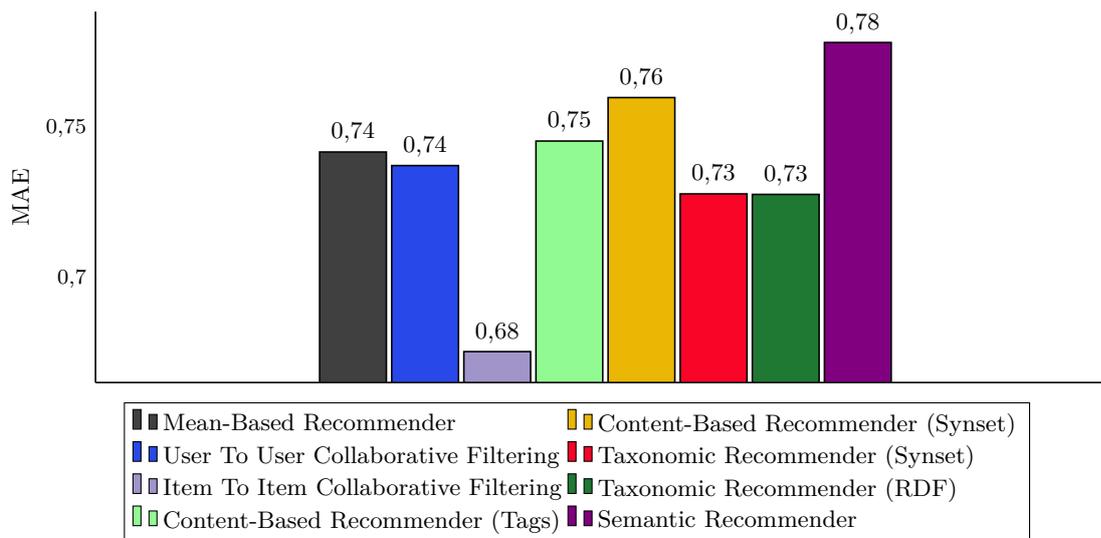


Abbildung 6.11 – Genauigkeit der verschiedenen Algorithmen nach der MAE-Metrik

commenders. Selbst der *Mean-Based Recommender* konnte genauere Ergebnisse als diese beiden Verfahren erzielen, obwohl er seine Vorhersagen nicht anhand der Vorlieben eines Benutzers erzeugt. Da die *semantisch* einfachen Verfahren ihre Vorhersagen ausschließlich anhand der von den Benutzern abgegebenen Bewertungen ermitteln, sind sie folglich unabhängig von der Qualität der Elementrepräsentationen, welche daher einen starken Einfluss auf die Genauigkeit eines *Content-Based Recommenders* und in noch größerem Maße auf den *Semantic Recommender* hat. Auch wenn ebenfalls in den verwendeten *Taxonomic Recommender* die Repräsentation der Elemente Auswirkungen auf die Genauigkeit hat, minimierte jedoch ihre *Hybridisierung* mit dem *Collaborative Filtering* diese Probleme. So wurde mit diesem Verfahren ein zum im Vergleich zu den *Content-Based Recommendern* erhöhte Genauigkeit .

Fehlerhafte Vorhersagen

Auch wenn durch den *MAE* die Genauigkeit eines Verfahrens über alle einzelnen Werte der durch einen *Recommender* verwendeten Bewertungsskala genau ermittelt werden kann, ist es für die Benutzer wichtiger, dass „relevante“ beziehungsweise „nicht relevante“ Elemente als solche klassifiziert werden. Der genaue Wert ist für sie von untergeordnetem Stellenwert. Insbesondere für die Betreiber eines *Online-Shops* ist entscheidend, dass eine möglichst geringe Anzahl von Produkten fehlerhaft klassifiziert wird.

Eine quantitative Ermittlung dieser Form von Fehlern bietet die Klassifikation der Vorhersagen in *False Positives* und *False Negatives* an (siehe Kapitel 6.3.1). Irrtümlich als „relevant“ identifizierte Elemente bilden die Menge der *False Positives*, und die trotz vorliegendem Interesse des Benutzers für „nicht relevant“ befundenen Elemente die der *False Negatives*. Da die Bewertungen des verwendeten Datensatzes in einer Bewertungsskala mit diskreten Werten zwischen eins und fünf vorlagen, mussten diese zur Ermittlung der *False Positives* und *False Negatives* zunächst in ein binäres Format umgewandelt werden (siehe Kapitel 6.3.1). Für die Evaluierung

wurden daher die mit den Werten vier und fünf bewerteten Elemente auf „relevant“ und die übrigen auf „nicht relevant“ abgebildet.

Für die Bestimmung der *False Positives* und der *False Negatives* wurden für jeden Benutzer der Testmenge die Vorhersagen auf seine bewerteten Elemente bestimmt. Dadurch konnte, ausgehend von der zuvor durchgeführten Einteilung der Bewertungen in „relevant“ und „nicht relevant“, ermittelt werden, inwieweit ein Verfahren die Elemente korrekt für einen Benutzer klassifiziert.

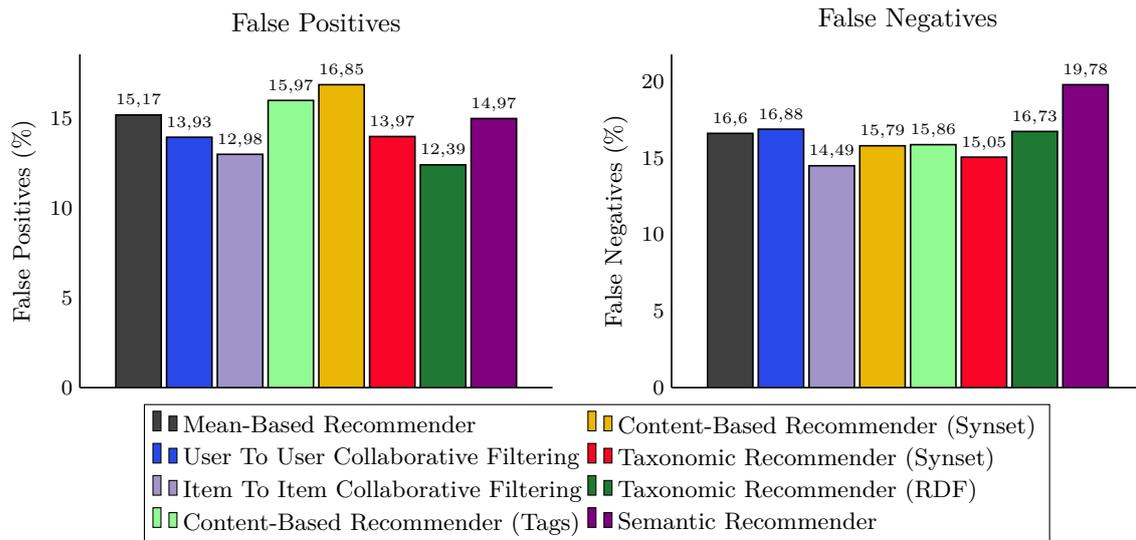


Abbildung 6.12 – Anzahl der *False Positives* und *False Negatives* bezogen auf die Vorhersagen

In Abbildung 6.12 wird jeweils der prozentuale Anteil der falsch klassifizierten Elemente in Form der *False Positives* beziehungsweise der *False Negatives* dargestellt. Dies ermöglicht den Vergleich der Verfahren im Hinblick auf eine für den Benutzer subjektive Genauigkeit.

Die hohe erzielte Genauigkeit bezüglich der *MAE*-Metrik manifestierte sich beim *Item to Item Collaborative Filtering* auch in der Präzision bei der Klassifikation der Elemente. Dabei erreichte es die besten Werte der untersuchten Verfahren für die Anzahl *False Negatives*, so dass es verlässlich die Elemente von Relevanz auch als solche erkannte. Ferner erzielte das Verfahren auch eine hohe Genauigkeit im Hinblick auf die *False Positives*.

Auch das *User to User Collaborative Filtering* klassifizierte „nicht relevante“ Elemente sicher als solche, auch wenn es nicht die Genauigkeit des *Item to Item Collaborative Filterings* erzielte. Jedoch wies es eine im Vergleich zu den anderen Verfahren hohe Anzahl an *False Negatives* auf. Aufgrund der Abhängigkeit der Vorhersagen des *User to User Collaborative Filterings* zu den Vorlieben der ermittelten Nachbarschaft, ist es dem Verfahren nicht möglich, „relevante“ Elemente zu erkennen, wenn die subjektiven Interessen des Benutzers in zu starkem Maße von seiner Nachbarschaft abweichen.

Beide *Content-Based Recommender* wiesen Probleme dabei auf, „nicht relevante“ Elemente zuverlässig als solche zu klassifizieren, was sich in der hohen Anzahl an *False Positives* zeigt. Aufgrund der durch andere Elemente ins Profil eines Benutzers aufgenommenen Terme schätzten sie ein Element auch dann als „relevant“ ein, wenn die sonstigen Terme des Elementes im Wi-

6.5. Evaluierung der Verfahren

derspruch zu den Vorlieben stehen. Dies ermöglicht allerdings eine zuverlässige Erkennung von Elementen, welche den bereits als „relevant“ bewerteten Elementen ähneln, so dass die Anzahl der *False Negatives* entsprechend klein ausfällt. Trotz alledem konnten sie nicht das Ergebnis des *Item to Item Collaborative Filterings* erzielen.

Im Gegensatz zu den nahezu identischen Ergebnissen bezüglich der anderen Metriken zeigten die *Taxonomic Recommender* bei der Klassifikation der Ergebnisse ein divergentes Verhalten, was durch Probleme ihrer jeweiligen verwendeten Elementrepräsentationen bedingt ist. Da für 10 % der *Schlüsselworte* fehlerhafte *Synsets* ermittelt wurden, führte dieser Umstand zur falschen Klassifikation der sie beinhaltenden Elemente durch den *Synset*-basierten *Taxonomic Recommender* und somit zu seiner höheren Zahl an *False Positives*. Durch das Fehlen der Kategorien für 9 % der Elemente konnte der *RDFS*-basierte *Taxonomic Recommender* diese nicht erfassen, was zu seiner geringeren Präzision in Bezug auf die *False Negatives* führte.

Der verwendete *Semantic Recommender* zeigte neben den beiden *Content-Based Recommendern* die höchste Anzahl an *False Positives*. Bedingt durch seine vergleichsweise hohe Ungenauigkeit (siehe Abbildung 6.11) war es im nicht möglich, die Elemente korrekt zu klassifizieren. Dieser Sachverhalt wirkte sich ebenfalls auf die Anzahl der *False Negatives* auf, so dass das Verfahren des *Semantic Recommenders* auch hierbei die schlechtesten Resultate aller getesteten Verfahren aufwies.

Precision / Recall

Um die *Precision* sowie den *Recall* (siehe Gleichungen 6.5 und 6.6) zu ermitteln, wurden die von einem Benutzer der Testmenge bewerteten Elemente anhand der zuvor beschriebenen Methode in eine Menge „relevanter“ beziehungsweise „nicht relevanter“ Elemente aufgeteilt. Ferner wurden fünf disjunkte Teilmengen aus der Menge der „relevant“ bewerteten Elemente eines Benutzers gebildet (sogenanntes *5-Fold Sampling*). Anhand jeweils vier dieser Teilmengen sowie den „nicht relevanten“ Elementen wurde das Benutzerprofil generiert, welches die Trainingsmenge darstellte. Die nicht enthaltene Teilmenge der „relevanten“ Elemente bildete dann die Testmenge eines einzelnen Testlaufs. Der Mittelwert aller Testläufe ergab schließlich anhand der in Kapitel 6.3.1 vorgestellten Metriken das Ergebnis für die *Precision* respektive den *Recall* des jeweiligen Verfahrens.

Abbildung 6.13 zeigt die für die Verfahren ermittelten Werte der *Precision* und des *Recalls* in Abhängigkeit zur Größe der Empfehlungsmenge. Während die Kurve der *Precision* erwartungsgemäß ein Maximum für eine bestimmte Größe aufweist, erhöht sich der *Recall* der Verfahren proportional zur Anzahl der Elemente. Da sich beide Werte gegenseitig invers beeinflussen und somit niemals gleichzeitig ihren Maximalwert erreichen, muss im Vorfeld abgewogen werden, ob für die verwendete Domäne eher die *Precision* oder der *Recall* von Bedeutung ist.

Die geringen Werte der *Precision* im Vergleich zum *Recall* lassen sich anhand der Definition beider Metriken begründen. Während der *Recall* das Verhältnis der ermittelten Elemente zur Größe einer bekannten Testmenge darstellt und damit eindeutig ermittelbar ist, basiert die *Precision* zusätzlich auf der Annahme, dass alle unbewerteten Elemente für einen Benutzer nicht relevant sind, obwohl diese durchaus von Interesse sein könnten. Da die Benutzer normalerweise

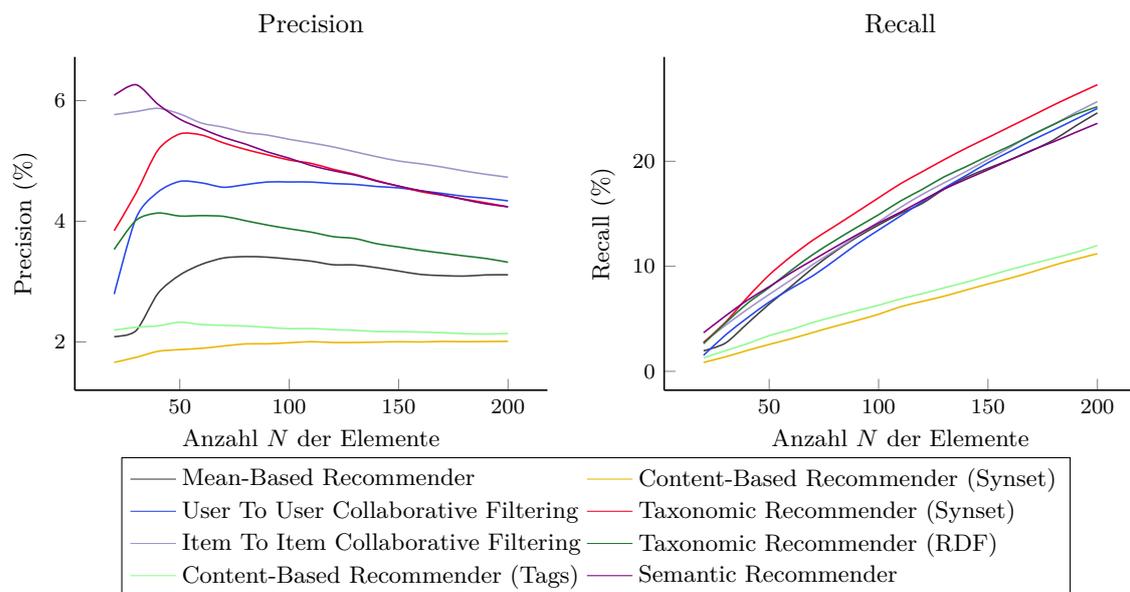


Abbildung 6.13 – Precision und Recall der verschiedenen Algorithmen für verschiedene Größen N der Empfehlungen

nur einen sehr kleinen Anteil der Elemente des Systems bewertet haben und somit nur wenige Vergleichswerte in den Evaluierungsdaten enthalten sind, fällt der Wert entsprechend gering aus. Nur wenn eine Klassifikation für einen Großteil der Elemente vorhanden ist, kann die *Precision* entsprechend genau ermittelt werden. Die geringe Zahl der Bewertungen der Benutzer hat auch weitere Auswirkungen auf die *Precision*.

Trotz alledem ermöglichen die Metriken ohne Berücksichtigung der absoluten Werte eine grobe Einschätzung der Genauigkeit der generierten Empfehlungen. Auf diese hat neben der Präzision bezüglich der *MAE*-Metrik auch die Anzahl der fehlerhaft klassifizierten Elemente Auswirkungen.

Um so überraschender ist die hohe Genauigkeit des *Semantic Recommenders* für seine generierten Empfehlungen und die niedrige des *RDFS*-basierten *Taxonomic Recommenders*. Während das Ergebnis des letzteren Verfahrens auf das Fehlen der Kategorien im *RDFS*-Format für 9 % der Elemente zurückzuführen ist, ist die hohe Genauigkeit des *Semantic Recommenders* mit den zuvor erfassten Daten nicht erklärbar und bedarf einer detaillierteren Untersuchung.

Die übrigen Verfahren zeigten hingegen bei der Messung der *Precision* über alle Größen der Empfehlungsmenge das gleiche Verhalten wie für die *MAE*-Metrik. So wies das *Item to Item Collaborative Filtering* die höchste *Precision* auf, während die *Content-Based Recommender* jeweils den geringsten Wert erzielten.

Der ermittelte *Recall* wies für alle Verfahren die erwartete Proportionalität zur Größe der Empfehlungsmenge auf. Dabei unterschieden sich die Ergebnisse der einzelnen Verfahren nur geringfügig. Lediglich beide *Content-Based Recommender* zeigten ein abweichendes Verhalten, was auf ihre im Vergleich zu den übrigen Verfahren hohen Anzahl an *False Positives* zurückzuführen ist. Aufgrund der vielen fälschlicherweise als „relevant“ klassifizierten Elemente, werden dadurch korrekt bewertete Elemente nicht in die Empfehlungsmenge aufgenommen.

6.5. Evaluierung der Verfahren

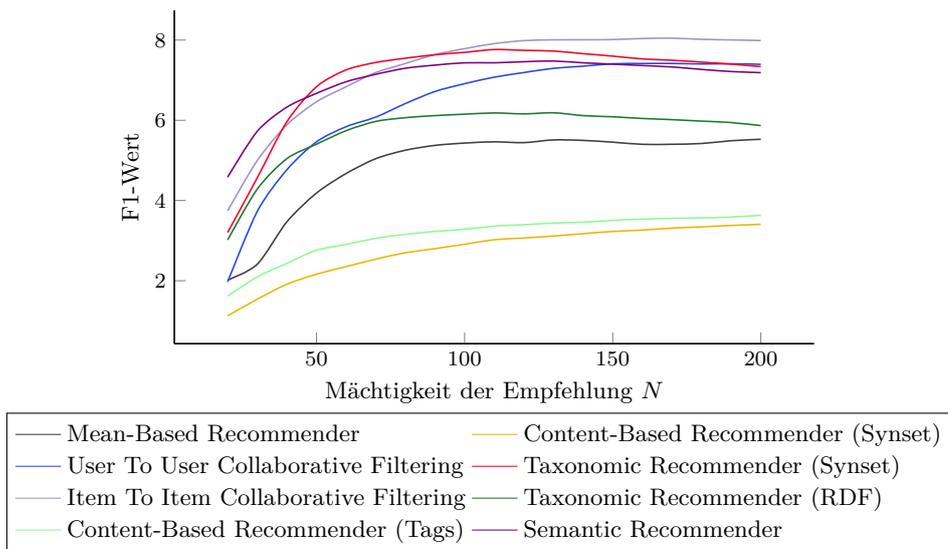


Abbildung 6.14 – Genauigkeit der verschiedenen Algorithmen nach der F1-Metrik

Abbildung 6.14 zeigt die Qualität der Empfehlungen anhand der $F1$ -Metrik (siehe Gleichung 6.4). Diese bildet die *Precision* und den *Recall* mit identischer Gewichtung auf einen einzelnen Wert ab, wodurch beide Werte für eine beliebige Größe der Empfehlungsmenge in Kombination betrachtet werden können.

In der durchgeführten Messung waren die Werte jedoch von einer unterschiedlichen Größenordnung (siehe Abbildung 6.13). Während für die *Precision* ein Maximum von 6,5 % ermittelt wurde, erreichte der *Recall* der Verfahren den fünffachen Wert von fast 30 %. Somit hing der Wert der $F1$ -Metrik bei einer größeren Anzahl an Elementen nahezu ausschließlich vom *Recall* ab. Zwar hatte die *Precision* für kleinere Empfehlungsmengen einen stärkeren Einfluß, jedoch war selbst dann der Wert des *Recalls* der entscheidendere Faktor für das Ergebnis der $F1$ -Metrik.

Daher ist keine objektive Betrachtung des Zusammenhangs zwischen der *Precision* und dem *Recall* möglich. Für eine Evaluierung der Empfehlungen insbesondere für praktikable Größen ist die *Precision* die geeignetere Metrik, da es für die Benutzer entscheidend ist, passende Elemente empfohlen zu bekommen.

6.5.2 Lernverhalten der Benutzerprofile

Mit einer Evaluierung bezüglich des Lernverhaltens der Benutzerprofile kann ermittelt werden, ob ein *Recommender* mit steigender Anzahl an Bewertungen eines Benutzer dessen Interessen und Vorlieben immer besser erfasst. Dabei ist die Lernkurve jedoch stark abhängig von den verwendeten Daten und lässt sich nicht ohne weiteres auf beliebige andere Domänen übertragen. Dennoch ermöglicht sie es, eine gewisse Tendenz für die jeweiligen Verfahren zu identifizieren. Aus diesem Grund werden im Folgenden nicht absolute Werte verglichen, sondern die Tendenzen der einzelnen Verfahren beschrieben.

In der durchgeführten Evaluierung wurden für die Generierung der Testmenge 250 Benutzer zufällig aus dem Datensatz gezogen. An diese wurde das Kriterium gestellt, dass jeder einzelne mindestens 200 Elemente bewertet haben musste. Ferner wurden die im Datensatz enthaltenen

Zeitstempel der Bewertungen verwendet, um das Bewertungsverhalten eines einzelnen Benutzers chronologisch und reproduzierbar zu simulieren.

Für die Benutzer der Testmenge wurden ihre 200 ältesten Bewertungen in eine subjektive Trainings- sowie in eine Testmenge aufgeteilt, wobei die Trainingsmenge aus den 100 ältesten und die Testmenge aus den restlichen 100 Bewertungen des jeweiligen Benutzers gebildet wurde. Um die Lernkurve zu ermitteln, wurde das Profil eines Benutzers nach und nach mit den Bewertungen seiner Trainingsmenge in ihrer chronologischer Reihenfolge gefüllt. Für jede Anzahl der im Profil enthaltenen Bewertungen wurde dabei die Genauigkeit der Vorhersagen für die Elemente der Testmenge anhand der *MAE*-Metrik ermittelt. Diese Kurve ist für jedes evaluierte Verfahren in Abbildung 6.15 dargestellt.

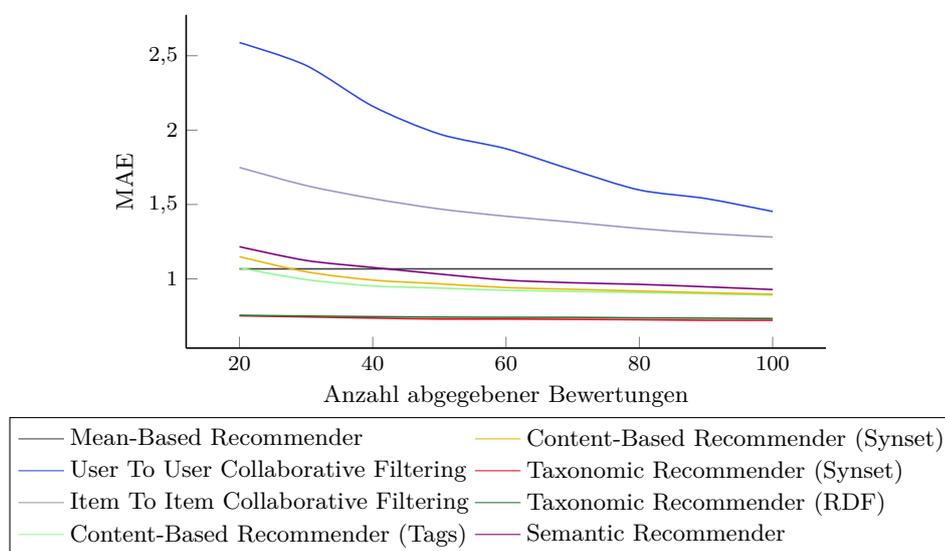


Abbildung 6.15 – Genauigkeit der Algorithmen anhand der *MAE*-Metrik in Abhängigkeit zu der Anzahl der Bewertungen (Lernkurve)

Während die meisten der vorgestellten *Recommender* auf einem lernenden Benutzerprofil basieren, generiert der *Mean-Based Recommender* unabhängig davon und ohne Berücksichtigung der Bewertungen eines einzelnen Benutzers seine Vorhersagen mit der stets für ihn optimalen Genauigkeit. Die übrigen Verfahren benötigen eine gewisse Anzahl an Bewertungen eines Benutzers, um seine Vorlieben hinreichend genau zu interpretieren und somit die Elemente entsprechend dieser subjektiven Vorlieben auch korrekt zu klassifizieren.

In der Evaluierung benötigte das *User to User Collaborative Filtering* die größte Anzahl an Bewertungen eines Benutzers, um seine Interessen entsprechend zu erfassen. Dies ist in der Notwendigkeit, Überlappungen zwischen seinen Bewertungen und denen der anderen Benutzer zu finden, begründet, die erst ab einer gewissen Anzahl abgegebener Bewertungen eines Benutzers ausreichend und passend zu ermitteln sind. Liegen zu wenige vor, können die dabei ermittelten ähnlichen anderen Benutzer auch abweichende Interessen aufweisen, wodurch die Genauigkeit des Systems sinkt.

Ein ähnliches Verhalten kann beim *Item to Item Collaborative Filtering* beobachtet werden. Auch dieses Verfahren verlangt vom Benutzer, eine große Anzahl an Elementen bewertet zu

6.5. Evaluierung der Verfahren

haben, ehe es seine Interessen entsprechend genau abbilden kann. Im Gegensatz zum *User to User Collaborative Filtering* basieren dabei die Vorhersagen nicht auf Überlappungen zwischen den Bewertungsvektoren sondern auf der Bestimmung der Ähnlichkeit zu den bereits bewerteten Elementen. Diese müssen im Vorfeld von anderen Benutzern in Kombination mit dem zu empfehlenden Element bewertet worden sein. Hat ein Benutzer noch nicht genügend Elemente bewertet, so kann keine ausreichende Anzahl sogenannter Co-Ratings zu den bereits bewerteten Elementen ermittelt werden, wodurch die Genauigkeit aufgrund zu weniger Informationen entsprechend sinkt.

Diese Problematik wiesen die beiden *Content-Based Recommender* sowie der *Semantic Recommender* nicht auf, so dass diese ein besseres Lernverhalten der Benutzerprofile zeigten. Dabei benötigte der *Semantic Recommender* weniger Bewertungen als die beiden *Collaborative Filtering* verwendenden Verfahren, um sein Optimum zu erreichen. Es ist jedoch zu beachten, dass der *Semantic Recommender* nicht die Genauigkeit aller übrigen vorgestellten Algorithmen erreichte.

Bereits wenige, in das Profil aufgenommene Terme ermöglichten es beiden *Content-Based Recommendern*, Elemente zu ermitteln, welche für den Benutzer dem Verfahren nach von größtem subjektiven Nutzen waren. Die Lernkurve flacht nach einer bestimmten Anzahl an Bewertungen allerdings ab, da die neu hinzugekommenen Terme nur noch einen geringen Informationszuwachs darstellen. Ein Vergleich zwischen den beiden verwendeten Arten von Termen zeigt, dass unter Verwendung von *Synsets* eine geringere Anzahl an Bewertungen als bei *Schlüsselworten* benötigt wird, um die für das Verfahren höchste Genauigkeit zu erzielen. Dies liegt darin begründet, dass ein einzelnes *Synset*, aufgrund der dadurch ermöglichten Betrachtung zusätzlicher *synonymer* Worte, eine größere Menge an Informationen als ein *Schlüsselwort* trägt.

Das beste Lernverhalten zeigten die *Taxonomic Recommender*, unabhängig von ihrer jeweils eingesetzten Taxonomie. Bereits nach einer geringen Menge von Bewertungen konnten beide Verfahren die Vorlieben des Benutzers genau erfassen und die Elemente für ihn korrekt klassifizieren. Auch wenn die *Taxonomic Recommender* auf den Ansätzen des *User to User Collaborative Filterings* basieren, ermöglicht es die verwendete Taxonomie selbst bei einer geringen Anzahl bereits bewerteter Elemente, andere Benutzer mit ähnlichen Interessen zu ermitteln. Anhand von Generalisierungen und Spezialisierungen der aus den Elementen ermittelten Vorlieben können selbst dann ähnliche Benutzer ermittelt werden, wenn überhaupt keine gemeinsam bewerteten Elemente vorliegen.

6.5.3 Lernverhalten der Verfahren

Alle auf dem kollaborativen Ansatz basierenden Verfahren sind in starkem Maße abhängig von der Größe und Qualität ihrer Bewertungsmatrix. Zwar verwenden die vorgestellten *Taxonomic Recommender* bei der Erzeugung ihrer Vorhersagen zusätzliches Wissen in Form ihrer Taxonomie, aber auch ihre Arbeitsweise basiert auf der Ermittlung einer Nachbarschaft aus den ähnlichsten Benutzern. Während diese, falls keine gemeinsam bewerteten Elemente vorliegen, auf gemeinsam verwendete Oberkonzepte in den Benutzerprofilen ausweichen können, benötigt das *User to User Collaborative Filtering* zwingend Überlappungen zwischen den Profilen. Wenn von ihnen eine nur ungenügende Anzahl vorliegt, fällt die Genauigkeit des Verfahrens in starkem

Maße. Auch das *Item to Item Collaborative Filtering* benötigt eine ausreichend große Menge an Bewertungen, da es zur Bestimmung der Ähnlichkeiten zwischen den Elementen auf Co-Ratings angewiesen ist. Existieren davon zu wenige in der Bewertungsmatrix, können die Ähnlichkeiten nur ungenau ermittelt werden.

Um das Lernverhalten der Verfahren zu bestimmen, wurden diese in Hinblick auf die Anzahl der in der Bewertungsmatrix enthaltenen Benutzerprofile untersucht. Dazu wurden 100, 250, 500, 1.000, 2.500 respektive 5.000 Benutzer zufällig aus dem für die Evaluierung verwendeten Datensatz gezogen und mit ihnen der *Recommender* trainiert. Für die Testmenge wurden aus den übrigen Benutzern 250 zufällig ausgewählt. Um die Genauigkeit zu messen, wurde für die jeweilige Anzahl gespeicherter Benutzer der *MAE* der Testmenge anhand der in Kapitel 6.4 vorgestellten Methode ermittelt. Beide *Content-Based Recommender* sowie der *Semantic Recommender* wurden hierbei nicht untersucht, da diese ihre Vorhersagen nicht anhand einer Basis von Benutzern erzeugen und somit auch nicht von deren Anzahl abhängig sind.

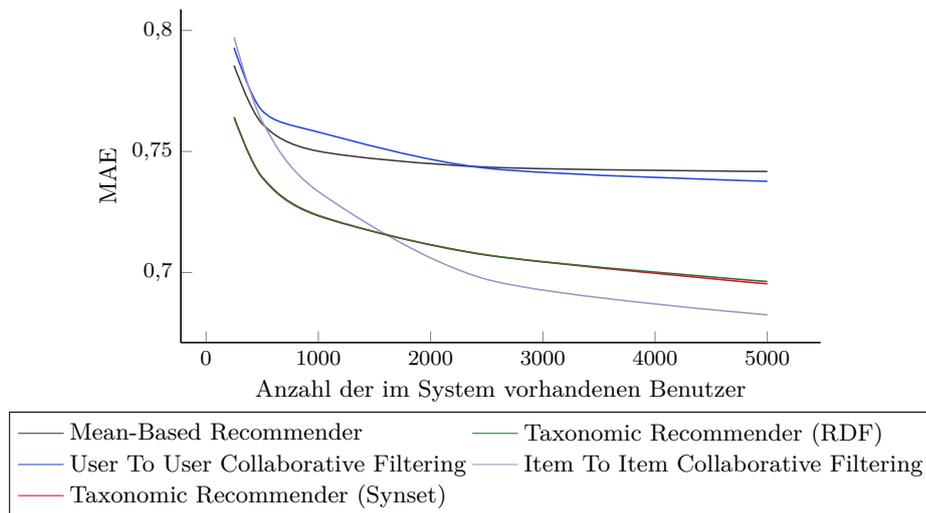


Abbildung 6.16 – Genauigkeit der Algorithmen anhand der *MAE*-Metrik in Abhängigkeit zur Anzahl der zum Training des Verfahrens verwendeten Benutzerprofile

Abbildung 6.16 zeigt die Genauigkeit der verschiedenen auf dem kollaborativen Ansatz basierenden Algorithmen in Abhängigkeit zur Größe der verwendeten Trainingsmenge. Dabei konnte für jedes einzelne der untersuchten Verfahren eine Verbesserung seiner Genauigkeit mit steigender Anzahl der ihm zur Verfügung stehenden Benutzerdaten ermittelt werden. Dies liegt in der immer größer werdenden Menge an Informationen, welche aus den Benutzerprofilen ableitbar sind, begründet und verdeutlicht den Einfluss der *Weisheit der Massen* im kollaborativen Ansatz, der sich insbesondere beim *Mean-Based Recommender* zeigt. Obwohl dieser keine *Personalisierung* anbietet, verbesserte sich die Genauigkeit des Verfahrens mit einer größer werden Anzahl an Profilen, da dann die Aussagen einer Vielzahl unterschiedlicher Benutzer berücksichtigt werden.

Existieren im System jedoch zu wenige Profile, so ist selbst für einen Benutzer mit vielen durchgeführten Bewertungen eine dem *New User*-Problem ähnliche Situation gegeben. Auch wenn an sich über den Benutzer genug Informationen vorhanden sind, können diese aufgrund des Mangels an Vergleichsmöglichkeiten vom System nicht hinreichend genau interpretiert wer-

6.5. Evaluierung der Verfahren

den. Während dies beim *Item to Item Collaborative Filtering* darauf zurückzuführen ist, dass keine ausreichende Anzahl an Co-Ratings zwischen den vorzuschlagenden und den vom Benutzer bereits bewerteten Elementen vorliegt, kann beim *User to User Collaborative Filtering* aufgrund zu weniger passender Vergleichsprofile keine ausreichend große Nachbarschaft ermittelt werden. Somit ist die Genauigkeit der Verfahren entsprechend geringer. Dieses Problem weisen die *Taxonomic Recommender* nicht auf. Falls noch keine Benutzer mit identischen Interessen vorhanden sind, können immer noch anhand von Generalisierungen oder Spezialisierungen Verbindungen zwischen ihnen gefunden werden. Trotz alledem erreichte das *Item to Item Collaborative Filtering*, nachdem genügend Profile im System enthalten waren, die höchste Genauigkeit der getesteten Verfahren.

6.5.4 Skalierbarkeit

Die Skalierbarkeit eines Systems beschreibt sein Verhalten, in Bezug auf eine Veränderung der Verarbeitungszeit bei einer steigende Menge an Daten. Um dies zu ermitteln, wurde zum einen die *Response Time* der Verfahren mit 100, 250, 500, 1.000, 2.500 respektive 5.000 Benutzern und einer festen Anzahl von 1.000 gespeicherten Elementen sowie mit 100, 250, 500, 1.000, 2.500 respektive 3.900 gespeicherten Elementen und einer festen Anzahl von 1.000 Benutzern gemessen. Vor der eigentlichen Messung wurden die für die Durchführung der Evaluierung notwendigen Repräsentationen von Benutzern und Elementen erzeugt sowie die Ähnlichkeiten zwischen den Elementen ermittelt. Die Resultate wurden im Arbeitsspeicher gehalten, so dass das vom *Online*-Betrieb eines *Recommenders* unabhängige *Preprocessing* keinen Einfluss auf die Bestimmung der *Response Time* verübte.

Betriebssystem	Windows 7 Professional 64-Bit
Prozessor	Intel Core 2 Duo E8400 @ 3,6 GHz
Arbeitsspeicher	8 GByte DDR2
Festplatte	500 GByte 5400 U/min
Java Version	1.6.0.24
Zugesicherter Speicher für die VM	6 Gbyte

Tabelle 6.4 – Leistungsdaten des bei der Evaluierung eingesetzten Computers

Das Zwischenspeichern der Repräsentationen im Arbeitsspeicher sowie das im Vorfeld durchzuführende *Preprocessing* hatte jedoch einen sehr hohen Speicherverbrauch zur Folge, so dass die verschiedenen Verfahren selbst mit den zugesicherten 6 GByte an Arbeitsspeicher nicht gleichzeitig auf dem in Tabelle 6.4 aufgeführten Testsystem evaluiert werden konnten. Die Kosten für das *Preprocessing* der Elemente stellen somit einen nicht zu vernachlässigenden Faktor dar. Während das Entfernen der *Stopwörter* aus den *Tags* nur eine geringe und kaum messbare Zeit benötigte, dauerte die Ermittlung der zu ihnen gehörenden *Synsets* aufgrund der vielen durchgeführten Vergleiche in der Taxonomie bereits 15 Minuten. Noch gravierender fiel der Zeitaufwand für die Iterationen des *Semantic Recommenders* aus. So wurde für eine einzelne Iteration auf

dem zur Evaluierung eingesetzten Computer für die verwendeten Testdaten über eine Stunde benötigt.

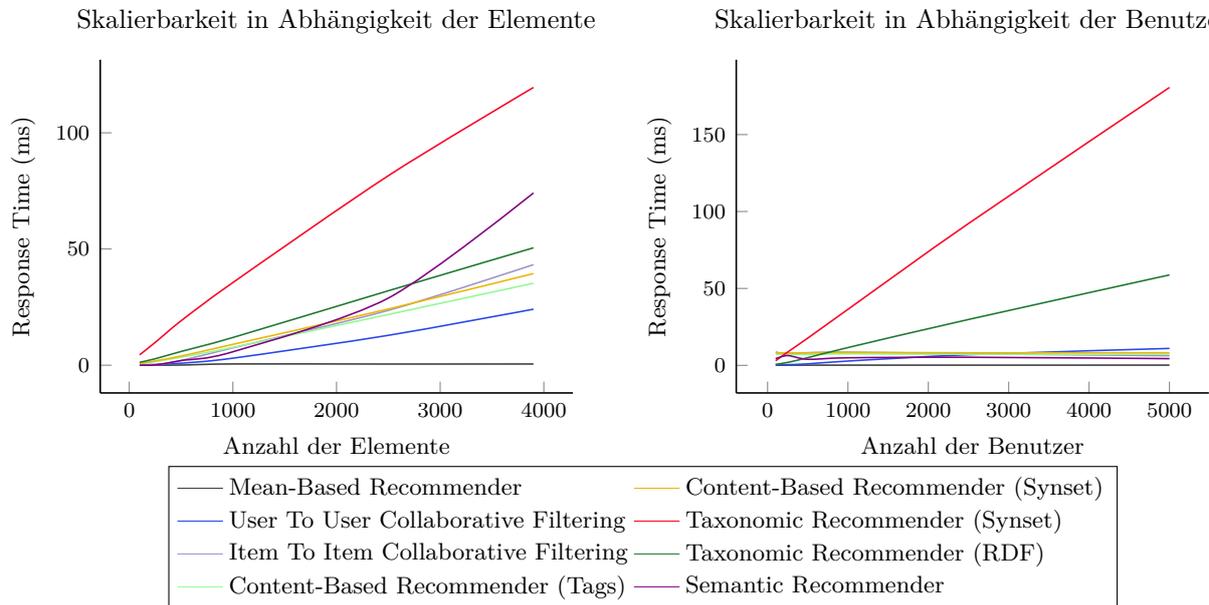


Abbildung 6.17 – Response Time der Verfahren bei der Generierung einer Empfehlung in Bezug auf die steigende Anzahl von Elementen beziehungsweise Benutzern.

Abbildung 6.17 zeigt das Verhalten der Verfahren bei einer steigenden Anzahl von Elementen respektive Benutzern. Der *Mean-Based Recommender* wies jeweils eine nahezu konstante *Response Time* unabhängig dieser beiden variablen Größen auf, was durch das von ihm eingesetzte Caching der Zwischenergebnisse zu begründen ist. Da seine fertigen Ergebnisse bereits im *Preprocessing* erzeugt werden, reduziert sich die Generierung einer Empfehlung auf den Zugriff aus das Ergebnis-Array. Zwar würden sich diese Kosten ebenfalls bei einer größeren Menge an Elementen erhöhen, dies war jedoch mit der bei der Evaluierung verwendeten geringen Anzahl von maximal 3.900 Elementen nicht nachweisbar.

Für den *Semantic Recommender* sowie das *Item to Item Collaborative Filtering* wurde jeweils ein auf die Anzahl der Elemente bezogener quadratischer Anstieg beobachtet. Dies liegt darin begründet, dass diese Verfahren die Elemente miteinander vergleichen, um deren Ähnlichkeiten zu ermitteln. Somit steigen die benötigten Rechenoperationen quadratisch mit steigender Anzahl an Elementen. Im Gegensatz dazu bleiben die Kosten in Bezug auf die Menge der Benutzer konstant, da diese Größe keinen Einfluss aufweist.

Die Unabhängigkeit von der Benutzeranzahl zeigten auch verfahrensbedingt beide *Content-Based Recommender*. Dabei stiegen die Kosten mit einer größeren Anzahl an Elementen linear an. Auch wenn hierbei die Elemente untereinander verglichen werden, beinhaltet ein einzelnes Benutzerprofil nur eine geringe Teilmenge der aus den Elementen extrahierten Terme. Selbst bei einer großen Anzahl an Elementen im System ist davon auszugehen, dass ein einzelner Benutzer nur eine sehr viel kleinere Menge an Elementen bewertet hat, so dass die Größe seines Profil weiterhin klein bleibt.

6.5. Evaluierung der Verfahren

Die *Response Time* des *User to User Collaborative Filterings* und beider *Taxonomic Recommender* zeigten, sowohl bezogen auf die Elemente als auch auf die Benutzer, eine Linearität. Da diese Verfahren die Benutzerprofile und zudem, im Falle einer Empfehlung, auch alle Elemente im System betrachten müssen, steigen die Kosten einer ermittelten Empfehlung entsprechend proportional zur Größe dieser beiden Mengen. Beide *Taxonomic Recommender* benötigten aufgrund der in die Profile aufgenommenen Terme der Elemente eine weit größere Anzahl an Operationen als das *User to User Collaborative Filtering*, welches nur die auf sie abgegebenen Bewertungen betrachtet. Dementsprechend fiel ihre *Response Time* höher aus. Insbesondere die im Vergleich zu den *RDFS*-basierten Repräsentationen erheblich größere Anzahl an enthaltenen *Synsets* wirkte sich gravierend auf die *Response Time* aus.

6.5.5 Coverage

Auch wenn die Coverage für die Benutzer eines *Recommenders* eine eher untergeordnete Rolle spielt, ist sie gerade für den Betreiber eines solchen Systems ein gewichtiger Faktor. Basierend auf der Annahme, dass die Interessen aller Benutzer des Systems über die Elementmenge gleich verteilt sind, zeigt der ermittelte Wert der Coverage inwieweit der *Recommender* Elemente für die unterschiedlichsten Vorlieben der Benutzer vorschlagen kann.

Verfahren	Größe N der Empfehlung
Mean Based Recommender	80
User to User Collaborative Filtering	90
Item to Item Collaborative Filtering	40
Content-Based Recommender (Tags)	50
Content-Based Recommender (<i>Synsets</i>)	110
Taxonomic Recommender (<i>Synsets</i>)	50
Taxonomic Recommender (RDFS)	40
Semantic Recommender	30

Tabelle 6.5 – Größe der Empfehlungsmenge zur Erzielung der höchsten *Precision*

Für die Ermittlung der *Catalog Coverage* wurden ein fester Wert von $N = 30$ verwendet, da sich die untersuchten Verfahren in der optimalen Größe ihrer Empfehlungsmenge stark unterschieden (siehe Tabelle 6.5). Da die Anzahl der in der Empfehlung enthaltenen Elemente den größten Einfluss auf die Coverage aufweist, würden zu divergente Werte diese zu stark zu Ungunsten eines Verfahrens beeinflussen. Ferner wiesen die ermittelten Größen, bei denen die Verfahren die optimale *Precision* erzielten, eine für den Benutzer zu unpraktikable Größe auf. Würde eine solch große Anzahl gewählt werden, so ginge der große Vorteil der *Recommender*, nämlich die Reduktion des Suchraums für den einzelnen Benutzer, verloren.

Nachdem die jeweiligen *Recommender* mit allen Benutzern (6.040) der Evaluierungsdaten trainiert wurden, wurde dann für alle Benutzer ihre Empfehlung der im Vorfeld festgelegten Größe von $N = 30$ bestimmt. Die Anzahl der so erfassten Elemente wurde dann mit der Anzahl aller Elemente (3.953) des Systems verglichen, um so die *Catalog Coverage* (siehe Gleichung 6.9) des *Recommenders* zu ermitteln.

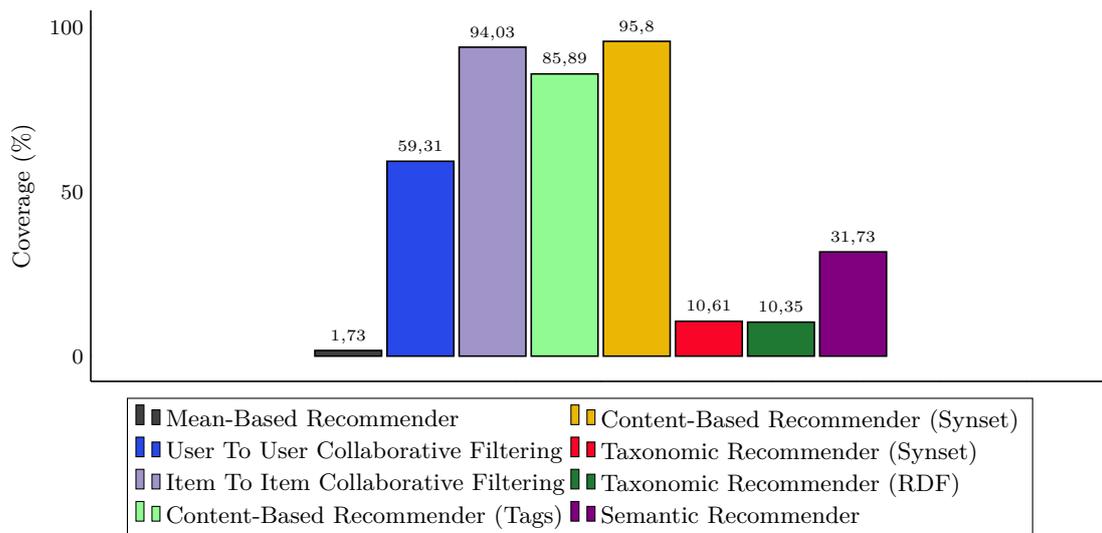


Abbildung 6.18 – Coverage der Verfahren bei einer Größe der Empfehlungsmenge von $N = 30$

Abbildung 6.18 zeigt für die verschiedenen implementierten Verfahren die *Catalog Coverage* bei einer festen Größe der Empfehlungsmenge von $N = 20$. Der *Mean-Based Recommender* erzielte erwartungsgemäß die geringste *Catalog Coverage*, was auf seine Arbeitsweise zurückzuführen ist. So ermittelt das Verfahren für die Empfehlungen der Benutzer unabhängig von ihren Interessen stets identische Elemente, weshalb es andere als diese niemals vorschlagen kann und die *Catalog Coverage* entsprechend gering ist.

Das *Item to Item Collaborative Filtering* hingegen konnte die meisten der Elemente erfassen und in die Empfehlungen aufnehmen. Dieses erlaubt aufgrund der in dieser Domäne hohen Genauigkeit präzise Empfehlungen zu erzeugen. Davon ausgehend, dass die subjektiven Vorlieben der Benutzer über die ganzen Raum der Elemente gleichmäßig verteilt ist, ermöglicht das *Item to Item Collaborative Filtering* das Erfassen nahezu aller Elemente.

Das *User to User Collaborative Filtering* erreichte nicht die *Catalog Coverage* des vorher genannten Verfahrens. Da dessen Vorhersagen von den Bewertungen der in die Nachbarschaft aufgenommenen Benutzer abhängig ist, können die einflussreichen Bewertungen der ähnlichsten Benutzer die Vorhersage negativ beeinflussen. Haben diese das Element schlecht bewertet, so wird es aufgrund ihrer im Speziellen abweichenden Vorlieben nicht aufgenommen.

Die *Content-Based Recommender* erzielten, unabhängig davon, ob *Schlüsselworte* oder *Synsets* eingesetzt wurden, hohe Werte für die *Catalog Coverage*. Da ein *Content-Based Recommender* Elemente, für die keine beschreibenden Terme vorliegen, nicht empfehlen kann limitiert das Fehlen von Daten einiger Elemente des Testsystems die *Catalog Coverage*. Dabei erzielte der *Synset* verwendende *Content-Based Recommender* einen höheren Wert als der *Schlüsselworte* basierte, was auf die durch die *Synsets* ermöglichte Erfassung von Synonymen zurückzuführen ist. Es gilt jedoch zu beachten, dass die *Content-Based Recommender* im Vergleich zu den übrigen Verfahren die größte Anzahl an Elementen fehlerhaft als „relevant“ für einen Benutzer identifiziert hatten.

Die geringe Genauigkeit ist auch die Ursache der geringen *Catalog Coverage* des eingesetzten *Semantic Recommenders*. Da dieser den größten Anteil an *False Negatives* aufwies, konnte er

6.6. Diskussion

nicht die Empfehlungen entsprechend der Benutzervorlieben erzeugen. Erschwerend kommt in diesem Fall die hohe Anzahl an *False Negatives*. Dies führte dazu, dass das System nur einige wenige „Favoriten“ auswählte und diese den Benutzern empfahl.

Ein ähnliches Problem wiesen auch beide *Taxonomic Recommender* auf. Im Gegensatz zum vorher genannten Verfahren erzielten sie jedoch eine hohe Genauigkeit sowohl in Bezug auf die Vorhersagen als auch bei der Klassifikation der Elemente. Trotz alledem unterschieden sich die Empfehlungen außer in den für den jeweiligen Benutzer korrekt „relevant“ klassifizierten nur geringfügig in ihren enthaltenen Elementen. Ferner existierten nicht für alle im System enthaltene Elemente komplette Beschreibungen, so dass insbesondere beim *RDF* basierten *Taxonomic Recommender* erschwerend auch Datenintegrationsprobleme hinzukamen, welche die *Coverage* negativ beeinflussen.

6.6 Diskussion

Nachdem die verschiedenen, für einen Einsatz der unterschiedlichen *Recommender Systeme* relevanten Parameter einzeln analysiert (siehe Kapitel 6.4) und in Hinblick auf verschiedene für den Betrieb eines *Recommenders* wichtige Kriterien untersucht wurden (siehe Kapitel 6.5), dient die Diskussion der Betrachtung der durch die Arbeit ermöglichten Erkenntnisse in ihrer Gesamtheit. Dies ermöglicht eine abschließende Bewertung, inwieweit der Einsatz von *semantischen* Technologien einen Vorteil für die Benutzer und Betreiber eines *Recommenders* darstellen kann.

Evaluierungsmethodik

Die in Kapitel 5.2 vorgestellten Verfahren wurden anhand eines im Vorfeld erhobenen festen Datensatzes quantitativ in einer *Offline*-Evaluierung untersucht (siehe Kapitel 6.5). Neben einer objektiven Vergleichbarkeit der Ergebnisse anhand anerkannter statistischer Verfahren bietet dies den Vorteil, über einen längeren Zeitraum gesammelte Bewertungen der Benutzer zu verwenden. Da der Zeitraum beliebig und vor allem unabhängig von der eigentlichen Evaluierung ist, kann den Benutzern auch ermöglicht werden, ihre Bewertungen nachträglich anzupassen, so dass diese sehr exakt ihre subjektiven Interessen repräsentieren. Im Gegensatz dazu werden in einer *Online*-Evaluierung die Elemente vom Benutzer im *Frontend* ohne Wissen über die mögliche Existenz eines passenderen Elementes bewertet und auch ohne die Möglichkeit, zu einem späteren Zeitpunkt eine Anpassung der Angaben vornehmen zu können.

Einige Kriterien lassen sich jedoch nicht alleine durch eine quantitative *Offline*-Evaluierung messen. Auch wenn bereits Metriken entwickelt wurden, um die subjektiven Kriterien der *Überspezialisierung* und der *Serendipität* numerisch zu approximieren, bilden die dabei ermittelten Werte das Verhalten der *Recommender* nur sehr ungenau ab. Erst die in einer qualitativen *Online*-Evaluierung durchgeführte Benutzerbefragung oder Beobachtung würde es ermöglichen, diese sehr subjektiven Kriterien entsprechend zu erfassen und die Verfahren dabei weitergehend zu vergleichen.

Ferner stellten sich die beiden Metriken der *Precision* sowie des *Recalls* für den vorliegenden Anwendungsfall als weniger geeignet für die Bestimmung der Genauigkeit von Empfehlungen heraus. Durch die Annahme, dass von einem Benutzer noch unbewertete Elemente keine Relevanz

für diesen aufweisen, sank die *Precision* auf kleine Werte, selbst wenn in der Menge auch Elemente von subjektivem Interesse enthalten waren. In einer *Online* durchgeführten Evaluierung wären hingegen, aufgrund des auf der subjektiven Wahrnehmung eines Benutzers basierenden Feedbacks, genauere Aussagen über die Gültigkeit der Empfehlungsmenge möglich gewesen.

Der *Recall* hatte, bedingt durch seine von der *Precision* abweichende Größenordnung, einen solch starken Einfluss auf den für die *F1*-Metrik ermittelten Wert, dass dieser ab einer gewissen Größe der Empfehlungsmenge nahezu ausschließlich von ihm abhing. Somit stieg der Wert der *F1*-Metrik entsprechend der Mächtigkeit. Diese Resultate haben jedoch keinen ausreichenden praktischen Bezug. Würde die Anzahl der vorgeschlagenen Elemente zu groß werden, verlieren die *Recommender* ihren entscheidenden Vorteil, nämlich die auf einen einzelnen Benutzer abgestimmte Reduktion seines Suchraums. Für kleinere und damit im realen Betrieb relevante Größen hingegen zeigte sich die *F1*-Metrik ausschließlich abhängig von der *Precision*. Trotz ihrer angeführten Einschränkung ermöglicht daher die *Precision* einen hinreichend guten Vergleich der einzelnen Verfahren.

Somit bleibt festzuhalten, dass in einer *Offline*-Evaluierung mit statistisch etablierten Metriken die Genauigkeit von Vorhersagen ermittelt werden kann. Um Empfehlungen zu evaluieren, müssten allerdings andere als die hier eingesetzten Metriken der *Precision*, des *Recalls* sowie der daraus abgeleiteten *F1*-Metrik verwendet werden. Zwar ermöglichen diese eine grobe Einschätzung der Genauigkeit der Verfahren, die beste Möglichkeit dazu würde eine Befragung oder Beobachtung der Benutzer in einer *Online*-Evaluierung bieten.

Genauigkeit der Verfahren

In der durchgeführten Evaluierung wies das *Item to Item Collaborative Filtering* die höchste Genauigkeit in den Vorhersagen der untersuchten Verfahren auf (siehe Abbildung 6.11). Diese hohe Genauigkeit wies das Verfahren sowohl bei der Klassifikation der Elemente (siehe Abbildung 6.12) als auch für die Empfehlungen auf (siehe Abbildung 6.13). Die Abstraktion dieses *semantisch* einfachen Verfahrens von den direkten Elementeigenschaften ermöglichte in der vorliegenden und vor allem von subjektiven Vorlieben geprägte Domäne von Filmen eine gute Erfassung der Bedeutung eines Elementes. Interessanterweise konnte der *Mean-Based Recommender*, auch ohne die Vorlieben der Benutzer im Einzelnen zu betrachten, gute Vorhersagen generieren. Dies kann mit der *Weisheit der Massen* in der hier verwendeten Domäne begründet werden. Insbesondere Filme hoher objektiver Qualität werden von verschiedenen Benutzern unabhängig vom persönlichen Geschmack meist ähnlich hoch bewertet. Auch das *User to User Collaborative Filtering* erzielte für die Vorhersagen eine höhere Genauigkeit als die inhaltsanalysierenden *Content-Based Recommender* und *Semantic Recommender*.

Die weiteren Verfahren benötigen zwingend eine aussagekräftige Repräsentation, um die Elemente entsprechend in ihrem Sinn erfassen zu können. Zwar konnte durch die Verwendung von den Inhalt der Elemente beschreibenden *Schlüsselwörtern* beziehungsweise *Synsets* auch der Einsatz eines *Content-Based Recommenders* für die vorliegende Domäne der Filme realisiert werden, dieser erzielte jedoch nicht die Genauigkeit der kollaborativen Verfahren. Problematisch war insbesondere die Tatsache, dass die einzelnen Filme durch eine verschiedene Anzahl an Termen

6.6. Diskussion

beschrieben wurden, so dass deren Repräsentation in unterschiedlicher Qualität vorlagen. Trotz alledem konnten die *Content-Based Recommender* zuverlässig für einen Benutzer relevante Filme auch als solche erkennen, was sich in ihrer geringe Anzahl an *False Negatives* darstellte (siehe Abbildung 6.12). Noch deutlicher zeigte sich die Abhängigkeit der inhaltsanalysierenden Verfahren von der Qualität der Modellierung beim eingesetzten *Semantic Recommender*. So konnte dieser nicht die Genauigkeit der anderen vorgestellten Verfahren erzielen, was jedoch nicht auf das Verfahren selbst, sondern vielmehr auf seinen Einsatz mit einem einfachen *Domänen-Modell* zurückzuführen ist.

Beide *Taxonomic Recommender* erreichten eine Genauigkeit bezüglich des *MAE*, welche die der meisten anderen Verfahren übertraf. Lediglich das *Item to Item Collaborative Filtering* konnte bessere Werte erzielen. Zwar generieren die *Taxonomic Recommender* ihre Vorhersagen analog zum *User to User Collaborative Filtering* durch die Ermittlung einer zum Benutzer ähnlichen Nachbarschaft, verwenden dabei aber eine verbesserten Vergleichsmethode (siehe Kapitel 5.2.4). Diese basiert nicht auf Überlappungen zwischen den Bewertungsvektoren der Benutzer, sondern auf ihren Interessensgebieten, welche anhand einer Analyse der von ihnen bewerteten Elemente ermittelt werden. Unter Verwendung einer diese Interessen hierarchisierenden Taxonomie werden die einzelnen Interessen untereinander in Relation gestellt, was die hohe Genauigkeit der *Taxonomic Recommender* erklärt. So war es ihnen möglich, eine geringere Anzahl an Elementen irrtümlich als „relevant“ zu klassifizierten, auch wenn sie für die *False Negatives* schlechtere Resultate erzielten. Beides lässt sich dadurch erklären, dass die von ihnen berechneten Werte für die Vorhersagen generell zu gering waren.

Probleme der offenen Verfahren

Das Verfahren des *Semantic Recommenders* schien für die, aus wenigen unterschiedlichen *Properties* und Konzepten bestehende Ontologie des Testsystems (siehe Abbildung 6.2) weniger geeignet zu sein. Insbesondere die strikte Baumstruktur der Taxonomien wurden durch den Netzwerk-basierenden Ansatz des Verfahrens nur unzureichend berücksichtigt. Ferner konnte die höchste Genauigkeit dann erzielt werden, wenn ausschließlich die Konzepte in einer kleinen Umgebung betrachtet wurden. Wurde diese zu groß gewählt, hatten auch solche Konzepte einen Einfluss, die an sich keinen Zusammenhang zum vorzuschlagenden Element aufwiesen. Dies liegt vor allem an der geringen Distanz zwischen den einzelnen Elementen und Konzepten (siehe Kapitel 6.4.4). In einer Domäne wie beispielsweise der Nachrichten, welche eine größere Menge an Konzepten unterschiedlicher Themengebieten beinhaltet und in der die vorzuschlagenden Elemente in der Ontologie eine höhere Distanz zueinander aufweisen, wären voraussichtlich bessere Ergebnisse zu erwarten.

Ferner können die Probleme auf die Offenheit des Verfahrens zurückgeführt werden. Da es aufgrund seiner domänenunabhängigen Arbeitsweise nicht die jeweilige Bedeutung einer *Property* berücksichtigt, werden diese immer unabhängig von ihrer Signifikanz für das Element identisch behandelt. Dabei können auch Eigenschaften verwendet werden, welche den Inhalt der Elemente nicht oder nur unzureichend beschreiben. So hat in der vorliegenden Repräsentation der Elemente das Entstehungsjahr eines Films sicherlich eine geringere Aussagekraft als das Gen-

re, dem er angehört. Dieser Umstand wird jedoch aufgrund der Offenheit des Verfahrens nicht berücksichtigt, so dass die errechnete Ähnlichkeit zwischen zwei Elementen negativ beeinflusst werden kann. Somit müsste, um genauere Vorhersagen zu erzielen, im Vorfeld ermittelt werden, welche der *Properties* betrachtet beziehungsweise wie diese ihrer Signifikanz entsprechend gewichtet werden.

Verwendung von Konzepten

In dem für die Evaluierung verwendeten Datensatz konnte durch den Einsatz von *Synsets* anstelle einfacher *Schlüsselworte* im Gegensatz zur Annahme (siehe Kapitel 5.2.3) keine Verbesserung der Genauigkeit beobachtet werden. Die dabei erzielten Ergebnisse verschlechterten sich sogar geringfügig gegenüber der Verwendung einfacher *Schlüsselworte* (siehe Kapitel 6.7). Dies ist durch die geringe Anzahl unterschiedlicher, für die Auszeichnung der Elemente verwendeten Worte zu erklären, sowie in der teils fehlerhaften Ermittlung der zu den *Schlüsselworten* gehörenden *Synsets* begründet. Anhand manueller Stichproben konnte ein Fehler von ungefähr 10 % bei der Zuweisung der Konzepte aufgrund des verwendeten Verfahrens beobachtet werden (siehe Kapitel 6.1). Dieser Fehler könnte durch verbesserte linguistische Verfahren minimiert werden.

Ferner verwendeten die Benutzer beim *Collaborative Tagging* der Elemente bereits identische Worte gleicher Deklination für die Beschreibung derselben Eigenschaften, so dass bereits unbewusst von ihnen ein eindeutiges Vokabular für die Domäne trotz des an sich offenen Prozesses realisiert wurde (siehe Kapitel 4.2.2). Für diese Annahme spricht, dass über 80 % der im System verwendeten *Synsets* aus genau einem Wort bestehen (siehe Tabelle 6.2), da dies einen Indikator für die Eindeutigkeit der zur Beschreibung der Elemente verwendeten *Schlüsselworte* darstellt. Würde das Verfahren jedoch auf eine andere Domäne mit einer größeren Menge unterschiedlicher Worte wie beispielsweise wissenschaftlicher Publikationen und vor allem auf eine Sprache mit einer größeren Anzahl an *Synonymen*, wie beispielsweise Deutsch, angewendet werden, so wären dabei bessere Ergebnisse zu erwarten.

Die Verwendung von *Synsets* führte auch beim *Taxonomic Recommender* zu einer leichten Verschlechterung der Genauigkeit in Bezug auf die Anzahl der *False Positives* 6.12. Dennoch generierte er Vorhersagen einer weit höheren Genauigkeit als die *Content-Based Recommender* (siehe Abbildung 6.11). Dies liegt jedoch darin begründet, dass *Synsets* die Erzeugung einer Taxonomie ermöglichen, in der diese anhand von Ober- und Unterbegriffen hierarchisiert werden. Eine weitere Taxonomie wurde anhand der Kategorien, welcher die Elemente angehören, gebildet und in *RDFS* formal beschrieben. Die jeweils unterschiedliche Arten von Taxonomien verwendenden *Taxonomic Recommender* genierte dabei Vorhersagen von vergleichbarer Genauigkeit (siehe Abbildung 6.11). Dies bestätigt die in Kapitel 4.2.3 enthaltene Aussage, dass der Einsatz von Generalisierungen und Spezialisierungen zur Verbesserung der Genauigkeit beiträgt.

Lernverhalten

Zwar erzielte das *Item to Item Collaborative Filtering* die höchste Genauigkeit für die ermittelten Vorhersagen, diese konnte jedoch erst erreicht werden, nachdem ein einzelner Benutzer mehrere hundert Elemente bewertet hatte. Ein ähnliches Verhalten konnte auch bei der Evaluierung des *User to User Collaborative Filterings* beobachtet werden (siehe Abbildung 6.5.2).

6.6. Diskussion

Dies lässt sich darauf zurückführen, dass beide Verfahren für die Generierung von Vorhersagen und Empfehlungen zwingend gemeinsam bewertete Elemente voraussetzen (siehe Kapitel 5.2.2), welche bei einer zu geringen Anzahl an Bewertungen eines Benutzers nicht vorliegen.

Beim *User to User Collaborative Filtering* führt dies zu einem Mangel an Überlappungen zwischen den Bewertungsvektoren, so dass nicht genügend ähnliche Benutzer ermittelt werden können. Beim *Item to Item Collaborative Filtering* liegt in diesem Fall keine ausreichende Menge an Co-Ratings vor, so dass die Ähnlichkeit zwischen dem zu empfehlenden und den bereits bewerteten Elementen nicht hinreichend genau ermittelt werden kann. Liegt hingegen eine Vielzahl an Bewertungen vor, so können die Interessen eines sogenannten *Heavy-Users* genau identifiziert und so für ihn Vorhersagen höchster Präzision durch das *Item to Item Collaborative Filtering* generiert werden.

Neben einem ermittelten besseren Lernverhalten für die Benutzerprofile erlaubten die Verfahren eines höheren *semantischen* Grades auch die *explizite* Angabe von Interessen (siehe Kapitel 4.2), die eine verbesserte Erfassung der Vorlieben ermöglicht und zudem zur Initialisierung eines noch neuen Benutzerprofils verwendet werden kann. Dies kann durch das *Collaborative Filtering* nicht realisiert werden, da es die subjektiven Interessen aus der Kombination der vom Benutzer abgegebenen Bewertungen ableitet. Dabei wurde das beste Lernverhalten für den verwendeten Datensatz unter Berücksichtigung der erzielten Genauigkeit bei den *Taxonomic Recommendern* beobachtet. Diesem Verfahren war es bereits nach wenigen Bewertungen eines einzelnen Benutzers möglich, für ihn Vorhersagen von optimaler Genauigkeit zu erzeugen. Zum einen ist dies auf die verbesserte Erfassung der Benutzervorlieben zurückzuführen, zum anderen ermöglicht es die Verwendung einer Taxonomie im kollaborativen Ansatz selbst dann ähnliche Benutzer zu ermitteln, wenn keine gemeinsam bewerteten Elemente vorliegen. In diesem Fall kann auf gemeinsam in den Profilen vorhandene Oberkonzepte ausgewichen werden. Dabei reichen bereits wenige bewertete Elemente aus, um eine hinreichend große Anzahl an Oberkonzepten zu bestimmen. Während sich das *Item to Item Collaborative Filtering* und das *User to User Collaborative Filtering* vor allem für *Heavy-User* eignen, bietet sich der Einsatz eines *Taxonomic Recommenders* insbesondere für sogenannte *Casual-User* an, die nur eine kleine Anzahl an Elementen bewerten.

Ferner konnte ermittelt werden, dass sich das verbesserte Lernverhalten nicht nur auf die Benutzerprofile bezog, sondern auch auf das Verfahren als solches. Da die Genauigkeit der Vorhersagen aller kollaborativen Verfahren in großem Maße abhängig von ihren gespeicherten Benutzerprofilen ist, zeigen auch diese ein Lernverhalten in Abhängigkeit zu deren Anzahl. Insbesondere in der Anlaufphase, wenn erst eine ungenügende Anzahl an Profilen vorhanden ist, konnten sowohl das *User to User Collaborative Filtering* als auch das *Item to Item Collaborative Filtering* nur ungenaue Vorhersagen generieren (siehe Abbildung 6.16). Beide *Taxonomic Recommender* zeigten ein identisches Verhalten und konnten bereits mit wenigen im System enthaltenen Profilen genauere Vorhersagen erzeugen als die übrigen Verfahren. Auch hier ermöglichte die Verwendung der Taxonomie, trotz Fehlens gemeinsamer Bewertungen, Benutzer mit ähnlichen Vorlieben anhand von Oberkonzepten ihrer bewerteten Elemente zu ermitteln. Dabei verbesserte sich ihre Genauigkeit mit steigender Anzahl vorhandener Profile. Sobald jedoch eine ausreichende Anzahl an Profilen vorlagen, erzielte das *Item to Item Collaborative Filte-*

ring seine hohe Genauigkeit, welche die der *Taxonomic Recommender* überstieg, was jedoch nur durch seine generell höhere Präzision bedingt ist.

Coverage und Datenintegration

Neben der Einschränkung der *Catalog Coverage* aufgrund ihrer jeweiligen Arbeitsweise (siehe Kapitel 6.5.5) ist der ermittelte geringe Wert der *Taxonomic Recommender* durch Probleme bei der Bestimmung der für die Evaluierung verwendeten Daten zu begründen. Erst durch eine Unterstützung der einzelnen Schritte in Form einer manuell durchgeführten Kontrolle konnten die Informationen für die meisten Filme des Datensatzes ermittelt werden (siehe Kapitel 6.1). Trotz alledem befanden sich für fast 10 % der Filme keine Einträge in der dbpedia, so dass diese nicht durch das Verfahren erfasst und somit auch nicht vorgeschlagen werden konnten.

Die vielen unter manueller Kontrolle vorgenommenen Schritte zeigen anschaulich die Notwendigkeit einer automatischen Datenintegration. Während die manuelle Kontrolle für den überschaubar kleinen Datensatz von knapp 4.000 Elementen noch durchführbar ist, wäre dies in einem realen System im *E-Commerce* mit mehreren hunderttausend Produkten nicht mehr umzusetzen. Weiterhin wären die nicht erfassten 10 % dann gleichzusetzen mit mehreren 10.000 Produkten, die niemals verkauft werden könnten und wertvollen Lagerplatz verbrauchen. Der Einsatz von *semantischen* Technologien in Form von standardisierten Vokabularen für die Beschreibung der Elemente kann eine Lösung dieser Problematik ermöglichen. Diese bewirken eine verbesserte Integration der Elemente aufgrund der unmissverständlichen Definition ihrer Eigenschaften und stellen einen Identifikationsmechanismus bereit, durch den die Elemente weltweit referenziert und eindeutig unterschieden werden können. Selbst wenn die vorhandenen Elemente noch Lücken in ihren Beschreibungen aufweisen, können unter Verwendung der Identifikatoren aus externen Quellen, wie beispielsweise den verschiedenen in der Linked Open Data Cloud enthaltenen Datensätzen, weitere, sie beschreibende Informationen ermittelt werden, die dann die fehlenden Daten ausgleichen. Kann dabei ein praktikabler Wert für die *Catalog Coverage* erzielt werden, so kann diese dazu dienen, „Ladenhüter“ zu identifizieren und den Produktkatalog eines *Online-Shops* zu optimieren.

Skalierbarkeit

Trotz ihrer Vorteile wiesen die auf der Verwendung *semantischer* Technologien basierenden Verfahren große Probleme im Hinblick auf die Skalierbarkeit auf. So benötigte der Algorithmus für den *Semantic Recommender* zur Bestimmung seiner Zwischenergebnisse bereits für die geringe Menge des Testsystems von 1.090.898 Statements 6 GByte Arbeitsspeicher. Für eine größere Menge an Daten ist von daher eine solche Lösung nicht praktikabel. Zwar können die Zwischenergebnisse in einen Hintergrundspeicher ausgelagert werden, was jedoch einen starken Zuwachs der für eine Iteration benötigten Zeit um den Faktor zehn zu Folge hat. Trotz alledem kann die eigentliche Generierung der Empfehlungen von der Bestimmung der Ähnlichkeiten entkoppelt werden, so dass dieser rechen- und zeitintensive Prozess des *Preprocessings* auch *offline* durchgeführt werden kann.

Ein solcher Prozess kann jedoch nicht für die *Taxonomic Recommender* eingesetzt werden, da hier die Ähnlichkeiten der Benutzerprofile *online* bestimmt werden müssen. Eine ausreichende

6.6. Diskussion

Verarbeitungsgeschwindigkeit kann nur dann erzielt werden, wenn die für die Vergleiche herangezogenen Benutzerprofile im Arbeitsspeicher gehalten werden. Im Testsystem wurde bereits für das Caching der 6.000 aus den Evaluierungsdaten generierten Profile 2 GByte Arbeitsspeicher benötigt. Damit das Verfahren auch bei einer größeren Anzahl an Benutzern einsetzbar ist, müsste die Menge der Vergleichsprofile daher reduziert werden. Dies könnte beispielsweise durch eine Verkleinerung der Matrixgröße anhand einer *Singular Value Decomposition* geschehen, wie sie bereits im *User to User Collaborative Filtering* erfolgreich eingesetzt wurde (siehe Kapitel 2.1.1). Eine weitere Möglichkeit besteht im Zusammenfassen von Benutzerprofilen zu *Stereotypen*. Im Gegensatz zu einem *Demographic Recommender* werden dabei die Cluster nicht statisch durch die demografischen Daten der Benutzer (siehe Kapitel 4.1.1), sondern dynamisch anhand von ähnlichen Interessen gebildet. Neben der Reduzierung des Speicherverbrauchs kann dadurch auch die Anzahl der durchgeführten Operationen stark verringert werden. Es ist anzunehmen, dass dies auch die Performanz in großem Maße positiv beeinflusst.

Als gravierender Faktor für die Skalierbarkeit, unabhängig von den einzelnen Verfahren, hat sich die verwendete Programmiersprache *JAVA* erwiesen. In der Implementierung der jeweiligen Verfahren wurden weitestgehend primitive Datentypen anstatt von Objekten verwendet und die langsamen Collections der Sprache durch performantere Bibliotheken getauscht. Selbst deren Einsatz wurde durch eine entsprechende Verwendung von Arrays anstelle abstrakter Datentypen minimiert. Die größte Einschränkung der Performanz für die Implementierung der Algorithmen liegt aber in der Programmiersprache *JAVA* selbst begründet. Zwar erreicht diese bei einfachen arithmetischen Operationen eine ähnliche Geschwindigkeit wie die maschinennaheren Sprachen *C* oder *C++*, der Zugriff auf die Felder eines Arrays ist jedoch mit zusätzlichen Kosten verbunden. Während die anderen genannten Sprachen einen direkten Zugriff ermöglichen, wird bei *JAVA* zuerst die Gültigkeit des Zugriffs überprüft, was bei mehreren Millionen durchgeführter Operationen für die Bestimmung einer einzelnen Empfehlung erhebliche Zusatzkosten aufwirft. Aus diesem Grund müsste ein *Recommender* für ein operatives System in einer Programmiersprache realisiert werden, welche diese Kosten nicht aufweist.

Fazit

Die große Genauigkeit des *Item to Item Collaborative Filterings* lässt sich anhand der verwendeten Domäne erklären. In dieser basieren die Vorlieben eines Benutzers hochgradig auf subjektive Kriterien. Da die einfachen Bewertungsvektoren des *Collaborative Filterings* von den konkreten Eigenschaften eines Elementes abstrahieren, ermöglichen sie auch diese Kriterien zu erfassen. Die *Content-Based Recommender* sowie der verwendete *Semantic Recommender* generieren hingegen ihre Vorhersagen anhand einer objektiven Beschreibung der Elemente, welche sich in der verwendeten Domäne nur schwer realisieren lassen. Deren Qualität hat jedoch einen hohen Einfluss auf die Genauigkeit der letztgenannten Verfahren, was ihre vergleichsweise hohe Ungenauigkeit erklärt. In einer anderen Domäne wie beispielsweise im Umfeld der Consumer-Electronics, in der das Nutzen eines Elementes anhand objektiver Kriterien festgestellt werden kann, würden auch diese Verfahren vergleichbare Resultate erzielen können.

Bereits heutzutage werden die Produktkataloge von *Online-Shops* in Form einer Taxonomie von Produktkategorien realisiert. Ausgehend von dieser vorhandenen Baumstruktur lassen sich die angebotenen Produkte anhand ihrer sie beschreibenden Kategorien klassifizieren und hierarchisieren. Dadurch liegt die von den *Taxonomic Recommenders* benötigte Taxonomie bereits vor. Durch die stetig wachsende Verwendung eindeutiger und formal definierter Vokabulare zur Beschreibung von Produkten und Dienstleistungen seitens deren Anbieter, entfällt weiterhin der erhebliche und meist fehlerbehaftete Aufwand, diese in das interne Repräsentationsformat des *Online-Shops* zu überführen. Somit können die aktuellen *Online-Shops* leicht für den Einsatz eines *Taxonomic Recommenders* angepasst werden, so dass deren vielfältige Vorzüge sofort genutzt werden können. Werden ihre vielen Vorteile im Vergleich zu den restlichen Verfahren betrachtet, so bestätigen sie die Annahme, dass mit der Kombination eines *Knowledge-Based Recommenders* mit dem *Collaborative Filtering* ein idealer *Hybrid Recommender* realisiert werden kann (siehe Kapitel 2.4).

Kapitel 7

Zusammenfassung und Ausblick

7.1 Zusammenfassung

Aufgrund der vielen Vorteile für die Benutzer und Anbieter von Informationsportalen und *Online-Shops*, findet die *Personalisierung* eine immer größer werdende Verbreitung. Gerade im *E-Commerce* stellt ihr Einsatz eine starke Verbesserung gegenüber den traditionellen *Online-Shops* dar. Die *Personalisierung* berücksichtigt die speziellen Wünsche einzelner Kunden und ermöglicht daher eine auf sie angepasste Navigation, welche die Kunden in ihrer Suche unterstützt. Für die Betreiber eines *Online-Shops* manifestiert sich die so gesteigerte Kundenzufriedenheit in höheren Umsätzen und einer gesteigerten Kundenbindung.

Recommender Systeme stellen die am stärksten verbreitete Lösung dar, um ein zum Benutzer *adaptives* System umzusetzen. Trotz vieler Vorteile der unterschiedlichen Verfahren, arbeitet keine der etablierten Techniken ohne Einschränkungen. Erst eine Kombination unterschiedlicher Grundsysteme zu einer neuen, *Hybrid Recommender* genannten Architektur ermöglicht es, zumindest einige der Probleme zu reduzieren. Dabei konnte insbesondere die Kombination mit einem anhand einer *Wissensbasis* arbeitenden *Knowledge-Based Recommender* entscheidende Vorteile aufzeigen.

Mit den Techniken des *Semantic Webs* existiert ein mächtiges Werkzeug, um die von den *Knowledge-Based Recommendern* benötigte *Wissensbasis* zu erzeugen. Durch die Prinzipien der global eindeutigen Beschreibung und der Verwendung etablierter Standards kann dabei eine modulare und wiederverwendbare *Wissensbasis* für die *Recommender* geschaffen werden. Weiterhin entfällt dadurch der fehlerbehaftete und aufwändige Prozess, die Elemente in das System des *Recommenders* zu integrieren. Durch den Einsatz *semantischer* Technologien in den *Recommendern* kann auch weiteres Wissen in Form eines Domänen-Modells sowie aus externen Quellen für die Erzeugung ihrer Empfehlungen bereitgestellt werden, wodurch neben der Erhöhung der Genauigkeit auch die Lösung weiterer Probleme ermöglicht werden kann. Diese umfassen beispielsweise die Abhängigkeit kollaborativer Verfahren von ihrer Benutzerbasis oder auch die verbesserte Erkennung der Interessen neuer Benutzer.

In der Modellierung der Benutzer und Elemente verwenden die einzelnen Verfahren in ihrem Betrieb Repräsentationen unterschiedlichen *semantischen* Grades. Während die einfachsten Verfahren lediglich simple Identifikatoren zur Unterscheidung der Benutzer beziehungsweise

7.2. Ausblick

der Elemente verwenden, kommen bei steigendem *semantischen* Grad Repräsentationen in Modellierungssprachen des *Semantic Webs* mit größer werdender Formalität zum Einsatz. Diese ermöglichen die Erzeugung immer differenzierterer und aussagekräftigerer Benutzerprofile und Elementrepräsentationen. Unter Verwendung eines *Domänen*-Modells in Form einer formal beschriebenen *Wissensbasis* werden die Profile und Repräsentationen mit weiteren Informationen angereichert, so dass deren Bedeutung von den *Recommendern* in einem besseren Maße und vor allem eindeutig interpretiert werden kann.

In einem durchgeführten objektiven Vergleich verschiedener implementierter Verfahren anhand einheitlicher Testdaten konnten mehrere Vorteile von *Recommendern* durch den Einsatz *semantischer* Technologien identifiziert werden. Dabei zeigten insbesondere die *Taxonomic Recommender* Vorteile gegenüber den traditionellen Verfahren. Auch wenn diese in der untersuchten Domäne nicht die Genauigkeit des *Item to Item Collaborative Filterings* erzielten, war es ihnen dennoch möglich, die Anzahl der fälschlicherweise für „relevant“ befundenen Elemente zu reduzieren. Ferner zeigten sie eine weit geringe Abhängigkeit von der Größe der zum Training des *Recommenders* benötigten Anzahl an Benutzern. Selbst für einen einzelnen Benutzer war eine viel geringere Anzahl von ihm abgegebener Bewertungen ausreichend, um seine Interessen und Vorlieben erfassen zu können. Diese Verbesserungen wurden von den *Taxonomic Recommender* erzielt, obwohl ihnen für den Betrieb benötigte Informationen bei 10 % der Elemente nicht zur Verfügung standen. Wären diese vorhanden gewesen, hätten voraussichtlich noch bessere Resultate erzielt werden können.

Anhand des verwendeten Verfahrens des *Semantic Recommenders* konnte die Notwendigkeit einer genauen Beschreibung der Domäne und Elemente gezeigt werden. Da diese nur unzureichend vorlag, konnte das System nicht sein ganzes Potential entfalten. Sowohl der *Taxonomic* als aus der *Semantic Recommender* wiesen zudem eine gravierende Einschränkung auf, welche beim Einsatz *semantischer* Technologien oftmals zu verzeichnen ist. Neben ihrer Abhängigkeit von einer „guten“ *Wissensbasis* zeigten sie nur eine mangelnde Skalierbarkeit für eine wachsende Menge an Elementen. Sollten die Verfahren im *E-Commerce* eingesetzt werden, müsste dieses Problem zuerst gelöst werden.

7.2 Ausblick

Auch wenn die vorgestellten Verfahren durch die Verwendung *semantischer* Technologien einerseits viele Vorteile gegenüber den traditionellen Algorithmen zeigen, weisen sie andererseits verschiedene Einschränkungen auf. Insbesondere das Problem, auch mit steigender Datenmenge hinreichend schnell Empfehlungen zu erzeugen, steht ihrem Einsatz im *E-Commerce* im Wege. Somit muss zuerst die Skalierbarkeit für einen Einsatz mit Millionen von Produkten und Benutzern gewährleistet werden, weshalb dieser Faktor ein hohes Optimierungspotential für die Verfahren bietet.

Während die *Taxonomic Recommender* durch eine Hierarchisierung der den Elementen zugeordneten Kategorien Vorhersagen mit zu den traditionellen Verfahren vergleichbarer Genauigkeit erzielten, konnte dies das eingesetzte Verfahren für den *Semantic Recommender* nicht erreichen.

Bedingt durch seine offene Arbeitsweise betrachtet es alle Eigenschaften identisch, unabhängig ihres eigentlichen Nutzens. Dadurch verwendet er auch solche Informationen, die nur eine geringe Bedeutung für das Element aufweisen, die ermittelte Ähnlichkeit jedoch negativ beeinflussen können. Offen bleibt daher die Frage, welche der Eigenschaften eines Elementes im Speziellen bei der Erzeugung der Empfehlungen verwendet werden sollten. Ferner müsste auch untersucht werden, ob sich das Verfahren für andere Domänen als die hier verwendete als besser geeignet herausstellt.

Neben den bereits identifizierten Fragen deren Beantwortung eine weitere Evaluierung erfordert, bietet der Einsatz *semantischer* Technologien im Umfeld der *Recommender Systeme* zusätzliches Forschungspotential. So muss der subjektive Nutzen einer Empfehlung für einen Benutzer immer im Kontext betrachtet werden. Während die aktuell eingesetzten Verfahren der *Recommender Systeme* lediglich eindimensionale Bewertungen unterstützen, spielen die Umstände einer abgegebenen Bewertung und die darin enthaltenen Informationen bei der Generierung von Empfehlungen ebenfalls eine entscheidende Rolle. Der Ort, der Zeitpunkt oder auch das weitere Umfeld stellen maßgebliche Kriterien für die subjektive Qualität einer Empfehlung dar, welche durch den Einsatz *semantischer* Technologien dem System eindeutig verständlich gemacht werden können. Eine solche *Multi-Criteria-Recommendation* kann mit weiteren *semantische* Filterkriterien helfen, Elemente den Umständen entsprechend angepasst vorzuschlagen und die aktuellen Ziele und Wünsche genau zu formulieren.

Neben der genaueren Beschreibung der Vorlieben ermöglicht der Einsatz der Technologien und Methoden des *Semantic Webs* auch die Erzeugung und Aktualisierung einer lokalen Version des Profils durch den Benutzer selbst. Dies ist erst durch die global eindeutige Referenzierung sowie die Verwendung eines eindeutigen Vokabulars zur Beschreibung von Interessen sowie gewünschten Elementeneigenschaften realisierbar. Ein solches Profil kann vom Benutzer in verschiedenen *Online-Shops* oder Informationsportalen verwendet werden, so dass diesen eine umfassendere Abbildung der Vorlieben eines Benutzers zur Verfügung steht. Somit können diese anhand einer größeren Menge an Daten die Vorhersagen generieren, welche auf weit mehr Informationen basieren, als dies bei einer Erfassung in einem einzelnen Portal aktuell möglich ist. Erst solche verteilten und persönlichen Benutzerprofile ermöglichen eine *Cross-Site-Recommendation*, die einen entscheidenden Vorteil für den Benutzer bringt. Während er bei den aktuell eingesetzten Systemen jedem einzelnen von ihm erstmalig verwendeten *Recommender* mühsam von Neuem seine Vorlieben verständlich machen muss, braucht dies bei einem lokal gepflegten Profil nur einmalig geschehen.

Literaturverzeichnis

- [Achananuparp et al., 2007] Achananuparp, P., Han, H., Nasraoui, O., and Johnson, R. (2007). Semantically enhanced user modeling. In *Proceedings of the 2007 ACM symposium on Applied computing, SAC '07*, pages 1335–1339, New York, NY, USA. ACM.
- [Adomavicius and Tuzhilin, 2005] Adomavicius, G. and Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, 17(6):734–749.
- [Agrawal et al., 1993] Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD international conference on Management of data, SIGMOD '93*, pages 207–216, New York, NY, USA. ACM.
- [Alspector et al., 1997] Alspector, J., Kolcz, A., and Karunanithi, N. (1997). Feature-based and clique-based user models for movie selection: A comparative study. *User Model. User-Adapt. Interact.*, 7(4):279–304.
- [Anderson, 2004] Anderson, C. (2004). The long tail. *Wired*, 12(10).
- [Angeletou et al., 2007] Angeletou, S., Sabou, M., Specia, L., and Motta, E. (2007). Bridging the gap between folksonomies and the semantic web: An experience report. In *Workshop: Bridging the Gap between Semantic Web and Web 2.0, European Semantic Web Conference*, page 93.
- [Ardissono et al., 2001] Ardissono, L., Console, L., and Torre, I. (2001). An adaptive system for the personalized access to news. *AI Commun.*, 14:129–147.
- [Ardissono et al., 2004] Ardissono, L., Gena, C., Torasso, P., Bellifemine, F., Difino, A., and Negro, B. (2004). User modeling and recommendation techniques for personalized electronic program guides. In *Personalized Digital Television – Targeting Programs to Individual Viewers, volume 6 of Human-Computer Interaction Series, chapter 1*, pages 3–26. Kluwer Academic Publishers.
- [Ardissono et al., 1999] Ardissono, L., Goy, A., Meo, R., Petrone, G., Console, L., Lesmo, L., Simone, C., and Torasso, P. (1999). A configurable system for the construction of adaptive virtual stores. *World Wide Web*, 2:143–159.

- [Asnicar and Tasso, 1997] Asnicar, F. and Tasso, C. (1997). ifWeb: a Prototype of User Model-Based Intelligent Agent for Document Filtering and Navigation in the World Wide Web. In *Proc. of 6th International Conference on User Modelling*.
- [Auer et al., 2007] Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., and Ives, Z. (2007). Dbpedia: A nucleus for a web of open data. In *In 6th Int'l Semantic Web Conference, Busan, Korea*, pages 11–15. Springer.
- [Balabanovic and Shoham, 1997] Balabanovic, M. and Shoham, Y. (1997). Content-based, collaborative recommendation. *Commun. ACM*, 40(3):66–72.
- [Basu et al., 1998] Basu, C., Hirsh, H., and Cohen, W. (1998). Recommendation as classification: using social and content-based information in recommendation. In *Proceedings of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence, AAAI '98/IAAI '98*, pages 714–720, Menlo Park, CA, USA. American Association for Artificial Intelligence.
- [Baudisch and Brueckner, 2002] Baudisch, P. and Brueckner, L. (2002). Tv scout: Lowering the entry barrier to personalized tv program recommendation. In *In Proc. 2 nd Int. Conf. On Adaptive Hypermedia and Adaptive Web-based Systems (AH2002)*, pages 58–68. Springer.
- [Beckett and Berners-Lee, 2006] Beckett, D. and Berners-Lee, T. (2006). Turtle - Terse RDF Triple Language. Technical report, W3C. <http://www.w3.org/TeamSubmission/turtle/> Zugriff: 16. Juni 2011.
- [Beckett and Grant, 2004] Beckett, D. and Grant, J. (2004). RDF test cases. W3C recommendation, W3C. <http://www.w3.org/TR/2004/REC-rdf-testcases-20040210/> Zugriff: 16. Juni 2011.
- [Belkin and Croft, 1987] Belkin, N. J. and Croft, W. B. (1987). *Retrieval techniques*, pages 109–145. Elsevier Science Inc., New York, NY, USA.
- [Berners-Lee, 1994] Berners-Lee, T. (1994). Universal Resource Identifiers in WWW: A Unifying Syntax for the Expression of Names and Addresses of Objects on the Network as used in the World-Wide Web. RFC 1630 (Informational). <http://www.ietf.org/rfc/rfc1630.txt> Zugriff: 10. Juni 2011.
- [Berners-Lee et al., 2005] Berners-Lee, T., Fielding, R., and Masinter, L. (2005). Uniform Resource Identifier (URI): Generic Syntax. RFC 3986 (Standard). <http://www.ietf.org/rfc/rfc3986.txt> Zugriff: 10. Juni 2011.
- [Berners-Lee et al., 2001] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific American*, 284(5):34–43.
- [Berners-Lee et al., 1994] Berners-Lee, T., Masinter, L., and McCahill, M. (1994). Uniform Resource Locators (URL). RFC 1738 (Proposed Standard). <http://www.ietf.org/rfc/rfc1738.txt> Zugriff: 11. Juni 2011.

- [Billsus et al., 2002] Billsus, D., Brunk, C. A., Evans, C., Gladish, B., and Pazzani, M. (2002). Adaptive interfaces for ubiquitous web access. *Commun. ACM*, 45:34–38.
- [Billsus and Pazzani, 1998] Billsus, D. and Pazzani, M. J. (1998). Learning collaborative information filters. In *ICML*, pages 46–54.
- [Billsus and Pazzani, 2000] Billsus, D. and Pazzani, M. J. (2000). User modeling for adaptive news access. *User Model. User-Adapt. Interact.*, 10(2-3):147–180.
- [Bizer and Seaborne, 2004] Bizer, C. and Seaborne, A. (2004). D2RQ-treating non-RDF databases as virtual RDF graphs. In *Proceedings of the 3rd International Semantic Web Conference (ISWC2004)*. Citeseer.
- [Bloedorn et al., 1996] Bloedorn, E., Mani, I., and MacMillan, T. R. (1996). Machine learning of user profiles: representational issues. In *Proceedings of the thirteenth national conference on Artificial intelligence - Volume 1, AAAI'96*, pages 433–438. AAAI Press.
- [Bradley et al., 2000] Bradley, K., Rafter, R., and Smyth, B. (2000). Case-based user profiling for content personalisation. In *Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*, pages 62–72. Springer-Verlag.
- [Bradley and Smyth, 2001] Bradley, K. and Smyth, B. (2001). Improving Recommendation Diversity. In O'Donoghue, D., editor, *Proceedings of the 12th National Conference in Artificial Intelligence and Cognitive Science*, pages 75–84, Maynooth, Ireland.
- [Bray et al., 2008] Bray, T., Paoli, J., Maler, E., Yergeau, F., and Sperberg-McQueen, C. M. (2008). Extensible markup language (XML) 1.0 (fifth edition). W3C recommendation, W3C. <http://www.w3.org/TR/2008/REC-xml-20081126/> Zugriff: 10. Juni 2011.
- [Breese et al., 1998] Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In *Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence (UAI98)*, pages 43–52.
- [Brickley and Miller, 2007] Brickley, D. and Miller, L. (2007). The Friend Of A Friend (FOAF) vocabulary specification. <http://xmlns.com/foaf/spec/> Zugriff: 21. Juni 2011.
- [Bridge, 2001] Bridge, D. (2001). Product recommendation systems: A new direction. In *Workshop on CBR in Electronic Commerce at The International Conference on Case-Based Reasoning (ICCBR-01)*, pages 79–86.
- [Brusilovsky, 1996] Brusilovsky, P. (1996). Methods and techniques of adaptive hypermedia. *User Model. User-Adapt. Interact.*, 6(2-3):87–129.
- [Brusilovsky and Millán, 2007] Brusilovsky, P. and Millán, E. (2007). User Models for Adaptive Hypermedia and Adaptive Educational Systems. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, chapter 1, pages 3–53. Springer Berlin Heidelberg, Berlin, Heidelberg.

- [Burke, 1999] Burke, R. (1999). Integrating knowledge-based and collaborative-filtering recommender systems. In *In AAAI Workshop on AI in Electronic Commerce*, pages 69–72. AAAI.
- [Burke, 2000] Burke, R. (2000). Knowledge-based recommender systems. In *ENCYCLOPEDIA OF LIBRARY AND INFORMATION SYSTEMS*. Marcel Dekker.
- [Burke, 2002] Burke, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*, 12(4):331–370.
- [Burke, 2007] Burke, R. D. (2007). Hybrid web recommender systems. In *The Adaptive Web*, pages 377–408.
- [Candillier et al., 2007] Candillier, L., Meyer, F., and Boullé, M. (2007). Comparing State-of-the-Art Collaborative Filtering Systems. In *MLDM '07: Proceedings of the 5th international conference on Machine Learning and Data Mining in Pattern Recognition*, pages 548–562. Springer-Verlag, Berlin, Heidelberg.
- [Cantador et al., 2008] Cantador, I., Bellog, A., and Castells, P. (2008). Ontology-based personalised and context-aware recommendations of news items. *Web Intelligence and Intelligent Agent Technology, IEEE/WIC/ACM International Conference on*, 1:562–565.
- [Cantador et al., 2010] Cantador, I., Bellogín, A., and Vallet, D. (2010). Content-based recommendation in social tagging systems. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 237–240, New York, NY, USA. ACM.
- [Cantador and Castells, 2009] Cantador, I. and Castells, P. (2009). Semantic Contextualisation in a News Recommender System. In *Proceedings of the 1st International Workshop on Context-Aware Recommender Systems*.
- [Carroll and Klyne, 2004] Carroll, J. J. and Klyne, G. (2004). Resource description framework (RDF): Concepts and abstract syntax. W3C recommendation, W3C. <http://www.w3.org/TR/2004/REC-rdf-concepts-20040210/> Zugriff: 16. Juni 2011.
- [Castells et al., 2007] Castells, P., Fernandez, M., and Vallet, D. (2007). An adaptation of the vector-space model for ontology-based information retrieval. *IEEE Transactions on Knowledge and Data Engineering*, 19(2):261–272.
- [Cerf, 1969] Cerf, V. (1969). ASCII format for network interchange. RFC 20. <http://www.ietf.org/rfc/rfc20.txt> Zugriff: 13. Juni 2011.
- [Chedrawy and Abidi, 2009] Chedrawy, Z. and Abidi, S. S. (2009). A web recommender system for recommending, predicting and personalizing music playlists. In *Proceedings of the 10th International Conference on Web Information Systems Engineering*, WISE '09, pages 335–342, Berlin, Heidelberg. Springer-Verlag.
- [Chen and Sycara, 1998] Chen, L. and Sycara, K. (1998). Webmate: a personal agent for browsing and searching. In *Proceedings of the second international conference on Autonomous agents*, AGENTS '98, pages 132–139, New York, NY, USA. ACM.

- [Chua et al., 2010] Chua, A. M., Chua, R. C., Ang, A. V. D. T., Espiritua, J. L., Lim, N. R., and Cheng, D. (2010). Crowd sourcing through social gaming for community driven ontology engineering, results and observations. In *Proceedings of the ISWC Workshop*, pages 243–244.
- [Claypool et al., 1999] Claypool, M., Gokhale, A., Miranda, T., Murnikov, P., Netes, D., and Sartin, M. (1999). Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR Workshop on Recommender Systems*.
- [Cleverdon and Mills, 1966] Cleverdon and Mills, J. (1966). Factors determining the performance of indexing systems. *Volume I - Design, Volume II - Test Results, ASLIB Cranfield Project, Reprinted in Sparck Jones & Willett, Readings in Information Retrieval*.
- [Daudé et al., 2000] Daudé, J., Padró, L., Rigau, G., Llenguatges, D., and Informhtics, S. (2000). Mapping wordnets using structural information. In *In Proceedings 38 th Annual Meeting of the Association for Computational Linguistics(ACL00). Hong Kong*.
- [Deerwester et al., 1990] Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407.
- [Fallside, 2001] Fallside, D. C. (2001). XML schema part 0: Primer. first edition of a recommendation, W3C. <http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/> Zugriff: 19. Mai 2011.
- [Farsani and Nematbakhsh, 2006] Farsani, H. K. and Nematbakhsh, M. (2006). A semantic recommendation procedure for electronic product catalog. *International Journal of Applied Mathematics and Computer Sciences*, 3(2).
- [Fellbaum, 1998] Fellbaum, C., editor (1998). *WordNet: An Electronic Lexical Database (Language, Speech, and Communication)*. The MIT Press, illustrated edition edition.
- [Fernandez-Lopez et al., 1997] Fernandez-Lopez, M., Gómez-Pérez, A., and Juristo, N. (1997). Methontology: from ontological art towards ontological engineering. In *Proceedings of the AAAI97 Spring Symposium*, pages 33–40, Stanford, USA.
- [Fielding et al., 1999] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and Berners-Lee, T. (1999). Hypertext Transfer Protocol – HTTP/1.1. RFC 2616 (Draft Standard). <http://www.ietf.org/rfc/rfc2616.txt> Zugriff: 11. Juni 2011.
- [Finin et al., 2005] Finin, T. W., Mayfield, J., Joshi, A., Cost, R. S., and Fink, C. (2005). Information retrieval and the semantic web. In *HICSS*.
- [Fink and Kobsa, 2000] Fink, J. and Kobsa, A. (2000). A review and analysis of commercial user modeling servers for personalization on the world wide web. *User Model. User-Adapt. Interact.*, 10(2-3):209–249.

- [Fix and Hodges, 1951] Fix, E. and Hodges, J. (1951). Discriminatory analysis. nonparametric discrimination: consistency properties. Technical Report No. 4, USAF School of Aviation Medicine, Randolph Field, TX.
- [Furnas et al., 1987] Furnas, G. W., Landauer, T. K., Gomez, L. M., and Dumais, S. T. (1987). The vocabulary problem in human-system communication. *COMMUNICATIONS OF THE ACM*, 30(11):964–971.
- [Gauch et al., 2003] Gauch, S., Chaffee, J., and Pretschner, A. (2003). Ontology-Based User Profiles for Search and Browsing. *Web Intelligence and Agent Systems*, 1(3-4):219–234.
- [Gauch et al., 2007] Gauch, S., Speretta, M., Chandramouli, A., and Micarelli, A. (2007). User Profiles for Personalized Information Access. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, chapter 2, pages 54–89. Springer Berlin Heidelberg, Berlin, Heidelberg.
- [Gaudin, 2003] Gaudin, S. (2003). Study: personalization not secret to e-commerce. DATAMATION - EARTHWEB.
- [Ge et al., 2010] Ge, M., Delgado-Battenfeld, C., and Jannach, D. (2010). Beyond accuracy: evaluating recommender systems by coverage and serendipity. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 257–260, New York, NY, USA. ACM.
- [George, 2005] George, T. (2005). A scalable collaborative filtering framework based on clustering. In *Fifth IEEE International Conference on Data Mining*, pages 625–628.
- [Ghani and Fano, 2002] Ghani, R. and Fano, A. (2002). Building recommender systems using a knowledge base of product semantics. In *In 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems, Malaga*.
- [Goldberg et al., 1992] Goldberg, D., Nichols, D., Oki, B. M., and Terry, D. (1992). Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 35:61–70.
- [Golder and Huberman, 2006] Golder, S. A. and Huberman, B. A. (2006). Usage patterns of collaborative tagging systems. *J. Inf. Sci.*, 32:198–208.
- [Gómez-Pérez et al., 2003] Gómez-Pérez, A., Fernández-López, M., and Corcho, O. (2003). *Ontological Engineering*. Advanced Information and Knowledge Pro. Springer.
- [Gómez-Pérez et al., 2004] Gómez-Pérez, A., Fernández-López, M., and Corcho, O. (2004). *Ontological Engineering : with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web. First Edition (Advanced Information and Knowledge Processing)*. Springer.
- [Gonzalo et al., 1998] Gonzalo, J., Verdejo, F., Chugur, I., and Cigarran, J. (1998). Indexing with WordNet synsets can improve text retrieval. *Proceedings of the COLING/ACL Workshop on Usage of WordNet in Natural Language Processing Systems*.

- [Good et al., 1999] Good, N., Schafer, J. B., Konstan, J. A., Borchers, A., Sarwar, B., Herlocker, J., and Riedl, J. (1999). Combining collaborative filtering with personal agents for better recommendations. In *Proceedings of the sixteenth national conference on Artificial intelligence and the eleventh Innovative applications of artificial intelligence conference innovative applications of artificial intelligence*, AAAI '99/IAAI '99, pages 439–446, Menlo Park, CA, USA. American Association for Artificial Intelligence.
- [Goy et al., 2007] Goy, A., Ardissono, L., and Petrone, G. (2007). Personalization in e-commerce applications. In *The Adaptive Web*, pages 485–520.
- [Grudin, 1994] Grudin, J. (1994). Groupware and social dynamics: eight challenges for developers. *Commun. ACM*, 37:92–105.
- [Guarino, 1998] Guarino, N. (1998). *Formal Ontology in Information Systems: Proceedings of the 1st International Conference June 6-8, 1998, Trento, Italy*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 1st edition.
- [Guarino and Giaretta, 1995] Guarino, N. and Giaretta, P. (1995). Ontologies and knowledge bases - towards a terminological clarification. In Mars, N. J., editor, *Towards Very Large Knowledge Bases - Knowledge Building and Knowledge Sharing*, pages 25–32. IOS Press.
- [Guha and Brickley, 2004] Guha, R. V. and Brickley, D. (2004). RDF vocabulary description language 1.0: RDF schema. W3C recommendation, W3C. <http://www.w3.org/TR/2004/REC-rdf-schema-20040210/> Zugriff: 20. Juni 2011.
- [Gunawardana and Meek, 2008] Gunawardana, A. and Meek, C. (2008). Tied boltzmann machines for cold start recommendations. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 19–26, New York, NY, USA. ACM.
- [Ha, 2002] Ha, S. H. (2002). Helping online customers decide through web personalization. *IEEE Intelligent Systems*, 17:34–43.
- [Hassanzadeh and Consens, 2009] Hassanzadeh, O. and Consens, M. P. (2009). Linked Movie Data Base. In *Proceedings of the WWW2009 workshop on Linked Data on the Web (LDOW2009)*.
- [Hayes, 2004] Hayes, P. (2004). RDF semantics. W3C recommendation, W3C. <http://www.w3.org/TR/2004/REC-rdf-mt-20040210/> Zugriff: 16. Juni 2011.
- [Hayes et al., 2004] Hayes, P., Patel-Schneider, P. F., and Horrocks, I. (2004). OWL web ontology language semantics and abstract syntax. W3C recommendation, W3C. <http://www.w3.org/TR/2004/REC-owl-semantics-20040210/> Zugriff: 23. Juni 2011.
- [Hendler and Golbeck, 2008] Hendler, J. and Golbeck, J. (2008). Metcalfe's law, web 2.0, and the semantic web. *Web Semant.*, 6:14–20.
- [Herlocker et al., 1999] Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. (1999). An algorithmic framework for performing collaborative filtering. In *SIGIR*, pages 230–237.

- [Herlocker et al., 2000] Herlocker, J. L., Konstan, J. A., and Riedl, J. (2000). Explaining collaborative filtering recommendations. In *CSCW*, pages 241–250.
- [Herlocker et al., 2002] Herlocker, J. L., Konstan, J. A., and Riedl, J. (2002). An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms. *Inf. Retr.*, 5(4):287–310.
- [Herlocker et al., 2004] Herlocker, J. L., Konstan, J. A., Terveen, L. G., and Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22:5–53.
- [Hill et al., 1995] Hill, W., Stead, L., Rosenstein, M., and Furnas, G. (1995). Recommending and evaluating choices in a virtual community of use. In *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI '95*, pages 194–201, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- [Hirst and St-Onge, 1998] Hirst, G. and St-Onge, D. (1998). Lexical chains as representations of context for the detection and correction of malapropisms. In Fellbaum, C., editor, *WordNet: An Electronic Lexical Database*, pages 305–332. MIT Press.
- [Höök, 1997] Höök, K. (1997). Evaluating the utility and usability of an adaptive hypermedia system. In *Proceedings of the 2nd international conference on Intelligent user interfaces, IUI '97*, pages 179–186, New York, NY, USA. ACM.
- [Hof et al., 1998] Hof, R., Green, H., and Himmelstein, L. (1998). Now its your web. *Business Week*, pages 68–74.
- [Jiang and Conrath, 1997] Jiang, J. J. and Conrath, D. W. (1997). Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR*, cmp-lg/9709008.
- [Kaplan et al., 1993] Kaplan, C., Fenwick, J., and Chen, J. (1993). Adaptive hypertext navigation based on user goals and context. *User Modeling and User-Adapted Interaction*, 3(3):193–220.
- [Kazienko and Kolodziejcki, 2006] Kazienko, P. and Kolodziejcki, P. (2006). Personalized integration of recommendation methods for e-commerce. *IJCSA*, 3(3):12–26.
- [Konstan et al., 1997] Konstan, J. A., Miller, B. N., Maltz, D., Herlocker, J. L., Gordon, L. R., and Riedl, J. (1997). GroupLens: applying collaborative filtering to Usenet news. *Communications of the ACM*, 40(3):77–87.
- [Konstan et al., 1998] Konstan, J. A., Riedl, J., Borchers, A., and Herlocker, J. L. (1998). Recommender Systems: A GroupLens Perspective. In *Recommender Systems. Papers from 1998 Workshop. Technical Report WS-98*, volume 8, pages 60–64.
- [Krulwich, 1997] Krulwich, B. (1997). Lifestyle finder: Intelligent user profiling using large-scale demographic data. *AI Magazine*, 18(2):37–45.

- [Kunze and Baker, 2007] Kunze, J. and Baker, T. (2007). The Dublin Core Metadata Element Set. RFC 5013 (Informational). <http://www.ietf.org/rfc/rfc5013.txt> Zugriff: 14. August 2011.
- [Kurapati and Gutta, 2002] Kurapati, K. and Gutta, S. (2002). Instant personalization via clustering tv viewing patterns. In *Proceedings of the Sixth International Conference on Artificial Intelligence and Soft Computing, Banff, Canada*.
- [Labrou and Finin, 1999] Labrou, Y. and Finin, T. (1999). Yahoo! as an ontology: using yahoo! categories to describe documents. In *Proceedings of the eighth international conference on Information and knowledge management, CIKM '99*, pages 180–187, New York, NY, USA. ACM.
- [Lang, 1995] Lang, K. (1995). Newsweeder: Learning to filter netnews. In *ICML*, pages 331–339.
- [Leacock and Chodorow, 1998] Leacock, C. and Chodorow, M. (1998). Combining local context and wordnet similarity for word sense identification. In Fellbaum, C., editor, *MIT Press*, pages 265–283, Cambridge, Massachusetts.
- [Liao et al., 2006] Liao, I.-E., Liao, S.-C., Kao, K.-F., and Harn, I.-F. (2006). A personal ontology model for library recommendation system. In Sugimoto, S., Hunter, J., Rauber, A., and Morishima, A., editors, *Digital Libraries: Achievements, Challenges and Opportunities*, volume 4312 of *Lecture Notes in Computer Science*, pages 173–182. Springer Berlin / Heidelberg.
- [Lin et al., 2002] Lin, W., Alvarez, S. A., and Ruiz, C. (2002). Efficient adaptive-support association rule mining for recommender systems. *Data Min. Knowl. Discov.*, 6:83–105.
- [Linden et al., 2003] Linden, G., Smith, B., and York, J. (2003). Amazon.com recommendations: item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80.
- [Lémdani et al., 2011] Lémdani, R., Polailon, G., Bennacer, N., and Bourda, Y. (2011). A semantic similarity measure for recommender systems. In *I-SEMANTICS*.
- [Lowe, 2004] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60:91–110.
- [Magnini and Strapparava, 2004] Magnini, B. and Strapparava, C. (2004). User modelling for news web sites with word sense based techniques. *User Modeling and User-Adapted Interaction*, 14:239–257.
- [Maltz and Ehrlich, 1995] Maltz, D. and Ehrlich, K. (1995). Pointing the way: Active collaborative filtering. In *CHI*, pages 202–209.
- [Mathes, 2004] Mathes, A. (2004). Folksonomies - cooperative classification and communication through shared metadata.

- [Menzel et al., 1993] Menzel, C. P., Mayer, R. J., Painter, M. K., and (U.S.), A. L. (1993). *IDEF5 ontology description capture method*. Armstrong Laboratory, Air Force Materiel Command, Wright-Patterson Air Force Base, Ohio .:
- [Metcalf, 1995] Metcalfe, R. (1995). Metcalfe’s law: A network becomes more valuable as it reaches more users. *Infoworld*, 17.
- [Micarelli and Sciarrone, 2004] Micarelli, A. and Sciarrone, F. (2004). Anatomy and empirical evaluation of an adaptive web-based information filtering system. *User Modeling and User-Adapted Interaction*, 14:159–200.
- [Middleton et al., 2002] Middleton, S. E., Alani, H., and Roure, D. C. D. (2002). Exploiting synergy between ontologies and recommender systems. In *IN PROCEEDINGS OF THE WWW2002 INTERNATIONAL WORKSHOP ON THE SEMANTIC WEB (MAUI)*.
- [Middleton et al., 2004] Middleton, S. E., Shadbolt, N. R., and Roure, D. C. D. (2004). Ontological user profiling in recommender systems. *ACM Transactions on Information Systems*, 22:54–88.
- [Miller et al., 2003] Miller, B. N., Albert, I., Lam, S. K., Konstan, J. A., and Riedl, J. (2003). Movielens unplugged: Experiences with a recommender system on four mobile devices. In *In Proceedings of the 2003 Conference on Intelligent User Interfaces*.
- [Miller, 1995] Miller, G. (1995). Wordnet A lexical database for English. *Communications of ACM*, 38(11):39–41.
- [Miller, 1968] Miller, R. B. (1968). Response time in man-computer conversational transactions. In *Proceedings of the December 9-11, 1968, fall joint computer conference, part I*, AFIPS ’68 (Fall, part I), pages 267–277, New York, NY, USA. ACM.
- [Mitra and Lafon, 2007] Mitra, N. and Lafon, Y. (2007). SOAP version 1.2 part 0: Primer (second edition). Technical report, W3C. <http://www.w3.org/TR/2007/REC-soap12-part0-20070427/> Zugriff: 19. Juni 2011.
- [Moats, 1997] Moats, R. (1997). URN Syntax. RFC 2141 (Proposed Standard). <http://www.ietf.org/rfc/rfc2141.txt> Zugriff: 11. Juni 2011.
- [Mobasher et al., 2003] Mobasher, B., Jin, X., and Zhou, Y. (2003). Semantically enhanced collaborative filtering on the web. In *EWMMF*, pages 57–76.
- [Montoyo et al., 2001] Montoyo, A., Palomar, M., and Rigau, G. (2001). Lexical enrichment of wordnet with classification systems using specification marks method. In *NLDB*, pages 109–119.
- [Mooney and Roy, 2000] Mooney, R. J. and Roy, L. (2000). Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, DL ’00, pages 195–204, New York, NY, USA. ACM.

- [Morita and Shinoda, 1994] Morita, M. and Shinoda, Y. (1994). Information filtering based on user behavior analysis and best match text retrieval. In *Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '94, pages 272–281, New York, NY, USA. Springer-Verlag New York, Inc.
- [Motik et al., 2009] Motik, B., Parsia, B., and Patel-Schneider, P. F. (2009). OWL 2 web ontology language structural specification and functional-style syntax. W3C recommendation, W3C. <http://www.w3.org/TR/2009/REC-owl2-syntax-20091027/> Zugriff: 23. Juni 2011.
- [Moukas, 1997] Moukas, A. (1997). Amalthea information discovery and filtering using a multiagent evolving ecosystem. *Applied Artificial Intelligence*, 11(5):437–457.
- [Musial et al., 2008] Musial, K., Juszczyszyn, K., and Kazienko, P. (2008). Ontology-based recommendation in multimedia sharing systems. *System Science*, 1(34):97–106.
- [Navigli and Velardi, 2003] Navigli, R. and Velardi, P. (2003). An analysis of ontology-based query expansion strategies. In *Workshop on Adaptive Text Extraction and Mining, (Cavtat Dubrovnik, Croatia, Sept 23)*.
- [Nguyen et al., 2007] Nguyen, A.-T., Denos, N., and Berrut, C. (2007). Improving new user recommendations with rule-based induction on cold user data. In *Proceedings of the 2007 ACM conference on Recommender systems*, RecSys '07, pages 121–128, New York, NY, USA. ACM.
- [Nichols, 1998] Nichols, D. M. (1998). Implicit rating and filtering. In *In Proceedings of the Fifth DELOS Workshop on Filtering and Collaborative Filtering*, pages 31–36.
- [Nielsen, 1994] Nielsen, J. (1994). *Usability Engineering*. Morgan Kaufmann, San Francisco.
- [Nielsen and Loranger, 2006] Nielsen, J. and Loranger, H. (2006). *Prioritizing Web Usability*. New Riders, Berkeley, CA.
- [Oard and Kim, 1998] Oard, D. and Kim, J. (1998). Implicit feedback for recommender systems. In *in Proceedings of the AAAI Workshop on Recommender Systems*, pages 81–83.
- [Oldakowski and Bizer, 2005] Oldakowski, R. and Bizer, C. (2005). SemMF: A Framework for Calculating Semantic Similarity of Objects Represented as RDF Graphs. *Poster at the 4th International Semantic Web Conference (ISWC 2005)*.
- [Oppermann, 1994] Oppermann, R., editor (1994). *Adaptive user support: ergonomic design of manually and automatically adaptable software*. L. Erlbaum Associates Inc., Hillsdale, NJ, USA.
- [Orio, 2006] Orio, N. (2006). Music retrieval: a tutorial and review. *Found. Trends Inf. Retr.*, 1:1–96.
- [Pareto, 1896] Pareto, V. (1896). *Cours d'économie politique*. Droz, Geneva.

- [Park and Tuzhilin, 2008] Park, Y.-J. and Tuzhilin, A. (2008). The long tail of recommender systems and how to leverage it. In *Proceedings of the 2008 ACM conference on Recommender systems*, RecSys '08, pages 11–18, New York, NY, USA. ACM.
- [Passant, 2010] Passant, A. (2010). dbrec - music recommendations using dbpedia. In Patel-Schneider, P. F., Pan, Y., Hitzler, P., Mika, P., 0007, L. Z., Pan, J. Z., Horrocks, I., and Glimm, B., editors, *International Semantic Web Conference (2)*, volume 6497 of *Lecture Notes in Computer Science*, pages 209–224. Springer.
- [Pazzani, 1999] Pazzani, M. J. (1999). A framework for collaborative, content-based and demographic filtering. *Artif. Intell. Rev.*, 13(5-6):393–408.
- [Pazzani and Billsus, 1997] Pazzani, M. J. and Billsus, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331.
- [Pazzani et al., 1996] Pazzani, M. J., Muramatsu, J., and Billsus, D. (1996). Syskill & webert: Identifying interesting web sites. In *AAAI/IAAI, Vol. 1*, pages 54–61.
- [Pearson, 1895] Pearson, K. (1895). Royal society proceedings, 58, pages 241.
- [Peis et al., 2008] Peis, E., del Castillo, J. M. M., and Delgado-López, J. A. (2008). Semantic recommender systems. analysis of the state of the topic. *Hipertext.net*, 6:(online).
- [Peppers and Rogers, 1997] Peppers, D. and Rogers, M. (1997). *The one to one future : building relationships one customer at a time*. Bantam Doubleday Dell Publishing.
- [Pine, 1995] Pine, B. J. (1995). *Do you want to keep your Customers forever?* Harvard Business School Review.
- [Pine, 1999] Pine, B. J. (1999). *Mass Customization: The New Frontier in Business Competition*. Harvard Business School Press.
- [Porter, 1980] Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.
- [Prud'hommeaux and Seaborne, 2008] Prud'hommeaux, E. and Seaborne, A. (2008). SPARQL query language for RDF. W3C recommendation, W3C. <http://www.w3.org/TR/2008/REC-rdf-sparql-query-20080115/> Zugriff: 25. Juni 2011.
- [Quiroga and Mostafa, 1999] Quiroga, L. M. and Mostafa, J. (1999). Empirical evaluation of explicit versus implicit acquisition of user profiles in information filtering systems. In *Proceedings of the fourth ACM conference on Digital libraries*, DL '99, pages 238–239, New York, NY, USA. ACM.
- [Rashid et al., 2002] Rashid, A. M., Albert, I., Cosley, D., Lam, S. K., McNee, S. M., Konstan, J. A., and Riedl, J. (2002). Getting to know you: learning new user preferences in recommender systems. In *Proceedings of the 7th international conference on Intelligent user interfaces*, IUI '02, pages 127–134, New York, NY, USA. ACM.

- [Reichheld and Sasser, 1990] Reichheld, F. F. and Sasser, W. E. (1990). Zero defections: Quality comes to services. *Harvard Business Review*.
- [Resnick et al., 1994] Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J. (1994). Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW*, pages 175–186.
- [Rich, 1979] Rich, E. (1979). Building and exploiting user models. In *Proceedings of the 6th international joint conference on Artificial intelligence - Volume 2*, pages 720–722, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Rohland and Streibel, 2009] Rohland, M. and Streibel, O. (2009). Algorithmic Extraction of Tag Semantics. Technical report, Networked Information Systems, Free University Berlin.
- [Saaty, 1980] Saaty, T. (1980). *The Analytic Hierarchy Process, Planning, Priority Setting, Resource Allocation*. McGraw-Hill, New York.
- [Salton, 1989] Salton, G. (1989). *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley.
- [Salton et al., 1975] Salton, G., Wong, A., and Yang, C. S. (1975). A vector space model for automatic indexing. *Commun. ACM*, 18:613–620.
- [Sarwar et al., 2001] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2001). Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web, WWW '01*, pages 285–295, New York, NY, USA. ACM.
- [Sarwar et al., 2000a] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2000a). Analysis of recommendation algorithms for e-commerce. In *Proceedings of the 2nd ACM conference on Electronic commerce, EC '00*, pages 158–167, New York, NY, USA. ACM.
- [Sarwar et al., 2002] Sarwar, B., Karypis, G., Konstan, J., and Riedl, J. (2002). Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth International Conference on Computer and Information Science*, pages 27–28.
- [Sarwar et al., 2000b] Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J. T. (2000b). Application of dimensionality reduction in recommender systems—a case study. In *ACM WebKDD Workshop*.
- [Schafer et al., 2007] Schafer, J. B., Frankowski, D., Herlocker, J. L., and Sen, S. (2007). Collaborative filtering recommender systems. In *The Adaptive Web*, pages 291–324.
- [Schafer et al., 1999] Schafer, J. B., Konstan, J. A., and Riedl, J. (1999). Recommender systems in e-commerce. In *ACM Conference on Electronic Commerce*, pages 158–166.
- [Schafer et al., 2001] Schafer, J. B., Konstan, J. A., and Riedl, J. (2001). E-commerce recommendation applications. *Data Min. Knowl. Discov.*, 5:115–153.

- [Schickel-Zuber and Faltings, 2006] Schickel-Zuber, V. and Faltings, B. (2006). Inferring user’s preferences using ontologies. In *AAAI*.
- [Schmid and Mohr, 1997] Schmid, C. and Mohr, R. (1997). Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:530–535.
- [Schwab et al., 2001] Schwab, I., Kobsa, A., and Koychev, I. (2001). Learning user interests through positive examples using content analysis and collaborative filtering. In *30 2001. Internal Memo, GMD*.
- [Shani and Gunawardana, 2011] Shani, G. and Gunawardana, A. (2011). Evaluating Recommendation Systems. In Ricci, F., Rokach, L., Shapira, B., and Kantor, P. B., editors, *Recommender Systems Handbook*, chapter 8, pages 257–297. Springer US, Boston, MA.
- [Shardanand and Maes, 1995] Shardanand, U. and Maes, P. (1995). Social information filtering: algorithms for automating word of mouth. In *Proceedings of the SIGCHI conference on Human factors in computing systems, CHI ’95*, pages 210–217, New York, NY, USA. ACM Press/Addison-Wesley Publishing Co.
- [Shimazu, 2002] Shimazu, H. (2002). Expertclerk: A conversational case-based reasoning tool for developing salesclerk agents in e-commerce webshops. *Artif. Intell. Rev.*, 18(3-4):223–244.
- [Sieg et al., 2004] Sieg, A., Mobasher, B., and Burke, R. (2004). Inferring user’s information context from user profiles and concept hierarchies. In *In: Proceedings of the 2004 Meeting of the International Federation of Classification Societies, IFCS 2004*, pages 563–574.
- [Simperl et al., 2009] Simperl, E. P. B., Mochól, M., Bürger, T., and Popov, I. O. (2009). Achieving maturity: The state of practice in ontology engineering in 2009. In Meersman, R., Dillon, T. S., and Herrero, P., editors, *OTM Conferences (2)*, volume 5871 of *Lecture Notes in Computer Science*, pages 983–991. Springer.
- [Simperl et al., 2006] Simperl, E. P. B., Tempich, C., and Sure, Y. (2006). Ontocom: A cost estimation model for ontology engineering. In *Proceedings of fifth ISWC*.
- [Siorpaes and Hepp, 2008] Siorpaes, K. and Hepp, M. (2008). Ontogame: Weaving the semantic web by online gaming. In Hauswirth, M., Koubarakis, M., and Bechhofer, S., editors, *Proceedings of the 5th European Semantic Web Conference*, LNCS, Berlin, Heidelberg. Springer Verlag.
- [Smyth, 2007] Smyth, B. (2007). The adaptive web. In Brusilovsky, P., Kobsa, A., and Nejdl, W., editors, *The adaptive Web*, chapter Case-based recommendation, pages 342–376. Springer-Verlag, Berlin, Heidelberg.
- [Smyth and McClave, 2001] Smyth, B. and McClave, P. (2001). Similarity vs. diversity. In *Proceedings of the 4th International Conference on Case-Based Reasoning: Case-Based Reasoning Research and Development, ICCBR ’01*, pages 347–361, London, UK. Springer-Verlag.

- [Stenback and Heninger, 2004] Stenback, J. and Heninger, A. (2004). Document object model (DOM) level 3 load and save specification. W3C recommendation, W3C. <http://www.w3.org/TR/2004/REC-DOM-Level-3-LS-20040407> Zugriff: 11. Juni 2011.
- [Studer et al., 1998] Studer, R., Benjamins, R., and Fensel, D. (1998). Knowledge Engineering: Principles and Methods. *Data and Knowledge Engineering*, 25:161–197.
- [Sure et al., 2002] Sure, Y., Staab, S., and Studer, R. (2002). Methodology for development and employment of ontology based knowledge management applications. *SIGMOD Rec.*, 31(4):18–23.
- [Szomszor et al., 2007] Szomszor, M., Cattuto, C., Alani, H., O’Hara, K., Baldassarri, A., Loreto, V., and Servedio, V. D. (2007). Folksonomies, the semantic web, and movie recommendation. In *4th European Semantic Web Conference, Bridging the Gap between Semantic Web and Web 2.0*.
- [Tam and Ho, 2003] Tam, K. Y. and Ho, S. Y. (2003). Web personalization: Is it effective? *IT Professional*, 5:53–57.
- [Tan et al., 1998] Tan, A., Teo, C., and Keng, H. M. (1998). Learning user profiles for personalized information dissemination. In *In proceedings, International Joint Conference on Neural Networks (IJCNN’98)*, pages 183–188. IEEE.
- [Tang and Mccalla, 2003] Tang, T. Y. and Mccalla, G. (2003). Mining implicit ratings for focused collaborative filtering for paper recommendations. In *in Workshop on User and Group Models for Web-based Adaptive Collaborative Environments, 9th International Conference on User Modeling (UM 2003)*, pages 45–56.
- [Tanudjaja and Mui, 2002] Tanudjaja, F. and Mui, L. (2002). Persona: A contextualized and personalized web search. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences (HICSS’02)-Volume 3 - Volume 3*, pages 67–, Washington, DC, USA. IEEE Computer Society.
- [Tempich et al., 2007] Tempich, C., Simperl, E. P. B., Luczak, M., Studer, R., and Pinto, H. S. (2007). Argumentation-based ontology engineering. *IEEE Intelligent Systems*, 22(6):52–59.
- [Terveen et al., 1997] Terveen, L., Hill, W., Amento, B., McDonald, D., and Creter, J. (1997). Phoaks: A system for sharing recommendations. *Communications of the ACM*, 40(3):59–62.
- [Towle and Quinn, 2000] Towle, B. and Quinn, C. (2000). Knowledge based recommender systems using explicit user models. In *2003c). Clustering, part II. Retrieved December 27*, pages 00–04. AAAI Press.
- [Tsiriga and Virvou, 2003] Tsiriga, V. and Virvou, M. (2003). Modelling the student to individualise tutoring in a web-based icall. *International Journal of Continuing Engineering Education and Lifelong Learning*, 6(13):350–365.

- [Uschold and Gruninger, 1996] Uschold, M. and Gruninger, M. (1996). Ontologies: Principles, methods and applications. *Knowledge Engineering Review*, 11:93–136.
- [van Rijsbergen, 1979] van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworth.
- [Vikas and Carly, 2010] Vikas, M. and Carly, F. (2010). Customer satisfaction: A strategic review and guidelines for managers. Technical Report 10-701, Marketing Science Institute: MSI Fast Forward.
- [Vrandečić et al., 2005] Vrandečić, D., Pinto, H. S., Sure, Y., and Tempich, C. (2005). The diligent knowledge processes. *Journal of Knowledge Management*, 9(5):85–96.
- [Wang et al., 2003] Wang, B. B., McKay, R. I. B., Abbass, H. A., and Barlow, M. (2003). A comparative study for domain ontology guided feature extraction. In *Proceedings of the 26th Australasian computer science conference - Volume 16, ACSC '03*, pages 69–78, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.
- [Wang et al., 2008] Wang, Y., Stash, N., Aroyo, L., Gorgels, P., Rutledge, L., and Schreiber, G. (2008). Recommendations based on semantically enriched museum collections. *J. Web Sem.*, 6(4):283–290.
- [Wang et al., 2009] Wang, Y., Stash, N., Aroyo, L., Hollink, L., and Schreiber, G. (2009). Semantic relations for content-based recommendations. In *Proceedings of the fifth international conference on Knowledge capture, K-CAP '09*, pages 209–210, New York, NY, USA. ACM.
- [Widyantoro et al., 1999] Widyantoro, D. H., Yin, J., Nasr, M. S. E., Seif, M., Nasr, E., Yang, L., Zacchi, A., and Yen, J. (1999). Alipes: A swift messenger in cyberspace. In *In Proc. AAAI Spring Symposium on Intelligent Agents in Cyberspace (Palo Alto)*, pages 62–67. AAAI Press.
- [Wu and Palmer, 1994] Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics, ACL '94*, pages 133–138, Stroudsburg, PA, USA. Association for Computational Linguistics.
- [Yang and Powers, 2005] Yang, D. and Powers, D. M. W. (2005). Measuring semantic similarity in the taxonomy of wordnet. In *Proceedings of the 28th Australasian Computer Science Conference*, pages 315–322.
- [Yang and Liu, 1999] Yang, Y. and Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, SIGIR '99*, pages 42–49, New York, NY, USA. ACM.
- [Yang and Pedersen, 1997] Yang, Y. and Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. In *Proceedings of the Fourteenth International Conference on Machine Learning, ICML '97*, pages 412–420, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- [Yergeau, 2003] Yergeau, F. (2003). UTF-8, a transformation format of ISO 10646. RFC 3629 (Standard). <http://www.ietf.org/rfc/rfc3629.txt> Zugriff: 13. Juni 2011.

- [Yildiz, 2006] Yildiz, B. (2006). Ontology evolution and versioning the state of the art.
- [Zanker et al., 2006] Zanker, M., Bricman, M., Gordea, S., Jannach, D., and Jessenitschnig, M. (2006). Persuasive online-selling in quality and taste domains. In *EC-Web*, pages 51–60.
- [Zanker and Jessenitschnig, 2009] Zanker, M. and Jessenitschnig, M. (2009). Collaborative feature-combination recommender exploiting explicit and implicit user feedback. In *Proceedings of the 2009 IEEE Conference on Commerce and Enterprise Computing*, pages 49–56, Washington, DC, USA. IEEE Computer Society.
- [Ziegler et al., 2004a] Ziegler, C.-N., Lausen, G., and Schmidt-Thieme, L. (2004a). Taxonomy-driven computation of product recommendations. In *CIKM*, pages 406–415.
- [Ziegler et al., 2004b] Ziegler, C.-N., Schmidt-Thieme, L., and Lausen, G. (2004b). Exploiting semantic product descriptions for recommender systems. In *Proceedings of the 2nd ACM SIGIR Semantic Web and Information Retrieval Workshop*, Sheffield, UK.

