Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

SUSANNE GOTTWALD

Recommender Systeme fuer den Einsatz in Bibliotheken
Survey on recommender systems

ZIB-Report 11-30 (July 2011)

Recommender Systeme für den Einsatz in Bibliotheken
Survey on recommender systems

Susanne Gottwald (gottwald@zib.de)
Konrad-Zuse-Zentrum für
Informationstechnik Berlin
Unit Scientific Computing
Scientific Information
Takustr. 7, 14195 Berlin
www.zib.de

Abstract: In diesem Report wird ein Überblick über Recommender Systeme (RS) im Bibliotheksbereich gegeben, welche Eigenschaften RS haben und welche Bedeutung sie damit für Bibliotheken besitzen. Es existieren semantische und nicht-semantische Lösungen, die dem Benutzer aus einer Menge von Publikationen die für ihn relevantesten empfehlen. Diese Arbeiten sowie eine Einleitung in das Thema werden präsentiert und mit den gewonnenen Kenntnissen ein Ausblick auf mögliche Entwicklungen gegeben.

1 Motivation

„Das möglichst umfassende Auffinden und Erschließen relevanter Informationen ist Wahrzeichen der Wissenschaft. [..] Die ständige Berücksichtigung und Auseinandersetzung mit wissenschaftlicher Literatur unterscheidet wissenschaftliches vom journalistischen Schreiben [..] Recherche und Beschäftigung mit wissenschaftlicher Literatur dient der fortschreitenden Erkenntnis ebenso wie dem Ausweis der Sachkenntnis [..]“ [Ben04]

Unter dieser Voraussetzung sehen sich Informationssysteme und Bibliotheken im Speziellen mit einer großen Herausforderung konfrontiert: effektives Management großer Dokumenten-Sammlungen, um Benutzern die Möglichkeit zu geben, leicht und schnell auf gewünschte Informationen zugreifen zu können.[PdCDL08] Analog zu [Ova] möchte ein Wissenschaftler bei einer expliziten Literaturrecherche ein bestimmtes Literaturbedürfnis befriedigen, wie in Abbildung 1 zu sehen ist.

Dabei ist die Art des Bedürfnisses vom Zeitpunkt der Recherche abhängig. Bei einem allgemeinen Interesse stehen die Überraschung und der Zufall im Vordergrund (0), der Wissenschaftler verfolgt lose Links, die ihm interessant erscheinen. Steht ein Wissenschaftler am Anfang einer Arbeit, eines Vorhabens oder einer Recherche (1), muss er zunächst den Titel des Forschungsgebiets herausfinden, sich einen Überblick über das Thema verschaffen und die relevanten Begriffe identifizieren, um im nächsten Schritt (2) konkrete Informationen zu finden. Befindet sich der Wissenschaftler dann in einer laufenden Forschungsarbeit (3), ist es essentiell, auf dem Gebiet dieser Arbeit ständig auf dem Laufen-

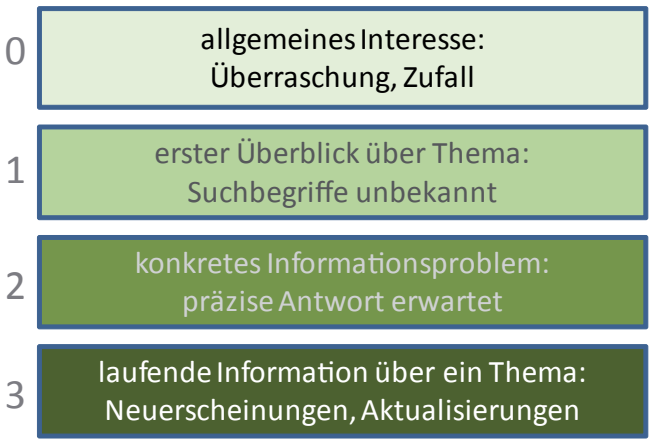

Abbildung 1: Verschiedene Literaturbedürfnisse

den zu bleiben und Antworten zu finden auf Fragen wie

- Welche Wissenschaftler forschen auf demselben Gebiet?
- Welche aktuellen Ergebnisse werden in Publikationen beschrieben?
- Welche Publikationen sollten im Forschungsvorhaben berücksichtigt werden?

Außerdem hat der Wissenschaftler das Problem, dass neuere Publikationen durchaus einen erweiterten Wortschatz benutzen können. Das kann dazu führen, dass diese Arbeiten innerhalb eines Informationssystems und früheren bekannten Suchbegriffen nicht gefunden werden können oder nur sehr weit hinten im Ergebnisranking auftauchen. Parallel dazu kann es auch vorkommen, dass dasselbe Problem oder Thema mit unterschiedlichen Bezeichnungen vorhanden ist, was das Finden von passender Literatur erneut erschwert.

An dieser Stelle (3) können dem Wissenschaftler Recommender Systeme helfen, die ihm neben einer expliziten Anfrage auch fortlaufend und ohne erneute Suche relevante Publikationen empfehlen. Sie sollen dem Wissenschaftler vor allem die Zeit der Suche ersparen und ihm auf seinem Gebiet einen Informationsvorsprung geben, in dem er sofort nach Veröffentlichung von neuen und passenden Publikationen erfährt. Dabei profitiert dieser nicht nur während einer laufenden Forschungstätigkeit von einem solchen System, sondern auch im sonstigen Arbeitsalltag. Oftmals schafft der Wissenschaftler es zeitlich gar nicht, die interessanten Zeitschriften nach neuen Artikeln durchzusehen. Oder interessante Artikel werden in Zeitschriften veröffentlicht, die unbekannt sind oder in denen man die Veröffentlichung nicht erwartet. Diesen Problemen wird begegnet, in dem dem Wissenschaftler ohne eigenes Zutun interessante Literatur vorgestellt wird, was vielfach auch neue Ideen und Ansätze für künftige Projekte mit sich bringt.

Recommender Systeme können als Service in Bibliotheken eingesetzt werden. Wenn in einer Bibliothek neue Publikationen herauskommen bzw. in den Bestand eingegliedert

werden, werden die Benutzer der Bibliothek sofort benachrichtigt und können sich für eine Ausleihe an die Bibliothek wenden.

1.1 Bedeutung von RS für Bibliotheken

Die Autoren in [GSNT03] schreiben, dass viele kommerzielle digitale Bibliotheken untereinander mit mehr oder weniger sinnvollen Services konkurrieren und dass onlineverfügbare Bibliotheken faktisch nicht überleben können, wenn sie solche Services nicht anbieten. Desweiteren siehen die Autoren Recommender Systeme als sehr viel versprechende Erweiterung für traditionelle Bibliotheken an. Ihre Notwendigkeit ergibt sich aus dem Bedürfnis von Wissenschaftlern und Studenten nach einer effizienten Literaturrecherche. Die Autoren nennen als Beleg die Ergebnisse einer im Auftrag des BMBF durchgeführten Studie von Klatt[1] über die Verwendung von elektronischen wissenschaftlichen Artikeln in der universitären Ausbildung. Sie zeigt, dass zwar drei Viertel aller Studierenden elektronische Literaturrecherchen als sehr wichtig einschätzen, mehr als sechzig Prozent von ihnen aber dennoch in erster Linie ihre Kommilitonen um Empfehlungen bitten. Qualifizierte Empfehlungen spielen also im universitären und wissenschaftlichen Umfeld eine bedeutende Rolle, sie verschaffen den Personen eine Zeitersparnis sowie einen Wissensvorsprung, den sie nur durch eine Recherche oder auch gar nicht erlangt hätten.
Als Gründe, warum wissenschaftliche Bibliotheken diesen Service bisher nicht breiter anbieten, werden in [GSNT03] folgende drei Ursachen genannt:

- Privatsphäre: Bibliothekare möchten die Privatsphäre ihrer Benutzer ernst nehmen und dass deren Daten (Benutzerverhalten, Ausleihlisten) geschützt werden.
- Finanzielle Einschränkungen: Öffentliche Bibliotheken hätten ein enges Budget, das Investitionen in neue elektronische Services für unter Umständen Millionen von Benutzern nicht einfach so zulässt.
- Datenumfang: Die Anzahl der Dokumente in öffentlichen oder akademischen Bibliotheken sei oftmals um ein Vielfaches höher als bei kommerziellen Organisationen. Viele Daten erhöhen die Qualität einer Empfehlung aber gleichzeitig auch deren Berechnungskomplexität.

In [MS08] wird ein RS als eine Art Katalog-Erweiterung oder Ersatz für traditionelle Klassifizierung zu Fachgebieten angesehen. Darüber hinaus könnte solch ein System die Bibliothekare darin unterstützen, den Bestand der Bibliothek auf einem aktuellen Stand zu halten. Wichtig ist, dass solch ein System entsprechend mit den sensiblen Daten der Benutzer umgeht, möglichst kostenlos als Open-Source-Software verfügbar ist und auch für große Datenmengen skaliert.

[1] Rüdiger Klatt, Konstantin Gavriilidis, Kirsten Kleinsimlinghaus, and Maresa Feldmann: Nutzung und Potenziale der innovativen Mediennutzung im Lernalltag der Hochschulen, 2001. BMBF-Studie, www.stefi.de

1.2 Bedeutung des Semantic Web für RS

Wie das World Wide Web, so geht auch das Semantic Web zurück auf Tim Berners-Lee. Auf der „The Emerging Technologies Conference“ am Massachusetts Institute of Technology (MIT) im Jahr 2004 nennt Berners-Lee die Anfänge des Internet: „a primeval soup of many things that know each other but haven't been put together“ und das Semantic Web nennt er: „killer application in the life sciences“. In seinem Artikel[BL01] definiert er das Semantic Web als Erweiterung des existierenden Web, in der Informationen eine definierte Bedeutung erhalten, wodurch Computer und Menschen zusammenarbeiten können. Er schreibt, dass es einen Unterschied gibt zwischen Informationen, die für den menschlichen Gebrauch produziert und jenen, die hauptsächlich für Maschinen geschaffen wurden. Damit das Semantic Web funktionieren kann, müssen Computer Zugang zu strukturierten Sammlungen von Informationen und Mengen von Inferenzregeln haben, die sie dann nutzen, um automatisch schlussfolgern zu können. Bereits 2001 nennt Berners-Lee zur Realisierung offene Standards wie XML (eXtensible Markup Language), mit deren Hilfe Benutzer ihren Dokumenten eine eigene Struktur mit eigenen Tags geben können, ohne Aussagen über dessen Bedeutung zu treffen. Er erwähnt auch RDF (Resource Description Framework), mit dessen Hilfe Bedeutung ausgedrückt werden kann, in dem sie kodiert in Tripeln, also elementaren Sätzen mit Subjekt, Prädikat, Objekt, maschinenlesbar wird. Als dritte Basiskomponente sieht Berners-Lee die Ontologien, die philosophisch gesehen Theorien über die Natur der Existenz sind. Viele der in Bibliotheken vorhandenen Informationen liegen heute bereits in maschinenlesbarer Form vor, ganz abgesehen davon, dass Publikationen mit ihrem gesamten Inhalt bereits alle Informationen über sich selber enthalten und lediglich intelligent verarbeitet werden müssen.

So wie die Autoren in [HKRS08] könnte man das heutige Web als das syntaktische Web betrachten, in dem charakterisiert ist, was wohlgeformte Daten sind und das semantische Web erweitert diese Syntax um Bedeutung. Der Gartner Hype Cycle[2] für aufkommende Technologien 2007 ordnet Semantic Web in der dritten von fünf Phasen ein und erwartet noch mehr als 10 Jahre, bis es die Masse der Bevölkerung nutzen wird. Berners-Lee schreibt dazu: „The real power of the Semantic Web will be realized when people create many programs that collect Web content from diverse sources, process the information and exchange the results with other programs“.

Recommender Systeme können nicht nur von den Eigenschaften und Funktionalitäten des Semantic Web profitieren sondern durch den Einsatz in Bibliotheken diesem auch auf dem Weg zur produktiven Nutzung helfen. Bibliotheken erreichen online und „offline“ vor Ort viele Benutzer, die wie in der genannten Klatt-Studie sehr häufig bei der Literaturauswahl auf Empfehlungen anderer vertrauen. Ein wissensbasiertes RS, dass den Inhalt von Publikationen und Relationen untereinander semantisch erfassen und in einem bestimmten Grad verstehen kann, ist prädestiniert dazu, qualitative Empfehlungen zu generieren und die Benutzer damit zufrieden zu stellen. Dieser Erfolg wird dann auch direkt mit der Bibliothek in Zusammenhang gebracht.

Semantische RS, die Semantic Web Techniken verwenden, werden in [PdCDL08] als die

[2] mehr Informationen unter `http://www.gartner.com/pages/story.php.id.8795.s.8.jsp`

erfolgversprechensden Systeme angesehen. Noch besser seien nur gemischte Systeme, die zusätzlich noch Filtertechniken (z.B. aus dem Trust Network) und kontextuelle Informationen verwenden. Durch die Verwendung von Ontologien würden die folgenden Probleme herkömmlicher RS abgeschwächt:

- homogene Darstellung der Informationen
- domänenspezifische Benutzer-Präferenzen in einen Zusammenhang bringen
- effiziente Empfehlungen in sozialen Netzwerken und für CF-Systeme (siehe) ermöglichen
- Kaltstartproblem eingrenzen durch Inferieren von fehlenden Informationen auf Ontologien
- semantische Erweiterung von Benutzerprofilen
- bessere Beschreibung der Systemlogik durch Regeln

2 Überblick über Recommender Systeme

Recommender Systeme (RS) sind Programme, die aus Software-Tools und Techniken bestehen, um Benutzern nützliche Ressourcen zu empfehlen. Sie haben sich als wertvoll erwiesen beim Umgang mit der den Benutzer umgebenden Flut an Informationen.[RRSK10] In [JZ10] heißt es: "Recommendation Systems reduce information overload by estimating relevance."Mittelpunkt der Empfehlungen sind beliebige Ressourcen wie z.B. Bücher, CDs, sonstige Produkte, Filme, Nachrichten, Artikel, Publikationen usw.[PdCDL08], wobei ein RS normalerweise auf eine Art Ressource fokussiert ist. Meistens wird das Problem einer Empfehlung verallgemeinert und auf das Vorhersagen eines Rankings für eine Menge von Ressourcen reduziert. Je nach Art der Vorhersage existieren verschiedenartige RS:[AT05], [JZ10]

a) Systeme mit **kollaborativen** Empfehlungen (social-based, collaborative): Dem Benutzer werden Ressourcen empfohlen, die von anderen ähnlichen Benutzern bevorzugt werden.

b) Systeme mit **inhaltsbasierten** Empfehlungen (content-based): Dem Benutzer werden Ressourcen empfohlen, die ähnlich zu bevorzugten Ressourcen sind.

c) **hybride** Systeme: Kombinieren die beiden genannten oder andere Methoden.

d) Systeme mit **wissensbasierten** Empfehlungen (knowledge-based): Dem Benutzer werden Ressourcen empfohlen, die für ihn anhand einer Wissensbasis am geeignetsten erscheinen. In der Wissensbasis liegen zusätzliche Informationen über die Ressourcen und die Benutzer, z.B. implizites Wissen aus einer Produkt-Ontologie.

e) Systeme mit **ökonomischbasierten** Empfehlungen (economic factor-based): Dem Benutzer werden die "günstigsten"Ressourcen empfohlen, wobei definiert werden muss, was günstig ist (z.B. Kosten-Nutzen-Relation)[PdCDL08]

Die Art der Empfehlung ist entscheidend für die Qualität aber auch Komplexität der Empfehlungen. Jedes System bringt zwar Vor- und Nachteile mit sich, doch grundlegend kann das Problem einer Empfehlung für alle Verfahren wie folgt formal beschrieben werden:

- Sei C die Menge aller Benutzer.
- Sei S die Menge aller zu empfehlenden Ressourcen.
- $u : C \times S \rightarrow R$ ist die Nützlichkeitsfunktion, die den Nutzen von Ressource $s \in S$ für Benutzer $c \in C$ misst, R ist das sortierte Ranking
- $\forall c \in C, s'_c = arg^{max}_{s \in S} r(c, s)$ Für jeden Benutzer soll Ressource $s' \in S$ gewählt werden, die die Nützlichkeitsfunktion für den Benutzer maximiert.

Je nach Art des RS wird die Funktion u aus verschiedenen Eingaben berechnet, was dann Schätzung der Nützlichkeit einer Ressource für einen Benutzer darstellt. Dabei tritt häufig das sogenannte Kaltstart-Problem[3] dann auf, wenn u nicht für alle Paare $C \times S$ existiert und extrapoliert werden muss. An dieser Steller können heuristische Methoden oder die Rating-Vorhersage von unbewerteten Ressourcen helfen.

2.1 Kollaborative Empfehlungen (Collaborative Filtering CF)

Hierbei wird die Nützlichkeitsfunktion $u(c, s)$ für Benutzer c und die unbewertete Ressource s basierend auf den Nützlichkeitswerten $u(c', s)$ der ähnlichen Benutzern $c' \in \widehat{C}$ generiert. Eine Variante ist das ressourcenbasierte CF (item-based CF), bei dem nicht die ähnlichen Benutzer gesucht werden, sondern zu den gut bewerteten ähnliche Ressourcen. Ein sehr populärer Vertreter dieser Algorithmus-Variante ist das sogenannte „item-to-item CF" von Amazon[LSY03]. Die unten genannten Verfahren funktionieren für die diese Alternative genauso.

Grob kann man die kollaborativen Methoden in die Klassen speicherbasiert und modellbasiert einteilen[JZ10].

Speicherbasierte Verfahren Die speicherbasierten (bzw. memory-based) Verfahren sind Heuristiken für Bewertungsvorhersagen, die oft auch als user-based nearest-neighbor CF bezeichnet werden. Sie basieren auf der Menge von Ressourcen, die bereits durch

[3]Das Kaltstart-Problem ist ein potentielles Problem bei Informationssystemen und existiert während einer gewissen Anlaufphase, in der noch keine oder nicht ausreichend Informationen über Ressourcen und Benutzerprofile vorliegen. Die Folge ist, dass keine Schlussfolgerungen, hier im Speziellen Empfehlungen, generiert werden können.

ähnliche Benutzer bewertet wurden, wobei keinerlei Wissen über die Ressourcen benötigt wird. Ein generelles Problem dabei ist, dass verschiedene Benutzer die Bewertungsskala auch unterschiedlich interpretieren können. Einige Methoden (wie in Gleichung (1)) überwinden dieses Problem, in dem die Bewertungen eines Benutzers c nicht mehr absolut sondern mit ihrer Abweichung vom Durchschnitt ($\overline{r_c}$) in die Berechnung eingehen. Es existieren u.a. folgende Vorhersagefunktionen $r_{c,s}$ für Benutzer c und die unbewertete Ressource s :

a) Mittelwert der Bewertungen aller N ähnlichen Benutzer:
$r_{c,s} = \frac{1}{N} \sum_{c' \in \widehat{C}} r_{c',s}$

b) normalisierte, nach Ähnlichkeit gewichtete Summe:
$r_{c,s} = k \sum_{c' \in \widehat{C}} sim(c, c') \times r_{c',s}$

c) normalisierte, justierte, nach Ähnlichkeit gewichtete Summe:
$r_{c,s} = \overline{r}_c k \sum_{c' \in \widehat{C}} sim(c, c') \times (r_{c',s} - \overline{r}_{c'})$

Es gibt diverse Ansätze, mit denen die Ähnlichkeit zwischen Benutzern berechnet werden kann. Die zwei populären Verfahren (1) Pearson-Korrelation und (2) Cosinusmaß basieren auf der Tatsache, dass für die Berechnung lediglich Ressourcen betrachtet werden, die von beiden Benutzern x und y bewertet wurden. Das spiegelt sich in der Menge $S_{xy} = \{s \in S | r_{x,s} \neq \emptyset \wedge r_{y,s} \neq \emptyset\}$ wieder. In (2) werden je zwei Benutzer anhand ihrer $m = |S_{xy}|$ Bewertungen als Vektoren in einem m-dimensionalen Raum betrachtet. Ihre Ähnlichkeit wird dann durch den Winkel, also den Abstand zwischen den beiden Vektoren beschrieben. Je kleiner der Winkel zwischen den Vektoren, desto ähnlicher sind sich die Benutzer.

$$sim(x, y) = \frac{\sum_{s \in S_{xy}} (r_{x,s} - \overline{r}_x)(r_{y,s} - \overline{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{x,s} - \overline{r}_x)^2 \sum_{s \in S_{xy}} (r_{y,s} - \overline{r}_y)^2}} \tag{1}$$

$$sim(x, y) = cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\|_2 \times \|\vec{y}\|_2} = \frac{\sum_{s \in S_x y} r_{x,s} r_{y,s}}{\sqrt{\sum_{s \in S_x y} r_{x,s}^2} \sqrt{\sum_{s \in S_x y} r_{y,s}^2}} \tag{2}$$

Ein grafisches Beispiel für das Cosiunusmaß wird in Abbildung 2 gezeigt. Hierbei wurden für dieselbe Ressource jeweils 3 Bewertungen durch die Benutzer x und y abgegeben. Die Länge der Vektoren ergibt sich aus den Bewertungen. Der Winkel zwischen beiden Vektoren drückt dann die Ähnlichkeit zwischen Ihnen aus.

Modellbasierte Verfahren Modellbasierte (bzw. model-based) Verfahren sind Algorithmen für Bewertungsvorhersagen und benutzen die Menge der bewerteten Ressourcen, um

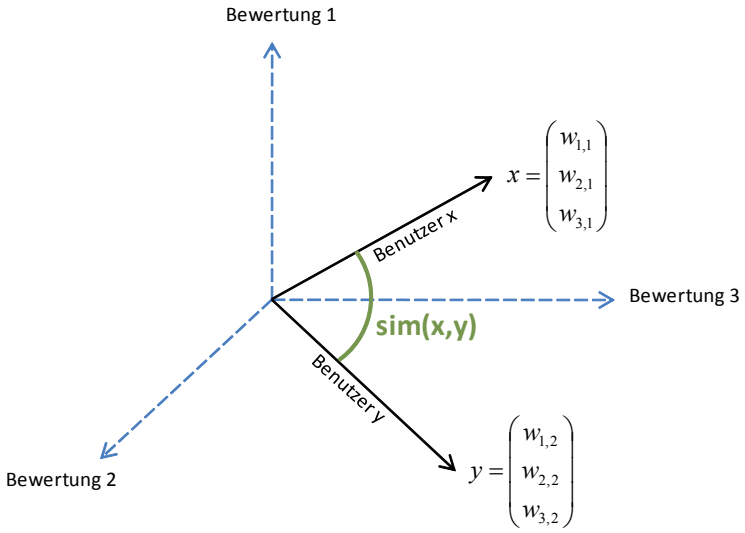

Abbildung 2: Winkel zwischen zwei Vektoren als Ähnlichkeitsmaß

ein Modell zu erlernen. Dabei wird dieses Modell in einer Preprocessing-Phase erstellt und zur Laufzeit lediglich zur Berechnung benutzt. Es muss in regelmäßigen Abständen geupdatet werden, was einen erneuten Berechnungsaufwand darstellt. Es kommen diverse Ansätze und Methoden zum Einsatz:

- Statistik, Matrizenfaktorisierung (LSI - Latent Semantic Indexing, SVD - Singular Value Decomposition)
- Ableitung von Regeln (z.B. Vergleich von Warenkörben und Aussagen wie ”Wenn Kunde X kauft, interessiert er sich auch für Y”)
- Probabilistische Modelle (Cluster, Bayessche Netze, Probabilistisches LSI)
- Algorithmen aus dem Bereich Machine Learning

Empirische Analysen zeigen, dass bei den modellbasierten Verfahren vor allem mit probabilistischen Methoden relativ gute Ergebnisse erzielbar sind

Der Vorteil von CF-Verfahren ist, dass die generierten Empfehlungen schon sehr effektiv sind. Eine mögliche Verbesserung für speicherbasierte Verfahren z.B. sieht vor, alle paarweisen Ähnlichkeiten zwischen Benutzern in einem Preprocessing-Schritt zu berechnen und dies nur periodisch zu wiederholen.
Ein großer Nachteil ist aber, dass CF vom Kaltstart-Problem für neue Benutzer ohne Bewertungen und neue unbewertete Ressourcen betroffen ist. Gute Bewertungen benötigen eine kritische Anzahl an Benutzern und Bewertungen, die in einer gewissen Anlaufphase eines neuen Systems noch nicht gegeben ist. Dieses Problem wird auch häufig als ”First-rater problem”bezeichnet. Auch später im laufenden Betrieb ist ein CF System davon

abhängig, dass Benutzer Bewertungen für möglichst viele Ressourcen abgeben. Ansonsten entwickelt sich in diesem Fall das ßparsity problem", bei dem z.B. Vektoren für modellbasierte Methoden nur dünn besetzt sind und es zwischen Benutzern kaum Überlappungen gibt. Dann können keine effektiven Bewertungen gegeben werden.
Diese Probleme kann man zwar mit Weiterentwicklungen oder anderen Algorithmen abschwächen (z.B. Transitive Nachbarschaften, Standardwert-Verfahren: fehlende Bewertungen werden mit einem Standardwert belegt und tauchen so trotzdem in Berechnungen auf), aber nicht vollständig beseitigen. In [RV97] findet man ein Übersicht über bereits entwickelte kollaborative Systeme. Einige davon, die Publikationen betreffen, werden in Abschnitt 3 näher beleuchtet.

2.2 Inhaltsbasierte Empfehlungen (Content-based recommendation CB)

Bei diesen Verfahren wird die Nützlichkeitsfunktion $u(c, s)$ für Benutzer c und die unbewertete Ressource s basierend auf den Nützlichkeitswerten $u(c, s_i)$ generiert, die c zu s ähnlichen Ressourcen $s_i \in S$ abgegeben hat. Man kann sagen, mit CB wird versucht, Zusammenhänge zwischen bewerteten Ressourcen zu erkennen und damit Präferenzen des Benutzers vorherzusagen. Dieses Verhalten macht die Abgrenzung zwischen inhaltsbasierten und wissensbasierten Methoden relativ unscharf.

Auch die inhaltsbasierten Ansätze kann man ungefähr in 2 Gruppen einteilen: die heuristischen Verfahren aus dem Information Retrieval (IR) und modellbasierten Techniken aus dem Bereich des Machine Learning.

Information-Retrieval Verfahren Für IR-Verfahren müssen Profile der Ressourcen generiert werden, also Repräsentationen des Inhalts einer Ressource. Für textbasierte Ressourcen hat sich das multidimensionale Vektorraummodell und darin speziell das Maß Term-Frequenz-Inverse-Dokument-Frequenz (TF-IDF) bewährt, bei dem Texte als gewichtete Vektoren dargestellt werden. Der Betrag eines Vektors ist abhängig von Vorkommen und Wichtigkeit der im Text vorhandenen Terme. Sei N die Anzahl aller zu empfehlenden Dokumente und Keyword k_j erscheint darin in n_i Dokumenten. $f_{i,j}$ definiert, wie oft Keyword k_j in Dokument $d_{i,j}$ erscheint und das Maximum $max_z f_{z,j}$ wird über die Frequenzen aller Keywords k_z in Dokument d_j berechnet. Dann definiert die Funktion $w_{i,j}$ die TF-IDF Gewichtung von Keyword k_j in Dokument d_j wie in Gleichung (3) gezeigt.

$$w_{i,j} = \underbrace{\frac{f_{i,j}}{max_z f_{z,j}}}_{\text{Term-Frequenz } TF_{i,j}} \times \underbrace{log\frac{N}{n_i}}_{\text{Inverse Dokument-Frequenz } IDF_i} \tag{3}$$

$$Content(d_j) = (w_{1,j}, ..., w_{k,j}) \text{ Dokument als Term-Vektor} \tag{4}$$

Die so entstandenen Vektoren können dann z.B. mit dem Cosinusmaß (siehe Gleichung

(2)) auf Ähnlichkeit geprüft werden. Da es vorkommen kann, dass Vektoren sehr lang und dünn besetzt sind, gibt es verschiedene Verbesserungsmaßnahmen (z.B. Stop-Words entfernen, Stemming-Algorithmen). Ein Nachteil ist jedoch, dass die semantische Bedeutung des Textes trotz Termgewichtung unbekannt bleibt und z.B. negierte Aussagen innerhalb des Textes den Vektor gar verfälschen können.
Eine weitere Verbesserung bildet das Relevanz-Feedback der Benutzer, das eine Bewertung der Empfehlungen dar stellt und Ressourcen als relevant oder unrelevant kennzeichnet. Für die Realisierung eines Relevanz-Feedbacks existiert z.B. der Rocchio Algorithmus[4]. Er ermöglicht für das Finden von Bewertungen ein Fine-Tuning für relevante und unrelevante Ressourcen, wie in Gleichung (5) zu sehen ist. a, b, c werden benutzt, um den Ergebnisvektor Q_m anzunähern an oder zu entfernen vom Ursprungsvektor Q_o (z.B. Ressource, zu der ähnliche Ressourcen gesucht werden) bzw. zu der Menge der relevanten Ressourcen D_r und der Menge der unrelevanten Ressourcen D_{nr}. Sollen lediglich relevante Ressourcen gefunden werden, muss $c = 0$ gesetzt werden.

$$Q_m = (a * \vec{Q_o}) + \left(b * \frac{1}{|D_r|} * \sum_{\vec{D_j} \in D_r} \vec{D_j} \right) - \left(c * \frac{1}{|D_{nr}|} * \sum_{\vec{D_k} \in D_{nr}} \vec{D_k} \right) \quad (5)$$

Modellbasierte Verfahren Als modellbasierte Verfahren seien an dieser Stelle Techniken wie Clustering, Entscheidungsbäume, neuronale Netze und probabilisitische Methoden genannt, die vielfach aus dem Bereich des Machine Learning kommen. So kann man zum Beispiel mit dem naiven Bayes-Klassifikator basierend auf einer Menge von relevanten und nicht-relevanten Ressourcen nicht bewertete Ressourcen annähernd korrekt klassifizieren. Mit den gegebenen Keywords $k_{1,j}, ..., k_{n,j}$ einer Ressource j bestimmt der Bayes-Klassifikator die Wahrscheinlichkeit, dass j in eine bestimmte Klasse C_i gehört (z.B. relevante und unrelevante Ressourcen).

$$P(C_i | k_{1,j} \& ... \& k_{n,j}) \quad (6)$$

Die Vorteile von inhaltsbasierten Empfehlungen sind, dass Ressourcen miteinander verglichen werden können und eine einfaches Benutzerfeedback über Bewertungen in weiteren Empfehlungen berücksichtigt werden kann. Ein Nachteil ist, dass CB-Verfahren Ressourcen aus nicht-textuellen Domänen wie Audio oder Video nicht ausreichend analysieren können. Ressourcen müssen zwingend mit einer maschinenlesbaren Inhaltsbeschreibungen vorliegen wie z.B. Text-Dokumente. Aber auch für textuelle Ressourcen kann kein Algorithmus die Qualität der Ressourcen beurteilen, also gut geschriebene Texte von schlecht geschriebenen Texten unterscheiden. Für andere Typen von Ressourcen muss man entweder eine automatisierte Extraktion der Inhaltseigenschaften anbieten, was in den meisten Fällen sehr schwierig ist, oder aber Ressourcen manuell Eigenschaften zuordnen, was für eine große Menge Ressourcen nicht praktikabel ist. Ein anderes Problem ist, dass der Benutzer mit den Empfehlungen in den seltensten Fällen überrascht werden kann. Es werden ihm schließlich nur Ressourcen empfohlen, die ähnlich zu bereits gut bewerteten Ressourcen sind. Ein gewisser Teil Zufall und Aha-Effekte bleiben aus, die für den Erfolg eines

[4]Der Rocchio Algorithmus wird erläutert in einem IR Buch

RS sehr wichtig sind. Darüber hinaus gibt es das Kaltstart-Problem für neue Benutzer, die noch keine Präferenzen angegeben haben. Sie erhalten keine oder nur unpassende Empfehlungen.

2.3 Wissensbasierte Empfehlungen (knowledge-based)

Diese Systeme orientieren sich an Verkaufsgesprächen, wie sie in einem Geschäft vorkommen. Der „Verkäufer“ ermittelt die Anforderungen des Benutzers und übermittelt Wissen über Produkte, das dem Kunden bei der Auswahl helfen soll. Dafür muss das System zu Beginn das Expertenwissen zu den entsprechenden Ressourcen der Domäne erlangen, um das Verhalten des Verkäufers imitieren zu können. Auf diesem Weg können dann korrekte (deterministische) Empfehlungen erzeugt werden.[JZ10] Die Systeme konzentrieren sich sehr stark auf Wissensbasen, die nicht durch Auswertung von kollaborativen oder inhaltsbasierten Methoden entstanden sind. Generell existieren auch hier wieder zwei Arten von Ansätzen: fallbasierte und bedingungsbasierte Empfehlungen.[RRSK10] Die fallbasierten Methoden ermitteln Empfehlungen auf der Basis von Ähnlichkeitsmetriken.
Bei den bedingungsbasierten Empfehlungen bildet eine regelbasierte Wissensbasis Grundlage. Die Regeln definieren, wie Benutzerbedürfnisse auf Ressourcen abgebildet werden müssen. In der Wissensbasis befinden sich Variablen (Benutzeranforderungen, Resourceneigenschaften) und eine Menge an Regeln: logische Folgerungen, schwer- und leichtgewichtige Bedingungen, Lösungspräferenzen. Das System kann in einer Variante verschiedene Aufgaben erfüllen:

- Anforderungen des Benutzers finden: es entsteht eine Untermenge der Ressourcen, die alle Regeln erfüllen. Dafür wird der Benutzer gefragt, welche Anforderungen abgeschwächt oder modifiziert werden können. Danach werden die Bedingungen erneut angepasst und in einem weiteren Interaktionsschritt erneut evaluiert.
- Untermenge an Ressourcen finden: sie erfüllen das Maximum an gewichteten Bedingungen. Vor allem erfüllen die Ressourcen alle dieselbe Menge an Bedingungen.
- Ressourcen in einem Ranking sortieren: es entsteht ein Ranking nach den Gewichten der erfüllten Bedingungen

Die Vorteile bei wissensbasierten System sind einerseits, dass die generierten Empfehlungen qualitativ hochwertig, deterministisch und damit erklärbar sind und andererseits, dass das Kaltstart-Problem nicht auftreten kann, da die Benutzerbedürfnisse direkt während einer Session abgefragt werden. Doch auch diese Systeme haben ihre Grenzen und Nachteile. Beim Anlauf des Systems muss erheblicher Aufwand für den Aufbau der Wissensbasis (domänenspezifisches Expertenwissen) und der Wissensverarbeitung betrieben werden. Wirklich qualitative Empfehlungen erfordern mehrere Interaktionszyklen zischen dem System und dem Benutzer, was Benutzern vielleicht negativ auffällt. Darüberhinaus generiert das System meistens statische Empfehlungen und kann nicht auf kurzfristige Trends reagieren. Damit geht ein gewisser Teil der Überraschung verloren.[JZ10]

2.4 Hybride Empfehlungssysteme

Zur Einschränkung der Nachteile einzelner Systeme werden verschiedene andere Systeme miteinander derart kombiniert, dass möglichst ihre Vorteile in ein neues System einfließen. Es gibt grob gesehen drei Entwürfe der „Hybridisierung“[JZ10], die im Folgenden näher beschrieben werden.

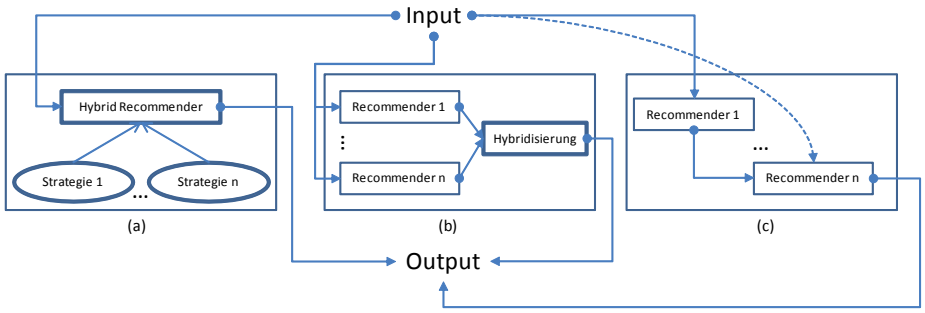

Abbildung 3: (a) monolithisches System, (b) paralleler Betrieb, (c) Pipelining Design

(a) Monolithisches Auswerten von verschiedenen Methoden Hierbei fließen Aspekte verschiedener Strategien in einen einzigen Empfehlungsalgorithmus ein. Die einzelnen Recommender tragen ihre Empfehlungen jedoch lediglich virtuell bei, da erst die hybride Komponente deren Eigenschaften mit zusätzlichen Eingaben zu Empfehlungen umwandelt. Zum Beispiel wäre ein System denkbar, dass inhaltsbasierte Empfehlungen generiert und zusätzlich auch kollaborative Daten auswertet.

(b) Paralleler Betrieb von mehreren Systemen Beim parallelen Betrieb berechnen die einzelnen Recommender nach ihrer Strategie Empfehlungen bzw. komplette Empfehlungslisten, die in einem Hybridisierungsschritt miteinander kombiniert werden. Dabei können Gewichtungen dafür sorgen, wie sehr die einzelnen Bewertungen in eine Gesamtempfehlung einfließen sollen. Der extremste Fall einer Gewichtung wäre ein „Switch“, also das Ein- und Ausschalten bestimmter Methodiken. Das findet beispielsweise Anwendung, wenn wenige Daten zur Ermittlung einer Empfehlung vorhanden sind. In diesem Fall soll eine wissensbasierte Empfehlung generiert werden, ansonsten eine kollaborative. Die Gewichtungen können entweder empirisch ermittelt und fest voreingestellt werden oder aber dynamisch anpassbar sein.

(c) Aufruf verschiedener Systeme in einer Pipeline Bei dieser Variante stellen die einzelnen Recommender eine Preprocessing Phase für die jeweils folgenden Recommender dar. Das führt zu nach bestimmten Kriterien verfeinerten Empfehlungslisten.

Wie bereits genannt, ist die Art des Empfehlungssystems enorm wichtig für die Qualität der Empfehlungen aber auch die Akzeptanz durch die Benutzer. Je nachdem welche Benutzer das System verwenden und welche Ressourcen vorhanden sind, kann das eine oder das andere System geeigneter sein. Eine Bibliothek sollte genau abwägen, was für für sie die beste Lösung ist.

3 Existierende Lösungen

Es existieren bereits diverse semantische und nicht-semantische Ansätze, Recommender Systeme in Bibliotheken zu etablieren. In dem folgenden Kapitel sollen diese Ansätze vorgestellt und untersucht werden. Im Fokus stehen dabei Fragen wie „In wie weit erfüllen sie die Anforderungen von Bibliotheken an ein solches System?“ und „An welchen Stellen gibt es Verbesserungspotential, das in die Entwicklung eines neuen Systems fließen kann?“

3.1 BibTip

BibTip wurde am Karlsruher Institut für Technologie (KIT) der Universität Karlsruhe von 2002 bis 2007 entwickelt und wird im OPAC der KIT Bibliothek eingesetzt. 2009 wurde die BibTip GmbH gegründet, um BibTip zu einem kostenpflichtigen Dienst auszubauen und die Weiterentwicklung abzusichern. Damit ist BibTip keine Software mehr sondern wurde zu einer Dienstleistung neukonzeptioniert, die mittlerweile von vielen Bibliotheken in Deutschland genutzt wird[5] Zusammenfassend kann BibTip als verhaltensbasierter Recommender beschrieben werden, der Empfehlungen auf Basis eines anonymisierten Monitoring des Benutzerverhaltens und seiner Auswertung generiert. Das Unternehmen selber definiert seine Empfehlungen als „Links auf inhaltlich verwandte Titel, die durch das Beobachten des Benutzerverhaltens bei der OPAC-Recherche und dessen statistische Analyse erzeugt werden.“ Da die Datenanalyse und Verwaltung der Empfehlungen auf den Servern von BibTip stattfinden, entsteht den Bibliotheken selber nur ein geringer technischer Aufwand zur Einbindung des Service. Sie müssen keine Software installieren und administrieren oder gar extra Hardware anschaffen.

Abbildung 4 zeigt in Anlehnung an [GSNT03] die 3-Schichten-Architektur von BibTip. Auf der untersten Schicht befindet sich die Datenhaltung, auf der mittleren Schicht arbeiten die diversen Agenten und die oberste Schicht beinhaltet die Schnittstelle zu den Benutzern. Die beiden unteren Schichten teilen sich auf Bibliothek und Empfehlungs-Server auf. So gibt es auf beiden Seiten bestimmte Daten und Agenten.
Jeder der Agenten übernimmt eine spezifsche Aufgabe. Der OPAC-Agent nimmt Anfragen über die Schnittstelle des Benutzers entgegen. Er trägt die HTTP-Anfrage selber in ein Transaktions-Log beim entsprechenden Agenten mit einer dazugehörenden Session-ID ein. Der Transaktions-Agent ruft den Aggregations-Agenten auf, der dann eventuell

[5] Wie man der Webseite zu Referenzen entnehmen kann, verwenden aktuell rund 50 deutsche Bibliotheken BipTip für ihre Online-Kataloge.

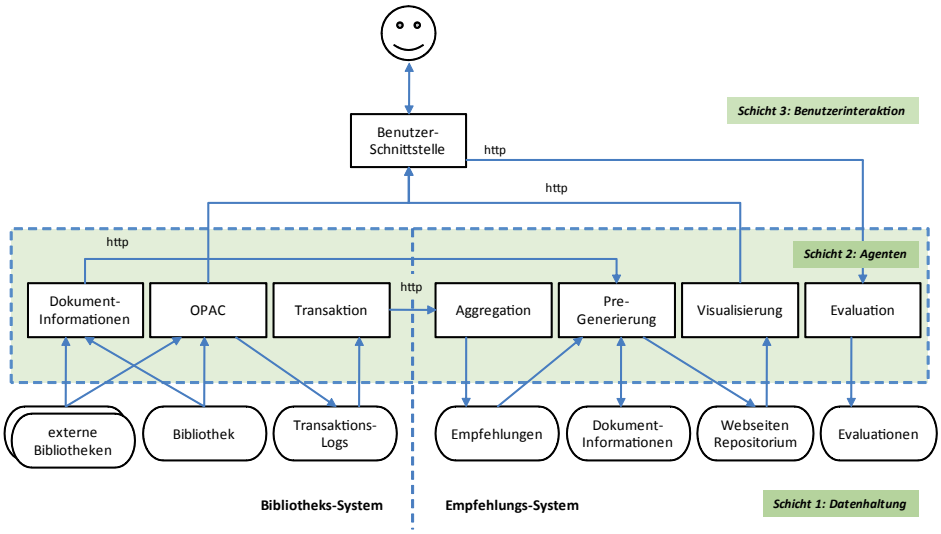

Abbildung 4: 3-Schichten-Architektur bei BibTip

vorkommende Empfehlungen extrahiert. Er leitet diese weiter an den Agenten der Pre-Generierung, der zusammen mit dem Agenten Dokument-Information auf Bibliotheksseite entsprechende Daten aus der eigenen und weiteren externen Bibliotheken sammelt und sie auf Serverseite abspeichert. Der Pre-Generierungs-Agent erzeugt die Webseite für das angeforderte Dokument und reichert sie, wenn denn vorhanden, mit Empfehlungen an. Der Agent Visualisierung schließlich leitet diese Webseite als Ausgabe an die Benutzerschnittstelle weiter. Der Agent Evaluation nimmt Bewertungen und Feedback des Benutzers entgegen und speichert sie für spätere Analysen.

BibTip basiert auf der Repeat-Buying-Theorie von Andrew Ehrenberg[GSNT03], die auf der Analyse von Konsumentenverhalten basiert. Ehrenberg konnte mit seiner Theorie zeigen, dass Menschen nach einer einmal getroffenen Kaufentscheidung auf ihre Erfahrung mit dieser Entscheidung zurückgreifen und beim nächsten Kauf erneut auf die Marke des gekauften Produkts vertrauen. Damit beschreibt die Theorie zwar das Vorkommen einer warenunabhängigen Kauf-Regelmäßigkeit, nennt aber keine Gründe dafür. BibTip macht von dieser Theorie insofern Gebrauch, als dass Benutzer mit ihrem „initialen Verhalten“ Schlussfolgerungen über Interessen zulassen. Beispiel: Wenn der Benutzer sich für Publikation X interessiert, wird er sich auch für Publikation Y desselben Autors interessieren.

Wie zu vermuten ist, benötigt BibTip eine gewisse Anlaufphase, in der Verhaltensdaten gesammelt und analysiert werden müssen[MS08]. Die Autoren schreiben auch, dass „es meist mehrere Monate dauern wird, bis genügend statistisches Material für valide Empfehlungen vorliegt.“ Ein stark frequentiertes BibTip-System kann die Anlauf-Phase reduzieren. Aber trotzdem ist auch dieser Recommender nicht frei vom Kaltstart-Problem.
Positiv sei angemerkt, dass BibTip auf jegliche Ressourcen angewendet werden kann, da

es seine Empfehlungen aufgrund von Benutzerverhalten generiert und somit medienneutral arbeitet. Der aufwendige Aufbau einer Ressourcen-Wissensbasis entfällt.

3.2 ExLibris bX

ExLibris als einer der größten Anbieter von Softwarelösungen für Bibliotheken bietet mit seinem Dienst bX seit 2009 auch ein RS für Nutzer von Bibliotheken an. Diese können den Service allerdings nur nutzen, wenn sie bereits das Produkt SFX von ExLibris erworben haben. Eine Bibliothek muss keine Software installieren, sondern kann den Dienst als On-Demand-Service lediglich im SFX-Menü aktivieren. bX ist ein Ergebnis der Forschungen der Autoren von [BdS06]. bX basiert auf Standards wie OpenURL und OAI-PMH, zwei populäre Protokolle bzw. Verfahren im Umfeld von Bibliotheken zur eindeuitgen Identifizierung von Ressourcen und dazugehörender Metadaten. bX ist ein System, das Empfehlungen basierend auf der Analyse von angeklickten Links innerhalb einer Session generiert. Damit ist auch bX ein verhaltensbasierter Recommender, der für Empfehlungen Statistiken auswertet. Diese werden vom durch ExLibris angebotenen Link Resolver jeder Institution in Logfiles gesammelt. Abbildung 5 zeigt das Ablaufdiagramm für bX. Auffallend ist, dass nicht nur die Informationen aus der eigenen Institution für eine Emp-

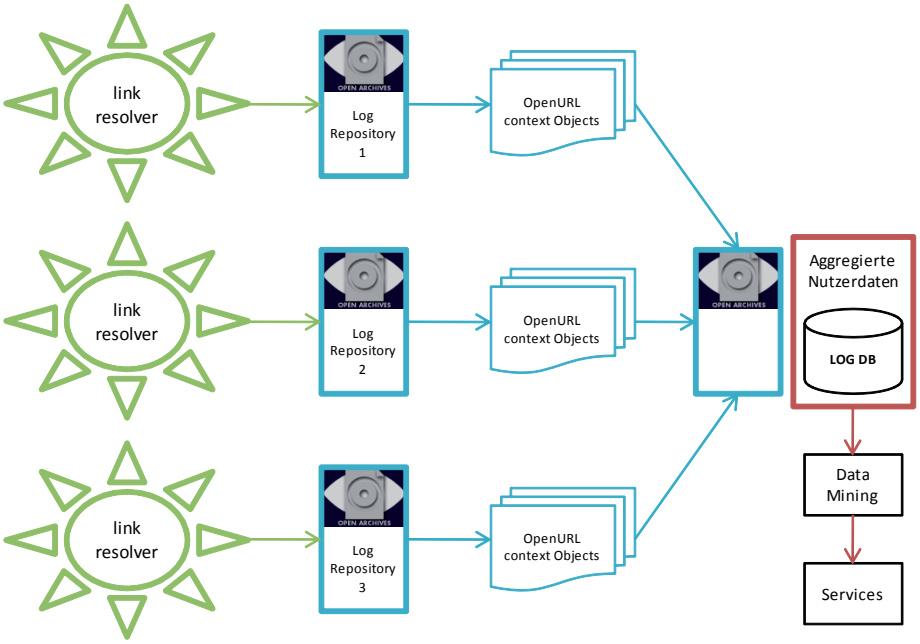

Abbildung 5: bX Ablaufdiagramm

fehlung herangezogen werden, sondern aus sämtlichen zur Verfügung stehenden Institutio-

nen und erzeugten Logdateien. Die Link Resolver zeichen Nutzungsmuster von Artikeln auf und Ergebnisse aus einem Data-Mining-Schritt werden als diverse Services, darunter RS, zur Verfügung gestellt. bx versucht sich Problemen von herkömmlichen verhaltensbasierten RS zu stellen. Darunter fallen die De-Duplizierung von Referenten (Ressourcen, Artikel, Publikationen) und Agenten (Benutzer). Recherchierbare Ressourcen können in in verschiedenen Formen existieren und beziehen sich doch auf denselben Inhalt. Ebenso können dieselben Benutzer z.B. durch das Nutzen von Proxyservern fälschlicherweise unterschiedlich identifiziert werden. Diese Probleme werden aktuell durch einfache Verfahren in einem Schritt angegangen (Identifizierung der Referenten durch die ersten 25 Zeichen des Titels, Zuordnung eines Benutzers zu einer Session) und sollen zukünftig noch zufriedenstellender gelöst werden.

In [ExL09] schreiben die Verfasser, dass ”durch das Anbieten des Empfehlungsdienstes bX eine Bibliothek ihre Benutzer mit einem wertvollen Werkzeug unterstützt, das den Anwendererwartungen in Bezug auf aktuelle Dienste in der Art des Web 2.0 entspricht und Forschenden hilft, sich auf Artikel von potentiellem Interesse und Relevanz für ihr Thema zu konzentrieren.”Durch den Einsatz von bX würde die Bibliothek ihre Position als Informationsversorger stärken. In der Ankünsigungsnachricht für den Dienst schreibt der Konzern ”We view this service as an extremely important piece of the triangle of the discovery-recommendation-fullfilment process. This is the next ’killer app’”[ExL09].

Als verhaltensbasiertes RS sammelt auch bx viele Daten über ihre Benutzer und erlaubt damit den Bibiotheken ”monitor usage at a high level of detail”. Wie bx aber mit Datenschutz und dem Schutz der Privatsphäre umgeht, wird offen gelassen bzw. erst als Ergebnis zukünftiger Forschungen erwartet. Die Herausgeber wollen ”technical issues associated with referent identificationßukünftig weiter erforschen und Ergebnisse verbessern. Das Problem des Kaltstarts wird offensichtlich nicht so stark eingeschätzt, da über die Verknüpfung verschiedener Linkresolver möglichst immer eine ausreichend große Datengrundlage für die Generierung von Empfehlungen zur Verfügung steht. Wenn allerdings keine sonstigen Institutionen den bx Dienst nutzen, muss auch bx erst in einer Anlaufphase genügend Daten sammeln, um Empfehlungen geben zu können.

3.3 Foxtrot

Die Autoren von [SS09] schreiben, dass die Menge an verfügbarem Inhalt im World Wide Web dazu führt, dass einfaches Browsen zum Finden relevanter Literatur zu zeitintensiv sei. Darüber hinaus würde die Formulierung einer geeigneten Suchanfrage den meisten Benutzern sehr schwer fallen. Die Autoren sehen als Lösung für dieses Problem das Semantic Web, weil intelligentere Suchen möglich seinen. Jedoch verhindere der derzeitige Mangel an semantischen Annotationen von Webseiten eine effektive Suche. Dass Semantic Web dennoch beim Retrieval von Publikationen helfen kann, zeigen die Autoren mit ihrem RS Foxtrot (vormals Quickstep). Abbildung 6 zeigt den ontologischen Aufbau von Foxtrot, einem Empfehlungssystem für online verfügbare Forschungsarbeiten (Paper).[MRS02]

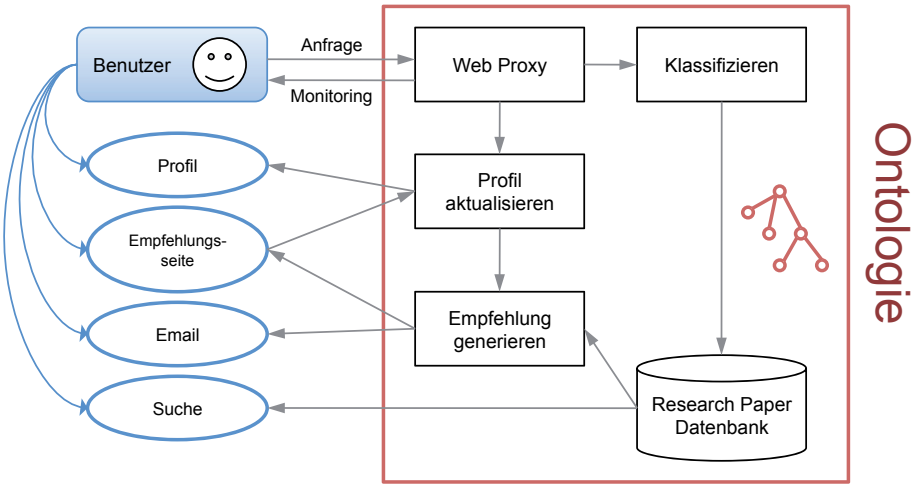

Abbildung 6: Aufbau des Foxtrot Recommender Systems, angeleht an [SS09]

Zur Beschreibung der Benutzer werden eine Ontologie und darauf aufbauende Profile verwendet. Foxtrot generiert Empfehlungen anhand ähnlicher Benutzer und ähnlicher Ressourcen und ist damit als Mischung aus kollaborativen (Pearson-Korrelation 2.1) und inhaltsbasierten Ansätzen (einfaches IR-Verfahren 2.1) ein hybrides Empfehlungssystem. Daneben werden die Interaktionen (besuchte URLs, Feedback) der Benutzer durch einen Web-Proxy geloggt und finden in einem visualisierten Benutzerprofil sowie bei den generierten Empfehlungen Verwendung. Das visualisierte Profil, das sich aus besuchten Publikationen und Datumsangaben zusammensetzt, ermöglicht dem Benutzer die erhaltenen Empfehlungen und damit auch die Funktionsweise des Systems zu verstehen. Die Klassifikation der Paper erfolgt anhand einer Research-Paper-Topic Ontologie[6], die sich als simple IS-A Taxonomie aus Konzepten der Open Directory Project Computer Science Topic Classification[7] und der CORA Digital Library Paper Classification[8][MRS01] zusammensetzen. Jedes der verfügbaren Paper in der Datenbank wird über einen Vektor von normalisierten Termen zusammen mit Metadaten wie Datum, Titel, Klassifikation, Links und der Paper-URL beschrieben. Foxtrot verwendet zur Klassifizierung der Paper den IBk Multiklassen-Classifier[9], eine Variante des k-nearest Neighbour Verfahrens (siehe 2.1), der durch einen AdaBoostM1 Algorithmus[10] unterstützt wird.

Die Profile, die die Interessen der Benutzer repräsentieren, werden durch ontologische

[6]Ein Auszug der Research-Paper-Topic Ontologie ist in [SS09] zu finden.

[7]Gerhart A (2002) OpenDirectory Project Search Results and ODPStatus. Search Engine Guide

[8]McCallum A K, Nigam K, Rennie J, Seymore K (2000) Automating the Construction of Internet Portals with Machine Learning, Information Retrieval 3, 2, 127-163

[9]Mehr Informationen zu IBk Classifier in Aha D., Kibler D, Albert M (1991) Instance-based learning algorithms, Machine Learning 6, 37-66

[10]Beschreibung des AdaBoostM1 Algorithmus in Freund Y., Schapire R.E. (1996) Experiments with a Boosting Algorithm, Proceedings of the Thirteenth International Conference on Machine Learning

Inferenz erweitert. Da es sich bei der Ontologie um eine einfache IS-A Taxonome handelt, sind die Möglichkeiten von inferiertem Wissen zwar gering, aber es werden dennoch Subklassen-Beziehungen ausgenutzt. Dem Benutzerprofil wird zu jedem Interesse und damit zu jedem Topic das generaliserte Super-Topic hinzugefügt. Abbildung 7 zeigt die Funktion zur Ermittlung eines Topic-Werts innherhalb eines Benutzerprofils.

$$Topic - Interesse = \sum_{j=1}^{n} \frac{Interessenswert(j)}{Alter(j)} \tag{7}$$

Erläuterungen:

- Altersangabe von Einträgen erfolgt in Tagen
- n ist die Anzahl der Profileinträgen
- Interessenswert von Ereignissen:
  - Paper angesehen: 1
  - Empfehlung gefolgt: 2
  - Topic als interessant bewertet: 10
  - Topic als uninteressant bewertet: -10
- Interessenswert des Super-Topics: 50% des Wertes des Topis

Außerdem bietet Foxtrot die Möglichkeit, eine externe Ontologie mit entsprechenden Daten einzubinden, um so das Kaltstartproblem auszuschalten.

Die Autoren von Foxtrot haben ein viel versprechendes Konzept eines RS für wissenschaftliche Publikationen vorgestellt. In den umfangreichen Testläufen hat sich gezeigt, dass ihr RS mit Ontologie-Unterstützung eine um 7%-15% verbesserte Empfehlungs-Qualität aufweist als ein System, das keine Ontologie verwendet. Und das wurde schon mit einer sehr einfachen Taxonomie erreicht, die längst nicht das gesamte Potential einer Ontologie ausschöpft. Eine Verbesserung wäre die Entwicklung einer detailliert gebauten Topic-Ontologie, mit der mehr implizites Wissen erzeugt und damit bessere Empfehlungen generiert werden können. Neben der Ontologie profitiert Foxtrot noch von der Mischung aus kollaborativen und inhaltsbasierten Methoden sowie von Benutzer-Monitoring. Gesamt gesehen ergeben diese Daten eine gelungene Empfehlungs-Grundlage.

3.4 Andere hybride Systeme

Neuere RS für Publikationen, wie die drei Systeme TechLens, Fab oder LIBRA konzentrieren sich ähnlich wie Foxtrot auf hybride Ansätze aus kollaborativen und inhaltsbasierten Methoden, um bessere Ergebnisse zu erzielen. Auch sonst ähneln sich sich in ihrer Funktionsweise recht stark, weshalb sie an dieser Stelle zusammen betrachtet werden.

TechLens TechLens ist ein RS, dass sich speziell für den Einsatz in Digitalen Bibliotheken versteht. Die Berechtigung für RS in diesem Bereich sehen die Autoren von [TMA+04] und [KKMB05] vor allem durch die jährliche 1-prozentige Wachstumsrate von neuen Publikationen, was in den nächsten 20 Jahren eine Summe mehr als 10 Millionen Paper ausmacht. Die Autoren haben 10 verschiedene Algorithmen entwickelt und mit über 102000 Paper von CiteSeer getestet. Die wichtigste Eigenschaft des TechLens Systems ist wohl, dass als kollaborative Methode die Paper selber benutzt werden. Damit bewertet ein Paper implizit seine Referenzen als gute Paper, die dann für Empfehlungen verwendet werden. Außerdem wird zwischen Zitation und Paper unterschieden. Dabei ist eine Zitation sozusagen ein Zeiger auf ein Paper, dass selber nicht verfügbar sein muss. Ein Paper ist allerdings eine Zitation, für die der Volltext vorhanden ist. Die Autoren nennen verschiedene Möglichkeiten des Aufbaus eines Benutzerprofils und entscheiden sich für die Verwendung der „One-Paper-Explicit-Short-Term“ Alternative. Damit wird lediglich ein einzelnes Paper (implizit: das letzte gesehene), das der Benutzer selber wählt, für das Benutzerprofil verwendet. Dieses repräsentiert nach Meinung der Autoren das aktuelle Interesse des Benutzers am besten. Der Vorteil von diesem Profil ist, dass auf Benutzerseite kein weiteres Monitoring-System gebraucht wird. Der Nachteil ist allerdings, dass genau dadurch keine Interessensentwicklung beobachtet werden kann.

Folgende Algorithmen wurden für TechLens entwickelt und in mehreren Testläufen evaluiert:

- Pure-CF: Ein reiner kollaborativer Ansatz, der auf dem Standard k-nearest neighbour Algorithmus basiert. In den Testläufen schnitt diese nicht hybride Alternative sehr gut ab. Jedoch konnte Pure-CF nur eine Coverage von 93% erreichen, was bedeutet, dass nicht alle verfügbaren Ressourcen empfohlen werden konnten.
- Denser-CF: Dies ist eine Erweiterung von Pure-CF, bei der die Liste von Zitationen um die Zitationen des aktiven Papers ergänzt wird.
- Pure-CBF: Reiner inhaltsbasierter Ansatz, der auf TF-IDF basiert und die ähnlichsten Paper empfiehlt.
- CBF-Seperated: Basiert ebenso auf TF-IDF und durchsucht als Erweiterung von Pure-CBF nicht nur den Text des Papers sondern auch die Texte aller zitierten Paper und generiert daraus eine gemeinsame Empfehlungsliste.
- CBF-Combined: Arbeitet ähnlich wie CBF-Seperated, jedoch wird als Empfehungsgrundlage der zusammengeführte Text von Paper und der der Zitationen genommen und eine einzelne Empfehlungsliste für diesen Text generiert.
- Hybrid: CF - CBF-Seperated: Empfehlungen aus Pure-CF werden als Eingabe für CBF-Seperated benutzt, wobei jeweils eine Menge an ähnlichen Papern für jede Empfehlung aus CF generiert wird.
- Hybrid: CF - CBF-Combined: Alle Texte der empfohlenen Paper aus CF werden zusammengeführt und als Eingabe für CBF benutzt.

- Hybrid: CBF-Seperated - CF: Für das aktive Paper erzeugt CBF Empfehlungen, die dann benutzt werden, um die Liste der Zitationen für das aktive Paper zu ergänzen. Das Paper mit seinen modifizierten Zitationen dient dann als Eingabe für Pure-CF.
- Hybrid: CBF-Combined - CF: Genau wie CBF-Seperated - CF, nur dass die Empfhelungsliste aus Schritt 1 anders generiert wird.
- Fusion: Beide Ansätze (kollaborativer und inhaltsbasierter) werden parallel laufen gelassen. Jede Empfehlung, die in beiden Listen vorkommt, wird der Ergebnis-Liste mit einem bestimmten Ranking hinzugefügt. Empfehlungen, die nicht in beiden Listen vorkommen, werden an die Ergebnis-Liste angehangen. Fusion konnte in den Test mit Abstand die besten Ergebnisse erzielen und darüber hinaus eine Coverage von 100% erzielen.

Die Ergebnisse zeigen, dass erneut ein hybrider Ansatz qualitativ bessere Ergbnisse geliefert hat als eine reine kollaborative Methode. Das verwendete Benutzerprofil wird dennoch langfristig nicht als geeignet betrachtet, da es mit einem einzigen Paper eine sehr beschränkte Wissenskapazität mit sich bringt.

Fab Fab, das innerhalb des Digital Library Projects an der Stanford University entwickelt wurde, ist wie seine hier genannten Vorgänger auch ein hybrides Recommender System und kombiniert kollaboratives Filtern mit inhaltsbasierten Methoden [BS97]. 1997 vor den großen Suchmaschinen entworfen hat FAB die Aufgabe, relevante und interessante Webseiten aus der Masse der verfügbaren zu finden und zu empfehlen. Fab ist damit auch eines der ersten hybriden Empfehlungssysteme. Empfehlungen basieren auf den bereits getroffenen Bewertungen des aktiven Benutzers und der Bewertungen anderer ähnlicher Benutzer. Benutzer werden in ihrem Profil mit einem gewichteten Termvektor beschrieben. Die Aktualisierung der Profile erfolgt mittels Relevanz-Feedback unter Verwendung des Rocchio Algorithmus (siehe 2.2. Die Webseiten werden ebenfalls durch einen gewichteten Termvektor der Dimension 100 dargestellt, die sich durch Anwendung von TF-IDF ergeben.

LIBRA LIBRA steht für „Learning Intelligent Book Recommending Agent“[MR00] und wurde speziell für digitale Bibliotheken entwickelt und dient der Empfehlung von Büchern. Der Benutzer muss bei einer Suche die ersten zehn Suchtreffer bewerten, woraufhin die Ergebnismenge gemäß den Vorlieben des Benutzers umsortiert wird. Die Ressourcen werden anhand folgender Attribute beschrieben, die derart bei Amazon vorhanden sind: Titel, Autor, Zusammenfassung, Rezensionen, Kundenkommentare, verwandte Autoren, verwandte Titel und Schlagworte. Zur Erstellung eines Profils muss der Benutzer zehn Bücher mit Werten von 1 (schlecht) bis 10 (gut) bewerten. Der Lernalgorithmus Naiv-Bayes (siehe 2.2) aus dem Bereich Machine-Learning wird verwendet, um ein Ranking der Bücher für den Benutzer vorzunehmen. Zur Generierung von Empfehlungen werden die ähnlichen Benutzer zum aktiven Benutzer gesucht. Die Ähnlichkeit der Benutzer untereinander wird mittels Pearson-Korrelation errechnet. Anschließend werden basierend auf

den Bewertungen der ähnlichen Benutzer Vorhersagen der Bewertung durch den aktiven Benutzer getroffen und die besten darunter als Empfehlungsliste ausgegeben.

Wie die genannten Arbeiten zeigen, wird lediglich ein kleiner Teil von möglichen Algorithmen und Methoden genutzt. Semantische Empfehlungen gibt es derzeit so gut wie gar nicht, abgesehen von Foxtrot, wobei allerdings der aktuelle Entwicklungsstand nicht umfassend geklärt werden konnte. Zusammenfassend kann man schließen, dass hybride Recommender Systeme bessere Ergebnisse produzieren, die Ausnutzung von möglichst vielen Informationen ebenso essentiell für effektive Empfehlungen ist und der Einsatz von Semantischen Methoden einen neuen Entwicklungsschritt für RS darstellt. Benutzer sollten die generierten Empfehlungen nachvollziehen können und möglichst keinen Mehraufwand mit einem solchen System haben. Dafür sollte der Mehrwert um so höher sein, um vielleicht die Benutzer doch davon überzeugen zu können, an einigen Stellen Feedback zu geben. Aber das sollte für die Funktionsweise nicht vorausgesetzt werden.

4 Ausblick

In diesem Abschnitt sollen Ideen und Lösungen skizziert werden, wie ein effektives Recommender System für Bibliotheken aussehen kann und der Rahmen erläutert werden, in den das RS integriert werden soll.

Ob ein RS erfolgreich sein kann, ist von verschiedenen Faktoren abhängig und wird in [JZ10] beschrieben:

- Recherche: Den Suchaufwand für den Benutzer reduzieren und korrekte Vorschläge unterbreiten.
- Empfehlung: Möglichst Aha-Effekte durch Empfehlung von unbekannten Ressourcen beim Benutzer erzeugen.
- Vorhersage: Gelungenes Abschätzen, wie sehr ein Benutzer eine Ressource mag.
- Interaktion: Den Benutzer stets informieren über die Domäne der Ressourcen und gegebene Empfehlungen mit anschauchlichen Mitteln erklären.
- Anwendung: Auch die wirtschaftliche Entwicklung der einsetzenden Institution beachten und RS mit dem Ziel einsetzen, Verkaufszahlen, Profit und „Clickraten“ zu steigern.

Diese Faktoren sollen für die Umsetzung berücksichtigt werden. Wobei im Umfeld von Bibliotheken der Teil Anwendung eher als Ausleihzahlen und Besucherstatistiken interpretiert werden würde. Parallel dazu könnten Neuanschaffungen innerhalb der Bibliothek durch ein RS beeinflusst werden und Ausgaben für uninteressante Literatur eingespart werden.

Die Begutachtung der RS im Umfeld von Publikationen hat eindeutig gezeigt, dass hybride Systeme bessere Empfehlungen produzieren, weil sie möglichst viele Informationen in die

Berechnung miteinbeziehen. Eine wissens- oder inhaltsbasierte Methode ist stark abhängig von Informationen und Wissen über die Publikationen. Aufgrund der betrachteten Domäne mit textuellen Ressourcen benötigt eine derartige Methode nicht mehr, als die Publikation selbst. Denn alles, was das RS über eine Publikation wissen muss, ist in der Publikation selbst vorhanden. Dieses Wissen muss lediglich extrahiert werden.

Eine kollaborative Methode sollte implizit die Vernetzungen zu anderen Personen ausnutzen. Das können z.B. die Interessen von Mitgliedern innerhalb von Arbeitsgruppen sein, da die einzelnen Mitglieder an ähnlichen Themen arbeiten. Ein weiterer wichtiger Faktor ist, dass die Benutzer mit so wenig Aufwand wie möglich so viele effektive Empfehlungen wie möglich mit einem solchen System erhalten sollten.

4.1 Szenario am Konrad-Zuse-Zentrum

Das Konrad-Zuse-Zentrum für Informationstechnik Berlin (ZIB) ist eine öffentliche Forschungseinrichtung für die Bereiche Mathematik und Informatik, besitzt eine eigene Spezialbibliothek und ist daneben auch Dienstleister für Bibliotheken im deutschsprachigen Raum. Das ZIB besitzt und verwaltet Millionen von interdisziplinären Publikationen für verschiedene Bibliotheken und Verbünde. Diese Publikationen liegen mit strukturierten Metadaten und zu einem beträchtlichen Teil auch mit Volltexten vor. Insbesondere existiert für die meisten Publikationen aus dem Umfeld der Mathematik und Informatik eine Zuordnung zu Klassifikationen aus den entsprechenden Bereichen. Darunter fallen z.B. die Mathematical Subject Classification (MSC [11]) und das ACM Computing Classification System (CCS [12])

Das ZIB entwickelt selber eine Open-Source-Software für das Management von Publikationen in einem Repositorium[13]. Außerdem soll mit dem aktuellen Projekt „myZIB“[14] eine Webanwendung entworfen werden, die es Wissenschaftlern ermöglicht, Dokumente zu recherchieren, sammeln, organisieren und neu zu publizieren. In myZIB stehen der Wissenschaftler und seine Bedürfnisse im Mittelpunkt. Und um dem Wissenschaftler die alltägliche Recherche zu erleichtern, soll in myZIB auch eine Komponente Recommender System integriert werden.

Die folgenden Unterpunkte definieren Funktionalitäten, die speziell das RS beinhalten soll. Nicht betrachtet werden Grundfunktionen, die in myZIB bereits vorgesehen sind.

Benutzerprofil Das Profil ergibt sich implizit aus den innerhalb von myZIB verwalteten Publikationen, da das Hinzufügen von Literatur zum Repositorium als Ausdruck von Interesse gewertet wird. Durch dieses Vorgehen muss der Benutzer keinen Mehraufwand erbringen, um Empfehlungen zu bekommen. Über eine Zuordnung zu einer Arbeitsgruppe erhält der Benutzer auch kollaborative Empfehlungen der Gruppenmitglieder.

[11] MSC im Netz zu finden unter http://www.ams.org
[12] CCS im Netz unter http://portal.acm.org
[13] OPUS4 ist im Netz verfügbar unter opus4.kobv.de
[14] myZIB wurde vorgestellt auf der Wiskomm2010 und ist hier zu finden: Link

Feedback Ein Relevanz-Feedback soll optional möglich sein und kann Ergebnisse des RS positiv beeinflussen. Das Fehlen von Feedback soll parallel aber nicht automatisch zu schlechteren Ergebnissen führen. Der Benutzer soll nicht gezwungen sein, einen extra Aufwand zu erbringen, um das RS nutzen zu können.

Alert-Funktion Wenn neue Literatur in der Bibliothek erschienen ist, wird vom RS überprüft, ob sie für Benutzer interessant ist. Ist dies der Fall wird dem Benutzer per Mail eine Empfehlung verschickt oder im persönlichen Bereich in myZIB benachrichtigt. Auf diese Weise erfährt der Benutzer unverzüglich nach Neuerscheinungen von neuer Literatur ohne erst danach suchen zu müssen.

Visualisierung Der Benutzer soll als Historie seiner Interaktionen eine visualisierte Zusammenfassung erhalten. Dies liefert ihm einen Hinweis auf die Arbeitsweise des RS und hilft, Empfehlungen nachvollziehbar zu machen. Des Weiteren wird der Benutzer eventuell auf diesem Weg ermuntert, häufiger Feedback zu geben, weil er begreift, dass er die Ergebnisse beeinflussen kann.

4.2 Ähnlichkeitsberechnung in drei Schritten

Zur Realisierung des beschriebenen RS für Bibliotheken sollen sukzessive die drei nachfolgend erläuterten Methoden jeweils umgesetzt und evaluiert werden.

4.2.1 Metadatengraph

Die Bildung und Nutzung des Metadatengraphen ist der initiale Schritt bei der Ähnlichkeitsberechnung. Die Publikationen werden in ihren beschreibenden Kontext gebracht. Aus diesem Grund sollen die Publikationen in einen Metadatengrahen eingebunden werden. Abbildung 7 zeigt Entitäten und Beziehungen in diesem Graphen. Eine Publikation besitzt einen oder mehrere Autoren, wobei mehrere Autoren eine Koautorenschaft bilden. Über Zitationen referenziert eine Publikation andere Literatur. Jede Publikation besitzt eine oder mehrere Klassifikationen, die in der Regel vom Autor angegeben werden. Publikationen erscheinen in verschiedenen Journals, die selbst häufig in bestimmten Klassifikationen einzuordnen sind. Weitere von den Autoren vergebene Keywords beschreiben das Paper zusätzlich. Die Publikationen besitzen darüber hinaus natürlich noch Metadaten wie Titel und Jahr. Und Autoren können Institutionen zugeordnet werden.
Das Bilden des Metadatengraphen setzt eine Autoren-Disambiguierung voraus, um Publikationen auch dem korrekten Autor zuzuordnen.

Publikationen, die innerhalb des Graphen nahe beieinander liegen, werden als ähnlich interpretiert. Das bedeutet, je weniger Kanten zwischen zwei Publikationen liegen, desto ähnlicher sind sie sich und sollten als Empfehlungen berücksichtigt werden. Das betrifft z.B. Paper desselben Autors bzw. derselben Koautorenschaft, Paper aus demselben Journal

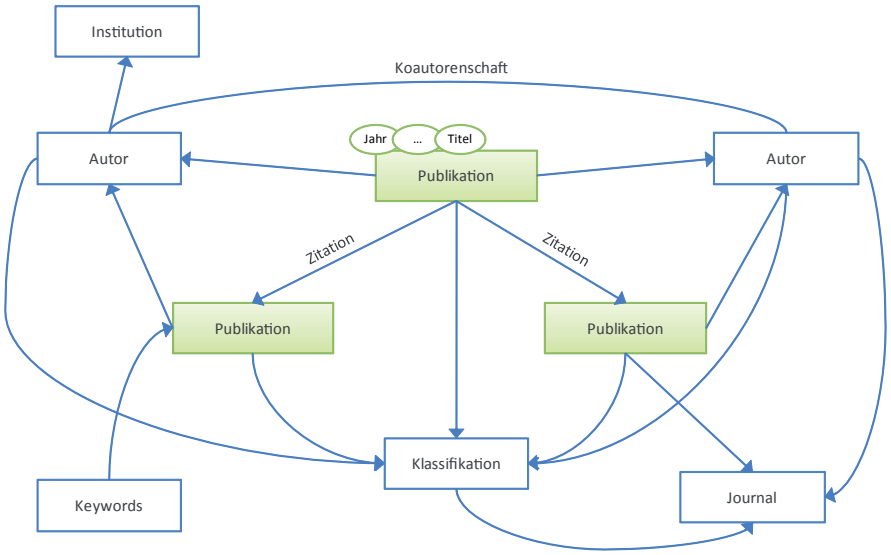

Abbildung 7: Entitäten und Relationen im Metadatengraph

oder Paper aus derselben Institution. Der Metadatengraph ermöglicht nicht nur die Berechnung von Ähnlichkeiten und Empfehlungen, sondern weitreichende semantische Analysen wie Zitationsanalysen, Finden von häufigen Cliquen und Zeitreihenanalysen.

4.2.2 Keyword-Extraktion und Ontologie

Der zweite Schritt bei der Ähnlichkeitsberechnung sieht einen wissensbasierten Ansatz vor, der Empfehlungen aufgrund relevanter Wörter innerhalb des Papers generiert. Damit wird der Versuch unternommen, das Paper zu verstehen und basierend auf der Thematik Vorschläge abzuleiten.

Aus jeder Publikation werden z.B. mittels TF-IDF (siehe 2.2) x relevante Wörter extrahiert, wobei die Höhe von x noch evaluiert werden muss. Das Paper kann zuvor segmentiert werden, so dass diese Wörter z.B. nur aus dem Abstract oder der Einleitung entnommen werden. In diesen Teilen wird die Arbeit zusammengefasst und ein Überblick über den Inhalt gegeben. Die Reihenfolge der Schlüsselwörter ist dabei irrelevant. Man könnte zwar den Ansatz verfolgen, dass ein Wort wichtiger ist, je näher es am Titel der Publikation steht. Aber parallel dazu kann man auch argumentieren, dass Wörter aus dem hinteren Teil den eigentlichen Inhalt tiefer beschreiben. Die gefundenen Schlüsselwörter werden somit unabhängig von ihrer Reihenfolge betrachtet. Sie werden mit der jeweiligen Klassifikation in Relation gesetzt. Damit wird ein Bezug zwischen themenrelevanter Begriffe und der Klassifikation geschaffen. Diese Zuordnung hilft beim Empfehlen von neuen Publikationen, in dem auch aus diesen die relevanten x Wörter extrahiert werden und mit denen in der Klassifikation verglichen werden.

Die Publikation erhält sozusagen mit der Gesamtheit der relevanten Keywords (z.B. als Hash) einen eindeutigen Wert, der auch als Repräsentation genutzt werden kann.

4.2.3 Ontologie und linguistische Methodik

Im finalen Schritt übernimmt eine computerlinguistische Methode das Verstehen des Publikationsinhaltes, weshalb das Extrahieren von relevanten Wörtern entfällt. Mögliche Lösungsansätze für automatisches Textverständnis finden sich z.B. in [Ste06]. Zur Realisierung dieser Lösung müssen noch weitere Anstrengungen in Recherchen und Forschungsvorhaben investiert werden. Dieser Ansatz verspricht effektive Ergebnisse. Er kann auch interdisziplinär umgesetzt werden, da keine Abhängigkeit mehr von Klassifikationen besteht.

Literatur

[AT05] Gediminas Adomavicius und Alexander Tuzhilin. Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions. *IEEE Trans. on Knowl. and Data Eng.*, 17:734–749, June 2005.

[BdS06] Johan Bollen und Herbert Van de Sompel. An architecture for the aggregation an analysis of scholarly usage data. Bericht, Los Alamos National Labatory, 2006.

[Ben04] Christian Benne. Einführung in wissenschaftliche Recherche und Bibliotheksbenutzung. online: Bibliotheksbenutzung.pdf - 11.1.2011, 2004.

[BL01] Tim Berners-Lee. The Semantic Web - A new form of Web content that is meaningful to computers will unleash a revolution of new possibilities. *Scientific American Magazine*, Mai 2001. URL: www.scientificamerican.com, - 25.02.2011.

[BS97] Marko Balabanovic und Yoav Shoham. Fab: Content-based, collaborative recommendation. *Communications of the ACM*, 40:66–72, 1997.

[ExL09] ExLibris. bX - Scholarly Recommender Service. online Broschüre - 6.12.2010, 2009.

[GSNT03] Andreas Geyer-Schulz, Andreas Neumann und Anke Thede. An Architecture for Behavior-Based Library Recommender Systems. *Information Technology and Libraries*, 22(4), 12 2003. ALA Website - 28.02.2011.

[HKRS08] Pascal Hitzler, Markus Krötzsch, Sebastian Rudolph und York Sure. *Semantic Web. Grundlagen*. Springer, 2008.

[JZ10] Dietmar Jannach und Markus Zanker. An Introduction to Recommender Systems. In *25th ACM Symposium on Applied Computing*, Seite 139, 2010.

[KKMB05] J.A. Konstan, N. Kapoor, S.M. McNee und J.T. Butler. TechLens: Exploring the Use of Recommenders to Support Users of Digital Libraries. *Communications of the ACM*, 2005.

[LSY03] Greg Linden, Brent Smith und Jeremy York. Amazon.com Recommendations - Item-to-Item Collaborative Filtering. *IEEE Internet Computing*, Februar 2003.

[MR00] Raymond J. Mooney und Loriene Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the fifth ACM conference on Digital libraries*, DL '00, Seiten 195–204, New York, NY, USA, 2000. ACM.

[MRS01] Stuart E. Middleton, David C. De Roure und Nigel R. Shadbolt. Capturing knowledge of user preferences: ontologies in recommender systems. Bericht, Department of Electronics and Computer Science, University of Southampton, 2001.

[MRS02] Stuart E. Middleton, David C. De Roure und Nigel R. Shadbolt. Foxtrot Recommender System: User profiling, Ontologies and the World Wide Web. Bericht, Department of Electronics and Computer Science, University of Southampton, 2002.

[MS08] Dr. Michael Mönnich und Marcus Spiering. Adding Value to the Library Catalog by Implementing a Recommendation System. *D-Lib Magazine*, 14(5/6), May/June 2008.

[Ova] Vincent Ovaert. Die Literaturrecherche: Suchstrategien. online: Suchstrategien.htm - 27.10.2009.

[PdCDL08] E. Peis, J. M. Morales del Castillo und J. A. Delgado-López. Semantic Recommender Systems. Analysis of the state of the topic. *Hipertext.net*, 6, 2008. hipertext.net - 05.01.2011.

[RRSK10] Francesco Ricci, Lior Rokach, BRacha Shapira und Paul B. Kantor, Hrsg. *Recommender Systems Handbook*. Springer, 2010.

[RV97] Paul Resnick und Hal R. Varian. Recommender systems. *Commun. ACM*, 40:56–58, March 1997.

[SS09] Steffen Staab und Rudi Studer, Hrsg. *Handbook on Ontologies*. International Handbook on Information Systems. Springer, second. Auflage, 2009.

[Ste06] M. Stede. Textverstehen in der Computerlinguistik am Beispiel der Automatischen Textzusammenfassung. In U.H. Wassner H. Blühdorn, E. Breindl, Hrsg., *Text-Verstehen. Grammatik und darüber hinaus*. de Gruyter, 2006.

[TMA+04] Roberto Torres, Sean M. McNee, Mara Abel, Joseph A. Konstan und John Riedl. Enhancing digital libraries with TechLens+. In *Proceedings of the 4th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '04, Seiten 228–236, New York, NY, USA, 2004. ACM.