Verteiltes Information Retrieval für nicht-kooperative Suchserver im WWW

Dissertation
zur Erlangung des Doktorgrades
der Naturwissenschaften

vorgelegt beim Fachbereich Biologie und Informatik
der Johann Wolfgang Goethe-Universität
in Frankfurt am Main

von
Martin Heß
aus Hanau

Frankfurt 2002
(D F 1)

vom Fachbereich Biologie und Informatik der
Johann Wolfgang Goethe-Universität als Dissertation angenommen.

Dekan:	Prof. Dr. Bruno Streit
Gutachter:	Prof. Dr. Oswald Drobnik Prof. Dr. Detlef Krömker
Datum der Disputation:	18. Juni 2002

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als Mitarbeiter an der Professur für Telematik des Fachbereichs Biologie und Informatik der Johann Wolfgang Goethe-Universität in Frankfurt am Main.

Mein besonderer Dank gilt Prof. Dr. Drobnik, der die Arbeit betreut hat. Die vielen Diskussionen mit ihm lieferten immer wieder wertvolle Anregungen zur Weiterentwicklung und haben wesentlich zum Erfolg dieser Arbeit beigetragen.

Bei Prof. Dr. Krömker möchte ich mich für die Übernahme des Zweitgutachtens bedanken.

Darüber hinaus danke ich den Kollegen an der Professur, insbesondere Thomas Schönberger für sein außerordentliches Engagement und Christian Mönch für seine fachliche und moralische Unterstützung.

Mein Dank gilt auch den Mitarbeitern der Stadt und Universitätsbibliothek Frankfurt und des Hochschulrechenzentrums für die gute und fruchtbare Zusammenarbeit.

Auch meinen Eltern und meinen Geschwistern, die meine Arbeit stets aufmerksam verfolgt haben, danke ich sehr. Ihre moralische Unterstützung war für mich sehr wertvoll.

Ganz herzlich bedanke ich mich bei Katharina, die mich stets mit viel Verständnis und noch mehr Geduld begleitet hat.

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ARC	Automatic Resource Compilation
BFS	Breadth-First Search
CACM	Communications of the Association for Computing Machinery
CORI	Collection Retrieval Inference Network
CVV	Cue Validity Variance
DAML	DARPA Agent Markup Language
DBMS	Database Management System
DC	Dublin Core
DDC	Dewey Decimal Classification
df	document frequency
DFS	Depth-First Search
DL	Description Logic
FC	Frankfurt Core
GlOSS	Glossary of Servers Server
HACM	Hierarchical Agglomerative Clustering Methods
HITS	Hyperlink Induced Topic Search
HiWE	Hidden Web Exposer
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
idf	inverse document frequency
isf	inverse server frequency
IR	Information Retrieval
IMDB	Internet Movie Database
KR	Knowledge Representation
MAB	Maschinelles Austauschformat für Bibliotheken
MARC	Machine-Readable Cataloging Format
MESH	Medical Subject Headings
MMMfT	MyMetaMaker for Thesis
OIL	Ontology Inference Layer
OPAC	Online Public Access Catalog
PR	Pagerank

QUEST	Querying Specialized Collections on the Web
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RSLP-CD	Research Support Library Programm Collection Description
sf	server frequency
SCC	Strongly Connected Component
SHOE	Simple HTML Ontology Extensions
STARTS	Stanford Protocol Proposal for Internet Retrieval and Search
STC	Suffix Tree Clustering
tf	term frequency
TREC	Text Retrieval Conference
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
VIR	Verteiltes Information Retrieval
W4F	WysiWyg Web Wrapper Factory
WCC	Weakly Connected Component
WWW	World Wide Web
XML	Extended Markup Language
XSL	Extensible Stylesheet Language
XSLT	Extensible Stylesheet Language Transformations

1. Einleitung

1.1. Problemstellung

Innerhalb der letzten Dekade hat sich das World Wide Web (WWW) von einem experimentellen Hypertext-System zur weltweit verbreitetsten Plattform elektronisch verfügbarer Dokumente entwickelt. Dabei entfernt sich das WWW immer mehr von seiner ursprünglichen Konzeption, in der weitgehend passiv agierende Web-Server die Verfügbarkeit von statischen HTML-Dateien gewährleisten. Vielmehr werden heutzutage komplexe Online-Anwendungen in das WWW integriert, um eine verstärkt benutzerorientierte und bedarfsgesteuerte Informationsaufbereitung zu unterstützen. Entsprechend stellen viele Informationsanbieter im WWW eigene Server bereit, auf deren Inhalte nur unter Verwendung einer lokalen Suchschnittstelle (z. B. über ein HTML-Formular) zugegriffen werden kann. Die Gesamtanzahl derartiger Suchserver im WWW wird auf über 100.000 geschätzt – mit steigender Tendenz [10]!

Neben dem Publizieren im WWW kommt dem gezielten Wiederauffinden von Informationen (Information Retrieval) eine zentrale Bedeutung bei. So zählen Suchportale, wie AltaVista oder Google, zu den am häufigst frequentierten Seiten des WWW. Allerdings können derartige Suchdienste immer weniger mit dem Wachstum des WWW schritthalten, was sich darin zeigt, daß deren anteilige Erfassung der verfügbaren Datenmenge immer weiter abnimmt [77].

Dies wird primär dadurch bedingt, daß derartige Suchtechnologien einen zentralistischen Ansatz verfolgen, d. h. die Bearbeitung von Anfragen erfolgt unter Auswertung eines Web-weiten Index. Dabei geschieht das Auffinden und die Erfassung neuer Web-Seiten weitgehend automatisiert durch systematisches Auswerten und Verfolgen von Hyperlinks. Eine solche Strategie hat zur Folge, daß Web-Ressourcen, auf die keine Links gesetzt wurden, auch nicht explizit erfaßt werden können. Dies gilt insbesondere für Daten, die in lokalen Datenbanken von Suchservern abgespeichert sind und nur über eine Web-basierte Suchschnittstelle recherchiert werden können.

Der Teil des WWW, der sich einer rein linkbasierten Erfassung verschließt, wird auch als das „Unsichtbare Web“ bezeichnet. Jüngste Schätzungen gehen davon aus, daß die

insgesamt im Unsichtbaren Web verfügbare Datenmenge 400 bis 500 mal größer ist als die des indexierbaren bzw. „Sichtbaren“ Webs [10]. Erschwerend kommt hinzu, daß sich die meisten Suchserver im WWW nicht kooperativ gegenüber automatisierten Indexierungsversuchen verhalten bzw. diese sogar explizit unterbinden – sei es aus technischen oder wirtschaftlichen Gründen heraus.

Die Erschließung des Unsichtbaren Webs bildet eine der wichtigsten Anforderungen für das Web der Zukunft, um der rapide anwachsenden Datenmenge Herr zu werden [20], [116]. Dabei wird es zunehmend wichtiger, Techniken zur wissensbasierten Kontexterschließung zu integrieren, auch im Hinblick auf die von Tim Berners-Lee formulierte Vision eines „Semantischen Webs“[13].

Im Verteilten Information Retrieval werden Suchserver durch eine adäquate Beschreibung repräsentiert, die eine Abstraktion der tatsächlich enthaltenen Daten darstellt. Auf der Basis einer Menge von derartigen Suchserver-Repräsentanten soll abgeschätzt werden, welche Suchserver relevante Dokumente für eine gegebene Suchanfrage enthalten könnten. Anschließend werden die Suchanfragen an die selektierten Suchserver versendet und lokale Anfragen an diesen durchgeführt. Eine solche Vorgehensweise erhöht in erheblichem Maße die Skalierbarkeit von Systemen für Verteiltes Information Retrieval. Allerdings setzen existierende Selektionsstrategien eine explizite Kooperation von Seiten der Suchserver voraus (z. B. durch das Exportieren vollständiger Term-Statistiken), weshalb sich derartige Strategien nur in beschränktem Umfang auf das reale Szenario des WWW übertragen lassen. Meist initiieren existierende Metasuchsysteme im WWW deshalb Anfragen an sämtlichen integrierten Suchservern, was eine erhebliche Belastung von Netz- und fremden Systemressourcen zur Folge hat.

Die vorliegende Arbeit entstand im Rahmen eines Forschungsprojektes mit dem Titel: „Entwicklung eines Systems zur Strukturierung, Speicherung und Bereitstellung von Dokumenten als Teil einer Infrastruktur für digitale Bibliotheken“ (siehe [59] für einen Ergebnisbericht). Das Projekt wurde von der Deutschen Forschungsgemeinschaft im Rahmen des Programms zur Förderung des wissenschaftlichen Bibliothekswesens unterstützt. In dem Projekt entstanden unter anderem zwei konkrete Suchserver: der *Dissertationsserver* der Johann Wolfgang Goethe-Universität[1], sowie der *1848 - Flugschriftenserver*, über den historische Flugblätter, Aufrufe und Plakate aus der Revolution von 1848 im WWW zugänglich gemacht werden[2]. Erkenntnisse aus diesen Arbeiten, sowie Erfahrungen im Bereich Management verteilter Anwendungen [57], erwiesen sich als wertvolle Orientierungshilfe für die Behandlung des Forschungskomplexes verteiltes Retrieval.

[1] *Dissertationsserver*: http://dbib.uni-frankfurt.de/diss [14. Nov. 2001]
[2] *1848 Flugschriften-Server*: http://dbib.uni-frankfurt.de/1848 [14. Nov. 2001]

1.2. Konkretisierung der Problemstellung

In der vorliegenden Arbeit werden neue Techniken zur Erschließung und Selektion von Web-basierten Suchservern vorgestellt und evaluiert, um hieraus eine integrierte Architektur für Verteiltes Information Retrieval über nicht-kooperativen Suchservern abzuleiten. Dabei gilt es zu überprüfen, inwieweit die Informationsmenge des Sichtbaren Webs ausreicht, um eine effektive Erschließung des Unsichtbaren Webs zu unterstützen.

Grundsätzlich gilt, daß je umfangreicher die Informationsmenge ist, die über jeden Suchserver zur Verfügung steht, desto zuverlässiger kann auch eine Selektionsentscheidung vorgenommen werden. Außerdem muß die einheitliche Verwendung und Interpretation der vorhandenen Daten durch eine allgemeingültige Vorschrift zur Repräsentantenerzeugung sichergestellt werden. Einen zentralen Aspekt dieser Arbeit bildet deshalb die Möglichkeit zur umfangreichen und einheitlichen Auszeichnung von Suchservern mit beschreibenden Informationen – also mit Metadaten (Daten über Daten).

Im Rahmen dieser Arbeit sollen gemeinsame Eigenschaften von Web-basierten Suchservern identifiziert werden, um hieraus einen Satz an Metadaten abzuleiten, der dazu geeignet ist, Suchserver als Ganzes einheitlich zu charakterisieren. Hierdurch soll die Heterogenität existierender Suchserver überwunden werden, um somit Qualität und Skalierbarkeit von verteilten Suchen im WWW generell zu verbessern. In Anbetracht des Überangebotes an potentiell relevanten Informationen im WWW, ergibt sich auch die Notwendigkeit, Suchserver verstärkt nach qualitativen Kriterien zu bewerten. Deshalb sollen auch Metadaten mitberücksichtigt werden, die es erlauben, Aussagen über die Qualität bzw. die Reputation von Suchservern abzuleiten.

Der nächste Schritt besteht in der Erzeugung von konkreten Metadaten für spezifizierte Suchserver, wobei dieser Prozeß weitestgehend automatisiert durchgeführt werden soll. Da der Schwerpunkt dieser Arbeit auf nicht-kooperativen Suchservern liegt, sollen die Metadaten durch systematisches Auswerten von Informationen aus dem sichtbaren Teil des WWW gewonnen werden, indem die verteilte Indexierungsarbeit, die kollektiv im Sichtbaren Web geleistet wurde, systematisch ausgewertet wird. So bildet die Gesamtheit der im WWW gespeicherten Datenmenge eine globale Wissensbasis von beachtlicher Größe. Insbesondere die strukturelle und textuelle Analyse von Hyperlinks erlaubt es, sowohl qualitative Metadaten als auch inhaltsbezogene Metadaten automatisiert zu generieren, sofern die Suchserver in die globale Hyperlinkstruktur eingebunden sind (z. B. durch Links auf deren Einstiegs- oder Suchseite).

Anschließend muß unter Auswertung der gesammelten Metadaten die Selektion der potentiell relevantesten Suchserver zu einer gegebenen Suchanfrage durchgeführt wer-

den. Es ergibt sich hierbei allerdings das Problem, daß zusätzliches Kontextwissen über die Suchanfrage notwendig ist, um die geeignetsten Suchserver zu identifzieren, zumal sich die meisten Suchanfragen oft nur auf ein oder zwei Terme beschränken.

1.3. Aufbau der Arbeit

Im Anschluß an die Einleitung werden in Kapitel 2 Grundlagen behandelt, die für das Verständnis der Arbeit notwendig sind. Zu Beginn wird auf die wichtigsten Techniken aus dem Bereich Information Retrieval eingegangen. Da in dieser Arbeit das Unsichtbare Web unter Auswertung des Sichtbaren Webs erschlossen werden soll, wird die Struktur und die Größe des Sichtbaren Webs dargelegt, sowie dessen gebräuchlichsten Such- und Erschließungsverfahren vorgestellt und im Hinblick auf Vor- und Nachteile diskutiert. Anschließend werden verschiedene Web-Mining Technologien besprochen, wobei für die vorliegende Arbeit insbesondere jene Verfahren von Bedeutung sind, die auf der systematischen Analyse von Hyperlinks basieren. Das Kapitel schließt mit der Betrachtung von Metadaten und deren Anwendung im WWW.

Kapitel 3 beschäftigt sich mit den Eigenschaften von Verteiltem Information Retrieval. Zunächst erfolgt eine Diskussion der Eigenschaften des Unsichtbaren Webs. Anschließend werden existierende Systeme und Strategien für Verteiltes Information Retrieval vorgestellt und im Hinblick auf deren Eignung für das WWW untersucht. Hieraus werden konzeptionelle Anforderungen an ein System zum verteilten Information Retrieval über nicht-kooperativen Suchservern im WWW abgeleitet.

In Kapitel 4 wird ein neuer Metadatensatz für Web-basierte Suchserver vorgestellt – der Frankfurt Core. Dieser enthält Felder für Informationen, die dazu geeignet sind, einen Suchserver als Ganzes einheitlich zu beschreiben. Dies können sowohl inhaltlich beschreibende Terme, als auch qualitätsbezogene Informationen sein. Die Verwendung des Frankfurt Cores wird anhand eines vollständig ausgezeichneten Suchservers demonstriert.

Darüber hinaus werden für einzelne Felder des Frankfurt Core Strategien vorgestellt, um Metadaten automatisiert aus dem sichtbaren Teil des WWW zu extrahieren. Der Frankfurt Core unterstützt auch eine Systematik zur wissensbasierten und einheitlichen Repräsentation von Suchfeldern (z. B. *Autor*, *Titel* etc.) verschiedener Suchserver.

Zur Ergänzung der inhaltlichen Auszeichnung von Suchservern werden in Kapitel 5 Web-Mining Verfahren für Suchserver vorgestellt und anhand von Experimenten evaluiert. Die Verfahren basieren auf der strukturellen und inhaltlichen Analyse von Hyperlinks. Konkret wurde ein auf dem HITS- (Hyperlink Induced Topic Search) Algorithmus basierendes Clustering Verfahren sowie ein hyperlinkbasiertes Kategorisie-

rungsverfahren entwickelt.

Um zu einer gegebenen Suchanfrage geeignete Suchserver auf der Basis der gesammelten Metainformationen auszuwählen, wurde ein Selektionsverfahren entwickelt und durch geeignete Experimente evaluiert. Das hierzu benötigte Kontextwissen wird automatisiert durch Auswertung eines web-weiten Index gewonnen. Die Beschreibung dieses Selektionsverfahrens erfolgt in Kapitel 6.

Die gewonnenen Erkenntnisse und Erfahrungen werden dazu genutzt, um QUEST (QUErying Specialized collecTions on the web) – eine Metasuchmaschinen-Architektur für spezialisierte Web-basierte Suchserver – abzuleiten. Deren Architektur und Implementierung wird in Kapitel 7 dargelegt.

Abschließend faßt Kapitel 8 die Ergebnisse der Arbeit zusammen und gibt Anregungen für mögliche Erweiterungen.

2. Grundlagen

In diesem Kapitel werden Grundlagen behandelt, die für das Verständnis der Arbeit notwendig sind. Zunächst wird auf einige wichtige Techniken aus dem Bereich Information Retrieval eingegangen. Es werden bewährte Modelle (Boolesches- und Vektorraummodell) vorgestellt und häufig verwendete Gewichtungs- und Bewertungsmethoden erläutert. Anschließend wird die Struktur und die Größe des Sichtbaren Webs dargelegt, sowie dessen gebräuchlichsten Such- und Erschließungsverfahren (Kataloge, Suchmaschinen und Metasuchmaschinen) vorgestellt und diskutiert. Des weiteren erfolgt ein kurzer Überblick über Methoden des Web-Minings, wobei im Rahmen dieser Arbeit insbesondere jene Verfahren von Bedeutung sind, die auf einer textuellen und strukturellen Analyse von Hyperlinks basieren. Das Kapitel schließt mit der Betrachtung von Metadaten und deren Anwendung im WWW.

2.1. Information Retrieval

Im Gegensatz zu anderen Gebieten der Informatik gibt es für den Bereich des Information Retrieval (IR) keine allgemeingültige Definition. Die 1991 gegründete Fachgruppe *Information Retrieval* innerhalb der Gesellschaft für Informatik[1] sieht ihre Aufgabe allgemein in der Beschäftigung mit jenen Fragestellungen, die im Zusammenhang mit vagen Anfragen und unsicherem Wissen entstehen. Dabei wird davon ausgegangen, daß die Darstellungsform des in einem IR-System gespeicherten Wissens im Prinzip nicht beschränkt ist. Dies können z. B. Texte, multimediale Dokumente, Fakten, Regeln etc. sein. Die Unsicherheit dieses Wissens resultiert meist aus der begrenzten Repräsentation von dessen Semantik.

Bei IR-Systemen kann – im Unterschied zu anderen Informationssystemen, wie z. B. Datenbanksystemen – weder davon ausgegangen werden, daß eine Anfrage an ein IR-System nur exakt ein Informationsobjekt beschreibt, noch gibt es eine klare Definition, auf welche Art und Weise ein bestimmtes Informationsobjekt einer Anfrage zugeord-

[1] Gesellschaft für Informatik/Fachgruppe *Information Retrieval*: http://ls6-www.informatik.uni-dortmund.de/ir/fgir/ [14. Nov. 2001]

net werden kann. Eine detailliertere Abgrenzung von Information Retrieval gegenüber Daten-Retrieval findet sich in dem Buch „Information Retrieval“ von C. J. van Rijsbergen [113].

Zentrales Anliegen eines IR-Systems ist es, die *Relevanz* von Informationsobjekten (wir bezeichnen diese im folgenden als *Dokumente*) im Hinblick auf eine *Benutzeranfrage* einzuschätzen. Eine solche Relevanz-Entscheidung wird von dem zugrundeliegenden Gewichtungsalgorithmus vorgenommen. Dabei etabliert der Gewichtungsalgorithmus eine Ordnung über den Dokumenten einer Kollektion, wobei gilt: Je höher ein Dokument gewichtet wird, desto höher ist dessen Relevanz für eine Benutzeranfrage einzustufen.

Existierende Gewichtungsalgorithmen unterscheiden sich in der Interpretation des Relevanzbegriffs. Man unterscheidet diesbezüglich zwischen verschiedenen IR-Modellen, die jeweils von unterschiedlichen Prämissen ausgehen, wann ein Dokument als relevant für eine Anfrage einzustufen ist und wann nicht. So hängt die Relevanz zwischen Dokument und Anfrage zum einen von der Art und Weise ab, wie diese repräsentiert sind, und zum anderen von der verwendeten Strategie zur Ähnlichkeitsbestimmung.

Ein *IR-Modell* ist folgendermaßen definiert:

Ein IR-Modell ist ein 4-Tupel $(D, Q, F, R(q, d))$ mit:

1. D ist eine Menge von Repräsentationen von Dokumenten einer Kollektion.

2. Q ist eine Menge von Repräsentationen von Informationswünschen von Benutzern. Diese werden als Anfragen bezeichnet.

3. F ist ein Rahmenwerk zur Modellierung von Dokumentrepräsentationen, Anfragen und ihren Beziehungen.

4. $R(q, d)$ ist eine Retrievalfunktion, die zu einer gegebenen Anfrage $q \in Q$ und einer Dokumentrepräsentation $d \in D$ eine Zahl zuordnet. Die Gewichtung definiert eine Ordnung zwischen den Dokumenten aus D bzgl. der Anfrage q.

Die klassischen Modelle des Information Retrieval sind das Boolesche Modell, das Vektorraummodell und das Probabilistische Modell.

Den verschiedenen Modellen ist gemein, daß ein Dokument durch eine Menge von Schlüsselworten repräsentiert wird. Diese Schlüsselworte werden im folgenden als *Index-Terme* bezeichnet. Index-Terme sind textuelle Begrifflichkeiten oder Worte, die ein Dokument stellvertretend repräsentieren und deren innewohnende Semantik dessen Inhalt treffend charakterisiert. Der Vorgang des Zuordnens von Index-Termen zu

Dokumenten wird als Indexierung bezeichnet. Die Zuordnung kann manuell oder automatisiert erfolgen.

Gleiche Index-Terme können von Dokument zu Dokument unterschiedlich bedeutsam sein, z. B. hat der Index-Term *Flugzeug* für ein Dokument, das sich mit der Konstruktion von Flugzeugen beschäftigt, einen sehr hohen Beschreibungswert. Für ein Dokument hingegen, das sich allgemein mit unterschiedlichen Arten von Transportmitteln beschäftigt, kann der Index-Term *Flugzeug* zwar auch bedeutsam sein, jedoch nicht in dem Maße wie für ersteres. Diesem Sachverhalt wird dadurch Rechnung getragen, daß jedem Index-Term eines Dokumentes zusätzlich ein numerisches Gewicht zugeordnet wird, d. h. : Sei t_i ein Index-Term und d_j ein Dokument und $w_{i,j} \geq 0$ ein Gewicht, das mit dem Paar (t_i, d_j) assoziiert ist, dann gilt: Je höher das Gewicht $w_{i,j}$, desto bedeutsamer ist der Index-Term t_i für das Dokument d_j.

Sei n die Anzahl aller vorkommenden Index-Terme in einem System und m die Anzahl aller Dokumente im System. $T = \{t_1, \ldots, t_n\}$ ist die Menge aller vorkommenden Index-Terme und $D = \{d_1, \ldots, d_m\}$ die Menge aller Dokumente. Mit jedem Paar (t_i, d_j) ist ein Gewicht $w_{i,j} \geq 0$ assoziiert. Ein Dokument d_j ist durch einen Index-Term-Vektor $\vec{d_j}$ repräsentiert mit $\vec{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{n,j})$.

Aufgrund ihrer Bedeutung für das WWW und die vorliegende Arbeit wird im folgenden das Boolesche Modell und das Vektorraummodell kurz vorgestellt. Eine Beschreibung des probabilistischen Modells findet sich z. B. in [46].

2.1.1. Boolesches IR-Modell

Boolesches Retrieval stellt das historisch älteste IR-Modell dar und basiert auf Mengentheorie und Boolescher Algebra. Benutzeranfragen werden als boolescher Ausdruck spezifiziert, d. h. sie bestehen aus Termen, die mittels der boolescher Operationen ($\wedge$,$\vee$,$\neg$) gemäß den folgenden Regeln miteinander verknüpft werden können (Dabei ist Q die Menge der Anfragen und T die Menge der Index-Terme):

1. $t_i \in T \rightarrow t_i \in Q$
2. $q_1, q_2 \in Q \rightarrow q_1 \wedge q_2 \in Q$
3. $q_1, q_2 \in Q \rightarrow q_1 \vee q_2 \in Q$
4. $q \in Q \rightarrow \neg q \in Q$

Im Booleschen Modell wird davon ausgegangen, daß ein Index-Term in einem Dokument entweder vorhanden oder nicht vorhanden sein kann, d. h. die Index-Term-Gewichte sind binär: $w_{i,j} \in \{0, 1\}$. Die Retrievalfunktion $R(q_i, d_j)$ liefert nur die

Werte 1 oder 0 zurück, d. h. die Dokumente einer Menge D werden in eine Menge relevanter Dokumente ($R(q_i, d_j) = 1$) und eine Menge nicht relevanter Dokumente ($R(q_i, d_j) = 0$) aufgeteilt.

In IR-Systemen wird aus Gründen der Vereinfachung oft die Verwendung der Negation eingeschränkt, so daß sie nur in Kombination mit Konjunktionen verwendet werden darf, d. h. Anfragen wie $q = t_1 \vee \neg t_2$ oder $\neg t_1$ sind ungültig.

Die Größe der Antwortmenge beim Booleschen Retrieval kann unter Umständen sehr stark variieren. Ist die Anfrage zu restriktiv, werden nur wenige Dokumente gefunden, ist sie zu generell, kann die Antwortmenge auf ein unübersichtliches Maß anwachsen. Ein weiteres Problem besteht darin, daß keine Ordnung auf den Dokumenten der Antwortmenge definiert ist, um die Antwortmenge nach stark relevanten und weniger relevanten Dokumenten zu strukturieren. In der Praxis werden deshalb verschiedene Maßnahmen getroffen, um nachträglich die gefundenen Dokumente nach bestimmten Kriterien zu gewichten. Häufig verwendete Kriterien sind:

- *Häufigkeit der vorkommenden Index-Terme* Dokumente, in denen die meisten Index-Terme in der Suchanfrage vorkommen, werden zuerst aufgelistet.

- *Benutzerdefiniertes Gewicht* Der Benutzer kann beim Spezifizieren der Suchanfrage zu jedem Term ein Gewicht angeben, nach denen das IR-System die Treffermenge sortiert.

- *Relative Positionen von Index-Termen* Es werden zusätzlich die relativen Positionen verschiedener Index-Terme der Anfrage im Dokument berücksichtigt (z. B. bei AltaVista +/- 10 Wörter). Tauchen in den Wortumgebungen weitere Index-Terme auf, werden diese Dokumente höher gewichtet.

- *Berücksichtigung von Dokumentstrukturen* Falls vorhanden, werden existierende Dokumentstrukturen berücksichtigt, wie sie z. B. in HTML- oder XML-Dokumenten gegeben sind. So werden oft Dokumente, in denen die Index-Terme innerhalb von Titeln gefunden werden, höher gewichtet.

2.1.2. Das Vektorraummodell

Die von Booleschen IR-Systemen vorgenommene Trennung in relevante und nicht relevante Dokumente wird oft als zu restriktiv empfunden. Zu einer Benutzeranfrage $q = t_1 \wedge t_2 \wedge t_3$ werden neben Dokumenten, die keinen der Terme enthalten, auch solche zurückgewiesen, die einen oder zwei der Anfrage-Terme enthalten. Außerden sind Benutzer erfahrungsgemäß oft mit der Semantik boolescher Operatoren nicht

ausreichend vertraut, wodurch deren Handhabung bei der Formulierung von Anfragen erschwert wird.

Dem Vektorraummodell hingegen liegt eine tolerantere Interpretation des Relevanzbegriffes zugrunde, und es kommt ohne boolesche Operatoren aus. Dies wird erreicht durch die Verwendung nicht-binärer Werte für die Term/Dokument-Gewichte $w_{i,j}$. Das Vektorraummodell wurde im Rahmen des experimentellen IR-Systems SMART [103], [105] entwickelt. Zentrale Idee ist, daß Anfragen und Dokumente als Punkte in einem Vektorraum der Dimension n aufgefaßt werden ($n =\mid T \mid$).

Definition:

Im Vektorraumodell ist mit einem Paar (t_i, d_j) ein positives, nicht-binäres Gewicht $w_{i,j}$ assoziiert. Die Index-Terme der Suchanfrage q sind ebenfalls gewichtet. Dabei sei $w_{i,q}$ das Gewicht, das mit dem Paar (t_i, q) assoziiert ist, mit $w_{i,q} \geq 0$. Die Anfrage wird repräsentiert durch den Anfragevektor $\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{n,q})$, dabei ist n die Gesamtanzahl von Index-Termen im System. Der Vektor eines Dokumentes $\vec{d_j}$ ist repräsentiert durch $\vec{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{n,j})$.

Durch in der Linearen Algebra gebräuchliche Ähnlichkeitsmaße für Vektoren, wie beispielsweise das Kosinus-Maß (siehe z. B. [7]), wird der Grad der Korrelation zwischen Anfrage und Dokument bestimmt und damit eine Ordnung der Dokumente nach absteigender Relevanz vorgegeben. Das Kosinusmaß entspricht dem Skalarprodukt der normierten Vektoren, und die Ähnlichkeitswerte werden durch die Richtungen der Vektoren bestimmt.

2.1.3. Gewichtungsmethoden

Ein weiteres Problem ist, wie die einzelnen Gewichte der Terme in Anfragen und Dokumenten, d. h. die Werte $w_{i,q}$ und $w_{i,j}$, bestimmt werden können. Eine intellektuelle Bestimmung der einzelnen Gewichte ist zu aufwendig, weshalb dies automatisiert vorgenommen werden soll.

Bei der Bestimmung von Termgewichten kann man zwischen lokalen und globalen Gewichtungskriterien unterscheiden. Ein lokales Kriterium wäre z. B. die Häufigkeit mit der ein bestimmter Term in einem bestimmten Dokument auftaucht, ein globales dahingegen wäre die Häufigkeit, mit der ein Term in einer bestimmten Sprache auftaucht.

2.1.3.1. Globale Gewichtung

Die Verteilung der Wörter in einer Sprache kann grob durch das Zipf'sche Gesetz beschrieben werden [130].

Zipf'sches Gesetz: Für einen repräsentativen Textcorpus C bezeichne $W(C)$ die Menge der Wörter, die in C vorkommen, und $h(w)$ die Häufigkeit, mit der das Wort $w \in W(C)$ in dem Korpus vorkommt. $r(w)$ bezeichne den Rangplatz von $w \in W(C)$, wenn die Wörter nach abfallender Häufigkeit sortiert werden. Dann besagt das Zipf'sche Gesetz, daß eine Konstante c existiert, mit $r(w) \cdot h(w) \sim c$ $\forall w \in W(C)$.

Aus dem Zipf'schen Gesetz ergibt sich, daß eine kleine Anzahl von häufigen Wörtern einen großen Anteil der Texte abdeckt, und die große Anzahl der seltenen Wörter nur einen kleinen Teil des Textes ausmacht. Häufige Terme sind keine guten Indexterme, weil sie nicht spezifisch für ein bestimmtes Dokument sind. Damit sind sie gute Kandidaten für Stopworte, d. h. Worte, die aus den Dokumenten und Anfragen herausgefiltert werden, wodurch sich die Dimension der Vektoren reduziert.

Die Verwendung von Stopworten erleichtert zwar die Berechnung der Ähnlichkeitswerte und bewirkt eine Reduktion des Index, jedoch gehen auch u. U. wichtige Deskriptoren verloren. Ein bessere Strategie stellt daher die Abschwächung häufiger Terme durch ein entsprechendes Gewicht dar. In der Praxis wird deshalb häufig die *inverse Dokumenthäufigkeit (inverse document frequency, idf)* für einen Term t_j verwendet (idf_j) [103]. Dabei bezeichnet $D = \{d_1, \ldots, d_m\}$ die Menge aller Dokumente, $T = \{t_1, \ldots, t_n\}$ die Menge aller Terme und df_j die Anzahl der Dokumente, in denen ein Term t_j vorkommt. Durch die Logarithmus-Funktion werden große Werte gedämpft; die Gewichte seltener Terme werden also wieder abgeschwächt:

$$idf_j = log(\frac{m}{df_j})$$

2.1.3.2. Lokale Gewichtung

Bei lokalen Gewichtungskriterien wird vor allem die Häufigkeit von Termen tf (*term frequency*) innerhalb der einzelnen Dokumente zur Bestimmung von Gewichten herangezogen. Die zugrundeliegende Annahme ist die, daß Terme, die häufiger in einem bestimmten Dokument auftauchen, bessere Deskriptoren für das Dokument darstellen, als selten vorkommende Terme.

Um den Einfluß der unterschiedlichen Länge von Dokumenten auszugleichen, wird die Häufigkeit eines Terms t_i in einem Dokument d_j zum Maximum aller Termhäufigkeiten in d_j in Bezug gesetzt. Sei $tf_{i,j}$ die absolute Häufigkeit, mit der ein Term t_i in einem Dokument d_j vorkommt, dann ist die *normalisierte Termhäufigkeit* $ntf_{i,j}$ von t_i in d_j ist gegeben durch:

$$ntf_{i,j} = \frac{tf_{i,j}}{max_{l \in \{1,\ldots,n\}} tf_{l,j}}$$

Dabei wird das Maximum über alle Terme ermittelt, die im Dokument d_j auftauchen.

Andere lokale Gewichtungsmethoden nutzen Informationen über die Struktur der Dokumente, falls diese explizit ausgezeichnet ist (z. B. in HTML- oder XML-Dokumenten). So kann z. B. ein Term, der in einem Titel auftaucht, höher gewichtet werden als einer, der in einer Fußnote auftaucht.

2.1.3.3. tf-idf Gewichte

Oft werden lokale und globale Gewichtungen miteinander kombiniert [104]. Die am häufigsten verwendeten Gewichte $w_{i,j}$ zur Bestimmung der Deskriptionsfähigkeit eines Index-Term t_i für ein Dokument d_j sind die sogenannten *tf-idf* Gewichte. Diesen liegt die folgende Formel oder eine Variante hiervon zugrunde:

$$w_{i,j} = ntf_{i,j} \cdot idf_i$$

Die Gewichte $w_{i,q}$ für eine Anfrage q können ebenfalls über tf-idf basierte Gewichte berechnet werden. Im SMART-System [103] wird z. B. die folgende Formel zur Bestimmung der Anfragegewichte verwendet:

$$w_{i,q} = (0,5 + \frac{0,5 \cdot tf_{i,q}}{max_{l \in \{1,\ldots,n\}} tf_{i,q}}) \cdot idf_i$$

Dabei bezeichnet $tf_{i,q}$ die absolute Häufigkeit, mit der ein Term t_i in der Anfrage q vorkommt.

2.1.4. Metriken zur Bewertung der Retrievalqualität

Die Herleitung von Metriken zur Bestimmung der Retrievalqualität von IR-Systemen erfordert eine Konkretisierung des Relevanzbegriffs: Durch die Relevanz wird eine Beziehung zwischen Anfragen und Dokumenten bestimmt. Diese ist formal durch die

folgende Relation definiert:

Die Relevanz eines Dokumentes für eine Anfrage ist gegeben durch eine Relation $r : D \times Q \to W$, wobei D die Menge der Dokumente, Q die Menge der Suchanfragen und W eine Menge von Wahrheitswerten (i. A. die Menge $\{0, 1\}$) darstellt.

Die Relation r kann z. B. durch das Befragen von Experten zu konkreten Anfragen $q \in Q$ bestimmt werden. Diese bestimmen jene Dokumente aus D, die sie im Hinblick auf die Anfragen aus Q als relevant einschätzen. Ist r a priori bekannt, lassen sich vergleichende Aussagen über die Retrievalqualität verschiedener IR-Systeme ableiten. Die beiden am häufigsten verwendeten Metriken sind *Abdeckung* (engl.: Recall) und *Präzision* (engl.: Precision). Diese sind folgendermaßen definiert:

Sei $D = \{d_1, \ldots, d_m\}$ eine Menge von Dokumenten, $q \in Q$ eine Anfrage und D_q die zur Anfrage q gefundenen Dokumente. Sei ferner $r : D \times Q \to W$ eine Relevanzrelation, und $r_q : D \to \{0, 1\}$ mit $r_q := r(q, d)$ die zur Anfrage q gehörende Relevanzfunktion, dann heißt:

$$Prec(q, D) := \frac{\mid D_q \cap r_q^{-1}(\{1\}) \mid}{\mid D_q \mid}$$

Präzision des Ergebnisses auf die Anfrage q und

$$Rec(q, D) := \frac{\mid D_q \cap r_q^{-1}(\{1\}) \mid}{\mid r_q^{-1}(\{1\}) \mid}$$

Abdeckung des Ergebnisses auf die Anfrage q.

$Prec(q, D)$ gibt den Anteil der relevanten Dokumente unter den gefundenen Dokumenten an: je niedriger die Präzision, desto höher ist der Anteil irrelevanter Dokumente in der Antwortmenge, und desto schwerer fällt es, relevante Dokumente herauszufiltern.

$Rec(q, D)$ dahingegen bestimmt den Anteil der relevanten Dokumente, die tatsächlich auch gefunden wurden. Der optimale Wert für Präzision und Abdeckung (also gleich 1) wird erreicht, wenn $D_q = r_q^{-1}(\{1\})$ gilt, also genau alle relevanten Dokumente als Antwortmenge zurückgeliefert werden.

Die beiden Maße sind einander gegenläufig. Dies läßt sich anhand zweier Extremfälle verdeutlichen:

1. Wenn $D_q = D$ gilt, wenn also alle Dokumente auf eine Anfrage hin zurückgeliefert werden, ist die Abdeckung gleich 1, da sich natürlich auch alle relevanten Dokumente unter der Antwortmenge befinden. Da die Antwortmen-

ge damit aber fast nur aus irrelevanten Dokumenten besteht, wird die Präzision dementsprechend niedrig ausfallen.

2. Wird umgkehrt nur ein einziges relevantes Dokument gefunden, so ist die Präzision gleich 1, aber die Abdeckung wird dementsprechend niedrig ausfallen, da die meisten relevanten Dokumente nicht gefunden wurden.

Im Vektorraummodell können die Größen der Antwortmengen und damit die Werte für Abdeckung und Präzision über einen Ähnlichkeitsschwellwert variabel gehalten werden, d. h. nur Dokumente die über einem festgelegten Ähnlichkeitswert liegen, werden als relevant eingestuft. Zur Bestimmung von Werten für Abdeckung und Präzision stehen eine Reihe von Testkollektionen zur Verfügung z. B. TREC (siehe [7], Kapitel 3.3 für eine Übersicht). Diese bestehen in der Regel aus einer Sammlung von Dokumenten, einer Sammlung von Anfragen und den (durch Experten bestimmten) Angaben, welche Dokumente aus der Kollektion für jede einzelne Anfrage relevant sind.

2.2. Struktur des WWW

2.2.1. Größe und Wachstum des WWW

Aufgrund des evolutionären Wachstums des WWW sowie dessen dezentraler Struktur erweist sich die Abschätzung der enthaltenen Gesamtdatenmenge als äußerst schwieriges Unterfangen. Dennoch gilt: Sind Größe und Wachstum bekannt, ermöglicht dies z. B. vergleichende Aussagen über den Grad der Abdeckung des WWW durch die verschiedenen generellen Web-Suchmaschinen wie AltaVista[2], Northernlight[3], Google[4] oder Fast[5]

Existierende Studien messen meist nur den sichtbaren Teil des WWW. Hierunter versteht man statische Web-Seiten, die öffentlich zugänglich sind und von Web-Crawlern automatisiert erfaßt und indexiert werden können. Im „Sichtbaren Web“ bleiben folgende Daten unberücksichtigt:

1. Web-Seiten, die nur autorisierten Benutzern zugänglich sind, z. B. in Intranets oder in passwortgeschützten Seiten,

2. Web-Seiten, für die unter Verwendung des Robot-Exclusion Protokolls oder mittels HTML-META-Tags klargestellt wurde, daß eine automatisierte Erfassung

[2] AltaVista: http://www.altavista.com [14. Nov. 2001]
[3] Northernlight: http://www.northernlight.com [14. Nov. 2001]
[4] Google: http://www.google.com [14. Nov. 2001]
[5] Fast: http://www.alltheweb.com [14. Nov. 2001]

nicht erwünscht ist[6], und

3. Web-Seiten, die dynamisch aufgebaut werden, z. B. als Ergebnis einer formularbasierten Suchanfrage erzeugt. Dieser Teil des WWW wird auch als das „Unsichbares Web“ bezeichnet und eingehend in Abschnitt 3.1 erläutert.

In den Studien wird zum einen versucht die Anzahl verschiedener Web-Server abzuschätzen und zum anderen die Anzahl der hierüber verfügbaren statischen Web-Seiten zu bestimmen.

Eine vielbeachtete Studie von Lawrence und Giles [77], durchgeführt im Februar 1999, schätzt die Anzahl betriebener Web-Server auf 2.8 Mio. Diese Zahl wurde durch systematisches Ausprobieren aller IP-Adressen ermittelt ($256^4 \approx 4.3Mrd$). Von den identifizierten Web-Servern wurden 2500 zufällig ausgewählt und die Anzahl der pro Server verfügbaren Seiten ermittelt. Diese betrug im Durchschnitt 289. Multipliziert mit der Anzahl der geschätzten Web-Server ergibt sich somit eine Gesamtanzahl von etwa 809 Mio Web-Seiten. Mehr als ein Jahr zuvor, im Dezember 97, wurde von derselben Gruppe eine Gesamtanzahl von 320 Mio Web-Seiten ermittelt [78], d. h. die Anzahl der Web-Seiten hat sich innerhalb eines Jahres in etwa verdoppelt. Ungefähr 6% aller Web-Server werden von Universitäten, Forschungseinrichtungen oder sonstigen wissenschaftlichen Institutionen betrieben.

Es ist klar, daß ein solches Web-Server basiertes Meßverfahren mit großen Unsicherheitsfaktoren behaftet ist. So gibt es einige wenige Web-Server, die eine sehr hohe Zahl an Web-Seiten bereitstellen (GeoCities stellt z. B. Millionen von Seiten bereit) und eine sehr große Anzahl von Web-Servern, die nur sehr wenig Seiten verwalten.

Eine weitere Studie jüngeren Datums (Januar 2000) [7] ergab eine Anzahl von 4,2 Mio Web-Servern[8].

In den zitierten Studien werden die Bezeichnungen Web-Server und Web-Site oft synonym verwendet. Dennoch ist es für die vorliegende Arbeit notwendig, diese beiden Bezeichnungen voneinander abzusetzen, indem für Web-Sites zusätzliche Eigenschaften spezifiziert werden. In Anlehung an [4] und [34] wird eine Web-Site deshalb folgendermaßen charakterisiert:

Eine Web-Site ist eine organisierte Sammlung von Seiten zu einem spezifischen Thema, die von einzelnen Personen oder Gruppen gepflegt wird. Web-Sites verfügen über spezielle Seiten, die die Strukturierung der Web-Site unterstützen. Dies kann z. B. eine Start- oder Einstiegsseite sein (engl.: Front-Page), ein Inhaltsverzeichnis oder eine

[6] siehe z. B. http://www.robotstxt.org/wc/exclusion.html [14. Nov. 2001]
[7] Inktomi WebMap: http://www.inktomi.com/webmap/ [14. Nov. 2001]
[8] Mirror-Server, sowie Server, die über eine 10 Tage währenden Zeitraum nicht erreichbar waren, wurden hiervon bereits abgezogen.

Suchschnittstelle, über die die Seiten der Web-Site durchsucht werden können.

2.2.2. Die Hyperlinkstruktur des WWW

In [23] wird die bis dato umfangreichste Analyse der Hyperlink-Struktur des WWW beschrieben. Dabei wurde das WWW als Graph modelliert; Web-Seiten entsprechen Knoten und ein Hyperlink von einer Seite u zu einer Seite v entspricht einer gerichteten Kante (u, v). Die Grundlage für diese Untersuchung bildeten zwei im Mai 1999 und Oktober 1999 vorgenommene Crawls von AltaVista, von denen jeder etwa 200 Mio Web-Seiten und 1,5 Mrd Links umfaßte.

In den Experimenten wurde der Web-Graph hinsichtlich des Vorhandenseins zusammenhängender Komponenten untersucht. Dabei wurde unterschieden zwischen stark und schwach zusammenhängenden Komponenten. Unter einer *stark zusammenhängenden Komponente (SCC)* versteht man eine Menge von Knoten, wobei für jedes Knotenpaar (u, v) aus dieser Menge gilt, daß ein Pfad von u nach v existiert. Zusammenhängende Knotenmengen in einem ungerichteten Graphen bezeichnet man als *schwach zusammenhängende Komponenten (WCC)*. Den ungerichteten Web-Graph G' gewinnt man aus dem gerichten Web-Graph G, indem man eine existierende Kante (u, v) in G als ungeordnetes Paar in G' auffaßt.

G und G' wurden mit wechselnden Startpunkten mittels einen Breadth-First Search Algorithmus traversiert. Dabei kristallisierte sich die makroskopische Graphstruktur des Sichtbaren Webs heraus. Diese besteht im wesentlichen aus vier Haupt-Komponenten:

- einer SCC mit insgesamt etwa 56 Mio Knoten,
- einer Menge von etwa 44 Mio Knoten, die in die SCC hineinverweisen (IN),
- einer Menge von etwa 44 Mio Knoten, auf die von der SCC heraus verwiesen wird (OUT), sowie
- sogenannte Tendrils; dies sind Seiten, auf die von IN-Seiten verwiesen wird bzw. Seiten, die in OUT-Seiten verweisen, die aber selber nicht Bestandteil der SCC sind (insgesamt etwa 44 Mio).

Durch die Studie konnte eine Behauptung von [2] widerlegt werden. Diese besagte, daß sich das Web im Sinne einer Small World verhält und jede Web-Seite von fast jeder anderen mit durchschnittlich 19 Clicks erreicht werden kann. Aus der aktuellen Studie ergab sich ein wesentlich differenzierteres Bild: In 75% aller Fälle existiert kein gerichteter Pfad zwischen zwei zufällig ausgewählten Web-Seiten. Existiert tatsächlich

einer, so beträgt seine Länge im Durchschnitt 16 im gerichteten und 7 im ungerichteten Graph.

2.3. Suchen im WWW

Insgesamt lassen sich grob drei Klassen von suchunterstützenden Strategien für das WWW unterscheiden. Hierbei handelt es sich um Kataloge, Suchmaschinen und Metasucher.

2.3.1. Kataloge

Seit dem 19. Jahrhundert verwenden Bibliothekare Klassifizierungssysteme wie die Dewey Decimal Classification[9], um die anfallende Informationsvielfalt in Bibliotheken zu organisieren und zu strukturieren. Web-Kataloge stellen die Übertragung dieser Idee auf die Bedürfnisse des WWW als Instanz einer globalen Bibliothek dar.

Web-Kataloge beruhen auf einem System vordefinierter und hierarchisch geordneter Schlagworte. Neue Webseiten werden redaktionell bearbeitet und von menschlichen Indexierern den einzelnen Kategorien zugeordnet. Innerhalb der ausgewählten Kategorien werden Links zu der Web-Seite sowie kurze Inhaltsbeschreibungen bereitgestellt. Den URLs der neu zu erfassenden Web-Seiten werden den Katalogbetreibern meist von den Autoren selbst über e-Mail Zuschriften bekanntgegeben. Zum Einsatz kommen aber auch Crawler-Programme zum Aufspüren neuer, populärer Web-Seiten.

Kataloge erleichtern die Navigation und ermöglichen das Auffinden von Dokumenten hoher Qualität und thematischer Relevanz im Hinblick auf einen Informationswunsch. Allerdings stellt die Kategorisierung der Web-Seiten durch menschliche Fachkräfte einen arbeitsintensiven Prozeß dar und kann deshalb nur schwer mit dem Wachstum des WWW Schritt halten.

Der Katalog von LookSmart[10] beinhaltete im August 2000 laut Searchenginewatch[11] beispielsweise nur 2 Mio Web-Seiten. Der Katalog von Yahoo[12] – einer der ältesten und bekanntesten Web-Kataloge – wies zum selben Zeitpunkt nur 1,5 bis 1,8 Mio kategorisierter Web-Seiten aus.

Problematisch ist auch das System von Kategorien selber. Der Medienforscher Hartmut Winkler schreibt hierzu [120]:

Die Konstruktion der Hierarchie erscheint als einigermaßen hybrides Projekt, zielt es

[9] Dewey Decimal Classification: http://www.oclc.org/dewey/ [14. Nov. 2001]
[10] LookSmart: http://www.looksmart.com/ [14. Nov. 2001]
[11] Searchenginewatch: http://www.searchenginewatch.com [14. Nov. 2001]
[12] Yahoo: http://www.yahoo.com [14. Nov. 2001]

doch darauf ab, Millionen völlig heterogener Netzbeiträge aus nahezu allen Bereichen der menschlichen Wissensbestände auf ein einheitliches Kategoriensystem zu bringen, ungeachtet ihrer Perspektivität, ihrer Widersprüche und Konkurrenzen.

Es zeigt sich also: Mit wachsendem Umfang des WWWs können interdisziplinäre Dokumente oder sich neu etablierende Themengebiete nicht mehr eindeutig in die bestehende Taxonomie eingeordnet werden. Beispielsweise kann die Kategorie *Umweltverschmutzung* sowohl als Unterkategorie von *Gesellschaft* als auch von *Natur* gesehen werden. Ebenso können Dokumente, die sich mit Bioinformatik beschäftigen, mehr als einer Kategorie zugehören.

2.3.2. Suchmaschinen

2.3.2.1. Inhaltserstellung

Zentrale Bestandteile einer Suchmaschinenarchitektur sind sogenannte *Crawler* (auch Robots, Spider, Wanderer, Walker oder Knowbots). Diese traversieren das WWW und versenden neu entdeckte Seiten, deren Inhalte sich seit dem letzten Besuch geändert haben, an einen zentralen Server. Der Server verwaltet einen Index, der sich über den für die Crawler sichtbaren Teil des WWW erschließt. Dieser wird gemäß der von den Crawlern entdeckten Inhalten erweitert und aktualisiert.

Über eine Web-basierte Suchschnittstelle können Benutzer Anfragen formulieren, die unter Auswertung des Indexes beantwortet werden. Auf der zurückgelieferten Ergebnisseite werden die gefundenen Dokumente nach absteigender Relevanz angeordnet. Dabei sind die Dokumente durch einen Link auf das Originaldokument sowie durch einen kurzen beschreibenden Text – einen sogenannten *Snippet* – repräsentiert. Ein solcher Snippet wird automatisch generiert und besteht meist aus dem Titel des Dokumentes, den ersten Zeilen des Volltextes, evtl. vorhandenen Metainformationen, die an ausgezeichneten Stellen im Dokument bereitgestellt wurden, oder den Textumgebungen, in denen der oder die Suchbegriffe gefunden wurden.

Die Crawler gehen beim Traversieren des WWW nach unterschiedlichen Strategien vor. Ausgehend von einer initialen Menge von Uniform Resource Locators (URLs), die als Startadressen fungieren, werden aus den Hyperlinks der besuchten Seiten neue URLs extrahiert. Diese werden dann mittels Breadth-First Search (BFS) oder Depth-First Search (DFS) rekursiv abgearbeitet. Die meisten der in Dokumenten vorgefundenen Links sind reine Navigationshilfen und verweisen somit auf Seiten, die auf demselben Server aufliegen. Dementsprechend bewirkt die Anwendung der DFS-Strategie eine Bombardierung des betroffenen Servers durch Anfragen vom Crawler. Deshalb wenden Crawler meist die BFS-Strategie an, da sie Anfragen auf denselben Server

zeitlich breiter streut.

Manche Crawler erfassen nur eine begrenzte Anzahl von Seiten eines Servers, andere wiederum erfassen sämtliche erreichbaren Seiten. Dabei besucht ein Crawler einen Server regelmäßig alle ein bis zwei Monate, um seinen Index zu aktualisieren. Deshalb zeigen die meisten Suchmaschinen den Zeitpunkt an, zu dem eine Seite indexiert wurde. Man geht in der Regel davon aus, daß 2-9% aller von den Suchmaschinen gespeicherten Links ungültig sind, also auf Dokumente verweisen, die nicht mehr vorhanden sind.

2.3.2.2. Eigenschaften

Suchmaschinen verwenden meist Varianten des Booleschen Modells und des Vektorraummodells (siehe Abschnitt 2.1.1 und 2.1.2) zur Berechnung der Relevanzgewichte von Anfrage und Dokument. Wie das Gewicht im Detail berechnet wird, wird von den Suchmaschinenbetreibern allerdings meist unter Verschluß gehalten.

In den bereits zitierten Studien [77], [78] wurde neben der Größe des Sichtbaren Webs auch der Grad der Abdeckung von generellen Suchmaschinen wie AltaVista, Northernlight und Google im Verhältnis zur Gesamtgröße des WWW betrachtet. Die Bestimmung der Anzahl der Web-Seiten, die pro Suchmaschine erfaßt sind, kann mittels der in [14] beschriebenen Überlappungstechnik durchgeführt werden. Im Dezember'97 deckten die Suchmaschinen zusammengenommen etwa 60% der insgesamt 310 Mio Web-Seiten ab, wobei die größte der einzelnen Suchmaschinen etwa ein Drittel hiervon abdeckte. Im Februar '99 dahingegen deckten die Suchmaschinen zusammengenommen nur noch 42% der 800 Mio Web-Seiten ab, die Abdeckung der größten Einzelsuchmaschine betrug nur noch 16%. Im November 2000 hatten die größten Suchmaschinen laut Searchenginewatch insgesamt etwa 600 Mio Dokumente erfaßt [111].

Es zeigt sich also, daß die klassischen Crawler-/Indexer Architekturen immer weniger mit dem Wachstum des WWW Schritt halten können. Zur generellen Einschätzung der Qualität von Suchmaschinen sei erneut ein Zitat von Hartmut Winkler gegeben [120]:

Eine mechanische Stichwortsuche setzt voraus, daß nur solche Fragen gestellt werden, die in Stichworten klar formulierbar sind und durch weitere Stichworte differenziert und konkretisiert werden können. Ebenso wird niemand erwarten, daß das System neben dem gefragten auch bedeutungsähnliche Begriffe einbeziehen oder Homonyme ausschließen kann. (...) Alle Fragen, die auf Stichworte nicht zu reduzieren sind, fallen aus dem Raster des Möglichen heraus; technische und naturwissenschaftliche Termini werden sich relativ gut für die Suche eignen, geisteswissenschaftliche Themen weit weniger gut.

2.3.3. Alternative Suchansätze

Die beiden im WWW dominanten Suchansätze, Kataloge und Suchmaschinen verhalten sich gegensätzlich gegenüber den Bewertungskriterien Abdeckung und Präzision. Der Suchbereich von Suchmaschinen deckt zu einem hohen Maß alle im Web vorhandenen (statischen) Dokumente ab – Suchmaschinen gewährleisten also einen hohen Abdeckungs-Wert. Aufgrund der oft sehr großen Treffermengen besitzen sie allerdings nur eine geringe Präzision, da die relevanten gefundenen Seiten aus der Masse der irrelevanten gefundenen Seiten nur schwer zu extrahieren sind. Kataloge dagegen bieten in Kategorien eingeordnete Links zu Web-Sites von hoher Qualität und hoher thematischer Relevanz, der Präzisions-Wert ist also sehr hoch. Allerdings decken die erfaßten Seiten nur einen Bruchteil der tatsächlich im WWW potentiell vorhandenen Seiten zu einem bestimmten Thema ab.

Als Kompromiß bieten die meisten Portale heute deshalb hybride Lösungen an, d. h. parallel sowohl einen Katalog als auch eine Suchmaschine.

Neben Katalogen und Suchmaschinen lassen sich noch eine Reihe alternativer Suchdienste für das WWW identifzieren, die im folgenden der Vollständigkeit halber kurz vorgestellt werden sollen:

- Natürlichsprachliche Suchdienste: Anfragen können natürlichsprachlich formuliert werden. Der Suchdienst wählt einige Seiten aus, die die Frage exakt beantworten. Fragen und Antwortseiten werden durch redaktionelle Bearbeitung einander zugeordnet. Natürlichsprachliche Suchdienste stellen somit nur eine Sonderform der Web-Kataloge dar. Bekanntestes Beispiel hierfür ist der Suchdienst AskJeeves[13].

- Benutzerverhaltensorientierte Suchdienste: Dienste wie DirectHit[14] messen, welche Web-Sites von wieviel Benutzern besucht wurden (z. B. durch Analyse von Log-Files). Gemessen wird auch die Dauer von Aufenthalten auf den einzelnen Sites. Häufig frequentierte Seiten erhalten eine höhere Gewichtung als selten besuchte.

- Bezahlte Suchdienste: Firmen bezahlen die Suchdienstbetreiber dafür, daß deren Seiten bei bestimmten Suchanfragen höher gewichtet werden.

[13] AskJeeves: http://www.askjeeves.com [14. Nov. 2001]
[14] DirectHit: http://www.directhit.com [14. Nov. 2001]

2.3.4. Metasucher

Studien (siehe z. B. [14]) zeigen, daß man den Suchraum im WWW erheblich erweitern kann, wenn man die Ergebnisse verschiedener Suchmaschinen kombiniert. Die Metasuch-Methode stellt eine direkte Umsetzung dieser Idee dar. Als Front-End wird eine einzige Suchschnittstelle bereitgestellt, unter der verschiedene Back-End Suchmaschinen zusammengefaßt sind. Benutzeranfragen werden durch das Metasuchsystem an die Back-End-Suchmaschinen weitergeleitet, die zurückgelieferten Ergebnisse werden zusammengeführt und in Form einer integrierten Resultatliste an den Benutzer zurückgeliefert.

Populäre Beispiele für Metasucher, die die Ergebnisse mehrerer genereller Web- Suchmaschinen kombinieren, sind SavvySearch [64], Metacrawler ([107], [108]) oder Highway61[15]. Derartige Metasuchmaschinen verwalten in der Regel keinen eigenen Index, vielmehr nutzen sie durch das automatisierte Abfragen der individuellen Suchschnittstellen die bereigestellten Dienste der autonomen Back-End-Suchdienste aus. Dabei lassen sich durch die Parallelisierung von Anfragen (z. B. durch das Verwenden von leichtgewichtigen Prozessen) die Antwortzeiten des Metasuchsystems erheblich reduzieren. Dennoch liegt die Anzahl der pro Suchanfrage abgefragten Suchmaschinen meist unter 20, um akzeptable Antwortzeiten ($<$ 15 Sekunden) zu gewährleisten. Manche Metasuchmaschinen erlauben auch die benutzergesteuerte Selektion von abzufragenden Suchmaschinen aus einer vorgegebenen Liste von Suchdiensten, wie beispielsweise die an der Universität Hannover betriebene Metasuchmaschine MetaGer[16].

Um die Treffer der einzelnen Suchmaschinen auszulesen, werden oft sogenannte „Screen Scraping“-Verfahren angewendet, d. h. die einzelnen Resultate müssen nach individuell festzulegenden Mustern aus den dynamisch generierten Resultatseiten extrahiert werden.

Zentrale Bestandteile einer Metasuchmaschinen-Architektur sind Wrapper und Mediatoren[119]. Diese erlauben eine technische Homogenisierung der einzubindenen Dienste. Mediatoren nehmen die über die Suchschnittstellen spezifizierten Anfragen entgegen, initiieren eine Reihe von Wrapperaufrufen und nehmen die von den Wrappern erzeugten Resultatlisten entgegen. Anschließend werden Duplikate entfernt und eine integrierte Resultatliste erzeugt, die an den Benutzer zurückgegeben wird.

Wrapper übermitteln eine vom Mediator übergebene Suchanfrage an den abzufragenden Suchdienst. Hierzu wird eine HTTP-Verbindung zum Suchdienst aufgebaut und unter Berücksichtigung der Zugriffsmethode (GET/POST) und der Abfragesprache

[15] Highway61: http://www.highway61.com/ [14. Nov. 2001]
[16] Meta-Ger: http://www.metager.de [14. Nov. 2001]

des Suchdienstes ein entsprechender HTTP-Request zusammengesetzt und versendet. Anschließend wird das Resultatdokument entgegengenommen, und es erfolgt eine – meist über reguläre Ausdrücke gesteuerte Extraktion – der einzelnen Resultate. Die hierdurch erzeugte Resultatliste wird anschließend dem Mediator übergeben.

Neben der Einbindung von generellen Suchdiensten wie AltaVista oder Google zur Erweiterung des Suchraums sind Metasuchsysteme insbesondere dazu geeignet, um Ressourcen des Unsichtbaren Webs zu erschließen, z. B. durch die Integration von lokalen Suchmaschinen mit spezialisierten Inhalten. Dieser Aspekt ist von zentraler Bedeutung für die vorliegende Arbeit, denn Metasuchsysteme bilden eine Schlüsseltechnologie für Verteiltes Information Retrieval. In Kapitel 3 finden sich weiterführende Betrachtungen zu dieser Thematik.

2.4. Web-Mining

In den letzten Jahren gewinnen Strategien zur Informationsgewinnung und Bewertung immer stärker an Bedeutung, die neben rein inhaltlichen Kriterien auch Informationen über die Struktur von WWW-Dokumenten mitberücksichtigen. So erlaubt das WWW die Anwendung von speziellen Techniken, welche die besonderen Gegebenheiten des WWW, wie existierende HTML-Auszeichnung von Web-Seiten, deren Hyperlinkstruktur sowie meßbares Benutzerverhalten (z. B. durch die Auswertung der Protokolldateien von Web-Servern) berücksichtigen.

Derartige Techniken werden oft unter dem allgemeinen Begriff Web-Mining zusammengefaßt [32]. Hierunter versteht man die Übertragung von bewährten Data-Mining Techniken auf das WWW, insbesondere durch das Auswerten von Hypertext (siehe hierzu auch [27]). Data-Mining bezeichnet den Prozess der automatischen Entdeckung von interpretierbaren Zusammenhängen in großen Datenmengen [42]. Dabei kommen Algorithmen zur Anwendung, die inhärent enthaltene Muster identifizieren.

Im folgenden Abschnitt wird die Bedeutung von Hyperlinks im Hinblick auf Web-Mining untersucht und die wichtigsten Techniken vorgestellt. In Abschnitt 2.4.2 und Abschnitt 2.4.3 wird auf zwei wichtige Mining-Strategien eingegangen, die eine Partitionierung von Daten in Gruppen von möglichst ähnlichen Objekten erlauben: *Clustering* und *Klassifikation*. Dabei sind insbesondere solche Verfahren von Interesse, die explizit für die automatisierte Gruppierung von Web-Ressourcen entwickelt wurden. Es wird sich erweisen, daß die Analyse der Linkstruktur sich auch hier als Kriterium zur Ähnlichkeitsbestimmung eignet.

2.4.1. Analyse von Hyperlinks

Das Setzen eines Hyperlinks zwischen zwei Web-Dokumenten durch einen Web-Autor ist das Ergebnis eines intellektuellen Prozesses, und somit bildet die Hyperlinkstruktur – systematisch ausgewertet – eine reichhaltige Informationsquelle dar, die in vielerlei Hinsicht genutzt werden kann.

Allgemein liegt hier die folgende Annahme zugrunde: Wenn eine Seite A und eine Seite B durch einen Hyperlink verbunden sind, dann ist die Wahrscheinlichkeit dafür, daß sich beide Seiten mit derselben Thematik beschäftigen, größer, als wenn diese nicht miteinander verbunden wären. Eine empirische Studie zur Belegung dieser Annahme findet sich in [36].

Die systematische Auswertung von Referenzen ist nicht neu; Forschungen aus dem Bereich der Bibliometrie reichen bis zu 30 Jahren zurück – also noch lange vor der Entstehung des WWW. Bereits in [102] schlug Salton vor, Terme, die mit einer Zitierung assoziiert sind, zur Unterstützung des Retrievalprozesses heranzuziehen. Neu hingegen ist, daß mit dem WWW ein dezentral wachsendes heterogenes Hypermediasystem enstanden ist, das Milliarden Referenzen in Form von Hyperlinks beinhaltet und damit Möglichkeiten für eine Vielzahl von Anwendungen eröffnet. Allein das Verfahren von Kleinberg [69], das später noch eingehender vorgestellt wird, zeigt auf, daß die Analyse der Hyperlinkstruktur dazu genutzt werden kann, um

- Web-Seiten qualitativ zu gewichten,
- thematisch ähnliche Web-Seiten aufzufinden und
- Web-Seiten nach verschiedenen Bedeutungskontexten zu gruppieren.

Hyperlinks, die auf eine Web-Seite p verweisen, werden als *Backlinks* von p bezeichnet. Entsprechend werden Web-Seiten, die einen Hyperlink auf p enthalten, auch als *Backlink-Seiten* von p bezeichnet. Die meisten Suchmaschinen unterstützen das Auffinden von Backlink-Seiten und erlauben die Formulierung entsprechender Anfragen – meist unter Verwendung des Schlüsselwortes $link$. Beispielsweise liefert eine Anfrage an AltaVista der Form $link : u$ Web-Dokumente zurück, die einen Hyperlink auf den spezifizierten URL u enthalten.

Hyperlinks werden oft als „noisy“ charakterisiert (auf deutsch etwa: verrauscht, verzerrt). Hierunter versteht man, daß die exakte Analyse der Gesamtstruktur durch das Vorhandensein von Hyperlinks erschwert wird, die Seiten miteinander verknüpfen, die nur einen geringen oder gar keinen thematischen Bezug aufweisen. Die hierdurch bedingte Ungenauigkeit kann sogar zu einer Verschlechterung existierender Verfahren führen [29].

Dennoch bietet sich durch Hyperlinks die Möglichkeit einer neuen Herangehensweise an einige klassische Aufgaben des Information Retrieval, wie z. B. Clustering. Die durch Hyperlinks vorgegebene Nähe zweier Dokumente zueinander kann als Grundlage für ein Ähnlichkeitsmaß fungieren, nach dem die Verschmelzung von Dokumenten zu Clustern erfolgen kann.

Neben der reinen Strukturinformation sind Hyperlinks im WWW meist mit einem gewissen Anteil an Volltext assoziiert. Dieser bietet oft eine treffendere und kompaktere Charakterisierung der verwiesenen WWW-Seite als diese Seite selber und kann somit für den Prozeß der automatischen Indexierung und Kategorisierung herangezogen werden.

Bedingt durch das explosionsartige und evolutionäre Wachstum des WWW und dem hiermit einhergehenden Überangebot an potentiell relevanten Informationen wächst das Bedürfnis nach Suchstrategien, die stärker als bisher die Qualität von Web-Seiten mitberücksichtigen. Auch hierfür eignet sich die Analyse der Hyperlinkstruktur, denn setzt ein Web-Autor einen Hyperlink auf eine bestimmte Web-Seite, so geschieht dies auch als Ausdruck einer qualitativen Einschätzung der dargestellten Inhalte.

2.4.1.1. Authorities und Hubs

Wie kann mittels Hyperlinks die Qualität einer Seite im WWW bestimmt werden? Ein erster Ansatz bestünde im einfachen Messen der eingehenden Hyperlinks. Jedoch berücksichtigt dieser Ansatz nicht, aus welchem Grund ein Hyperlink letztendlich gesetzt wurde, ob als reine Navigationshilfe oder um tatsächlich einen thematischen Querbezug zu schaffen. Unberücksichtigt bleibt hierdurch außerdem die Reputation der Quelle, von der aus verwiesen wurde. Intuitiv ist klar, daß ein Hyperlink von einer Yahoo-Seite bedeutsamer zu gewichten ist, als einer von Seiten aus der breiten Masse privater Homepages, deren Zugriffszahlen oft unter Hundert im Monat liegen.

In [69] wurden im Zusammenhang mit der Analyse der Hyperlinkstruktur des WWW erstmals zwei zentrale Begriffe eingeführt: *Hubs* und *Authorities*. Unter einer Authority versteht man eine Web-Seite, auf die sehr häufig durch Hyperlinks verwiesen wird und die eine hohe Autorität in Bezug auf ein bestimmtes Themengebiet aufweist. Hubs sind dadurch charakterisiert, daß sie auf Authorities verweisen, die Informationen zu einem bestimmten Themengebiet enthalten.

Eine bedeutsame Eigenschaft von Hub- und Authority-Webseiten ist, daß diese sich gegenseitig verstärken: Je mehr Hyperlinks von Hub-Seiten auf eine Authority verweisen, desto höher kann deren Qualität eingeschätzt werden. Hubs und Authorities, die sich gegenseitig verstärken, werden als *Communities* bezeichnet. Ein guter Hub zeichnet sich dadurch aus, daß auf ihm Links hoher Qualität, also Links zu gu-

ten Authorities, zusammengestellt sind. Aus dieser Eigenschaft läßt sich ein iteratives Verfahren ableiten, mittels dessen für eine Web-Seite p ein Hub-Gewicht $H(p)$ und ein Authority-Gewicht $A(p)$ berechnet werden kann. Dieses sogenannte *HITS-(Hyperlink-Induced Topic Search)* Verfahren [69] wird detailliert in Abschnitt 5.1.1 diskutiert.

In Abbildung 2.1 ist die Beziehung zwischen Authorities und Hubs vereinfacht dargestellt.

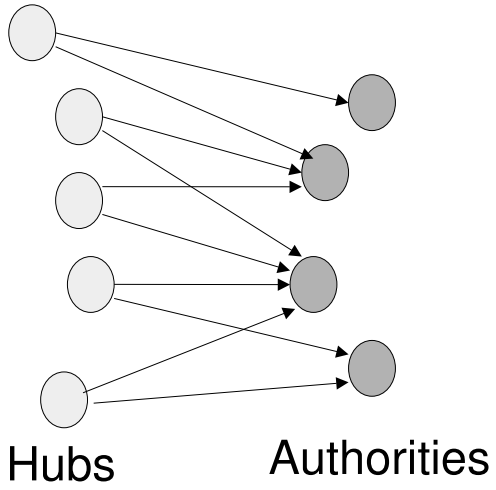

Abbildung 2.1.: Authorities und Hubs

Das HITS Verfahren läßt sich auch auf die Bibliometrie anwenden, in der Untersuchungen zur Zitierungsstruktur von Dokumenten vorgenommen werden. Behandelt man Zitate in wissenschaftlichen Arbeiten wie Hyperlinks, so lassen sich Hubs und Authorities auch für diese Dokumente berechnen. Beispielsweise wird im System ResearchIndex [17] von Giles [50] aus den Bibliographien elektronisch verfügbarer wissenschaftlicher Veröffentlichungen – meist aus dem Bereich der Informatik – automatisiert ein Zitierungsgraph aufgebaut und zu verschiedenen Themengebieten Hubs und Authorities errechnet. Bei der Durchsicht der angebotenen Authority- und Hub-Veröffentlichungen aus verschiedenen Teilbereichen der Informatik wie *Information Retrieval* oder *World Wide Web* zeigt sich, daß die besten Authorities – also die prominentesten Veröffentlichungen – zuverlässig bestimmt werden. Als gute Hub-Dokumente jedoch würde man ausführliche Übersichten oder didaktisch gut aufbereitete Tutorials erwarten. Tatsächlich werden in ResearchIndex jedoch meist nur solche Arbeiten als beste Hubs ausgezeichnet, die über umfangreiche Bibliographien

[17] ResearchIndex: http://www.citeseer.com [14. Nov. 2001]

verfügen, was allein natürlich noch nichts über die Qualität der zitierenden Arbeit aussagt.

Weitere Verfahren zur Bestimmung von Hubs und Authorities finden sich in [1], [16], [79] und [88].

2.4.1.2. Die Pagerank-Gewichtung

Ein anderes bekanntes Verfahren, das die Hyperlinkstruktur des Webs ausnutzt, um die Präzision von Suchergebnissen zu erhöhen, ist die *Pagerank-Gewichtung* [22], [90], ein Verfahren, das für den Rankingmechanismus der Suchmaschine Google angewendet wird.

Für jede Seite p wird dabei ein Pagerank-Gewicht $PR(p)$ berechnet. Dieses ist umso höher, je mehr Seiten auf p verweisen, also je mehr Backlinks auf p existieren. Wie im HITS-Algorithmus, wird im Pagerank-Gewicht die Bedeutung und Qualität der einzelnen Backlink-Seiten mitberücksichtigt.

Pagerank basiert auf der Modellierung des Benutzerverhaltens durch *Random Walks*. Dabei wird davon ausgegangen, daß ein Benutzer zwei Möglichkeiten hat, um das WWW zu durchstreifen: Entweder er folgt beginnend von einer zufälligen Seite beliebig ausgehenden Links, oder er startet erneut auf einer zufällig ausgewählten Seite. Die Wahrscheinlichkeit, daß der Benutzer eine beliebige Seite p auf diese Weise auffindet, wird als Pagerank von p bezeichnet ($PR(p)$).

Dieses Benutzerverhalten wird durch die folgende Formel ausgedrückt:

$$PR(p_0) = (1 - q) + q \sum_{i=1}^{n} \frac{PR(p_i)}{outlink(p_i)}$$

Die Seite p_0 besitzt n eingehende Links, und zwar von den Backlink-Seiten p_1 bis p_n. Die Anzahl der von einer Seite p_i ausgehenden Links wird durch $outlink(p_i)$ erfaßt. Jede Seite gibt somit in Abhängigkeit von der Anzahl der von ihr ausgehenden Links einen entsprechenden Anteil ihres eigenen Pagerank-Gewichtes an jede ihrer Folgeseiten weiter. Besitzt eine Seite z. B. nur einen ausgehenden Link, wird das Pagerank-Gewicht vollständig an die Folgeseite weitergegeben. Das vollständige Pagerank-Gewicht ergibt sich somit aus der Summe der weitergegebenen Anteile der Pagerank-Gewichte der Backlink-Seiten.

Durch den Faktor q wird die Wahrscheinlichkeit dafür ausgedrückt, daß der Benutzer einen neuen „Random Walk“ startet. Dies ist insbesondere deshalb von Bedeutung, weil es Web-Seiten oder Zyklen von Web-Seiten geben kann, aus denen kein Link mehr herausführt.

$PR(p)$ kann durch Verwendung eines iterativen Algorithmus bestimmt werden, wobei für q normalerweise ein Wert von $0,85$ angenommen wird (siehe hierzu [90]).

Es sei an dieser Stelle darauf hingewiesen, daß sich auch der HITS-Algorithmus (siehe Abschnitt 2.4.1.1) auf das Random Walk-Modell abbilden läßt. Betrachtungen hierzu finden sich in [16] und [94].

2.4.1.3. Textuelle Analyse von Hyperlinks

Die Analyse der reinen Hyperlinkstruktur bietet bereits eine wertvolle Quelle an Informationen, insbesondere um Web-Seiten im Hinblick auf ihre Qualität zu beurteilen. Darüberhinaus können hyperlinkbasierte Analyseverfahren verbessert werden, indem existierende textuelle Beschreibungen ausgewertet werden, die mit Hyperlinks assoziiert sind. Wird in HTML ein Hyperlink spezifiziert, so kann der Web-Autor diesem eine beliebig große Menge an Volltext zuordnen, der vom Browser zur Kenntlichmachung des Links während der Präsentation herangezogen wird. Auch image-basierte Hyperlinks verfügen meist über einen Alternativtext, der angezeigt wird, falls das Bild nicht geladen werden kann.

Dieser sogenannte Ankertext, der in der Regel nur aus einigen wenigen Worten besteht, kann von hyperlinkbasierten Verfahren in verschiedener Hinsicht genutzt werden. Die Suchmaschine Google nutzt diesen Ankertext, um eine Beschreibung der Ressource zu erzeugen, auf die verwiesen wird. Brin und Page argumentieren in diesem Zusammenhang, daß der Ankertext oft eine exaktere Beschreibung der verwiesenen Web-Seite liefert, als die Web-Seite selber [22]. Durch die Berücksichtigung des Ankertextes können auch multimediale Internet-Ressourcen indexiert werden, wie z. B. Graphiken oder Video-Dateien. Die Indexierung von Ankertexten birgt allerdings auch den Nachteil in sich, daß für bestimmte Suchanfragen unter Umständen scheinbar vollkommen irreführende Dokumente gefunden werden.

In [28] wird eine Variante des HITS-Algorithmus beschrieben, bei der zusätzlich textuelle Informationen berücksichtigt werden, die mit den Hyperlinks assoziiert sind. Hierzu wird ein Link von einer Seite p auf eine Seite q mit einem Gewicht $w(p,q)$ versehen. Dieses ist umso größer, je mehr themenbezogener Text sich auf der Seite p in der Nachbarschaft des Links befindet. Berücksichtigt wird der Text innerhalb eines 50 Byte großen Ankerfensters um den Link herum. Für jeden Begriff, der sowohl innerhalb des Ankerfensters als auch einer kurzen textuellen Beschreibung eines Themengebietes auftaucht, wird das Gewicht um eins erhöht. Die dahinterstehende Annahme ist, daß je mehr themenbezogener Text in der Umgebung eines Links auf einer Seite q auftaucht, desto höher ist die Autorität von q bzgl. des Themas einzuschätzen. Mit diesem sogenannten ARC-Verfahren (Automatic Resource Compilation) wurden

die 15 besten Hubs und Authorities zu 28 Themengebieten ermittelt. Die Auswahl der Themengebiete umfaßt relativ spezielle Themengebiete, wie *zen buddhism*, *cheese* oder *lyme disease*, aber auch sehr allgemein gehaltene wie *gardening* oder *graphic design*. In einem Test zur Evaluierung der Ergebnisse mußten verschiedene Benutzer die Qualität der Hubs und Authorities mit WWW-Ressourcen vergleichen, die Yahoo und Infoseek zu denselben Themengebieten zusammengestellt haben. Die Qualität der von ARC gesammelten Ressourcen wurde dabei ähnlich gut wie die von Yahoo und Infoseek beurteilt und wurde in bestimmten – meist sehr speziellen – Gebieten wie *shakespeare* oder *zen buddhism* sogar als überlegen eingestuft. Dabei muß bedacht werden, daß die ARC-Ressourcen automatisch zusammengestellt wurden, wohingegen die Sammlungen von Yahoo und Infoseek größtenteils manuell bestimmt wurden.

Alternativ zu ARC werden in [15] weitere Varianten von HITS unter Verwendung von Textanalysen beschrieben. Dabei werden die berechneten Hub- und Authority-Gewichte für ein Dokument d zusätzlich mit einem tf-idf basierten Ähnlichkeitsgewicht zwischen d und dem Themengebiet multipliziert. Das Themengebiet wird dabei als die Menge aller Worte in den ersten 1000 Dokumenten, die AltaVista auf die initiale Suchanfrage zurückgeliefert hat, repräsentiert.

2.4.2. Clustering

Retrieval-Systeme basieren auf der grundlegenden Annahme, daß die zu einer Anfrage relevanten Dokumente sich von den irrelevanten unterscheiden. Daraus kann gefolgert werden, daß die relevanten Dokumente sich untereinander mehr ähneln als den nicht-relevanten. Van Rijsbergen [113] formulierte seine Schlußfolgerungen aus diesen Beobachtungen in der sogenannten *Cluster Hypothese*:

Die Ähnlichkeit relevanter Dokumente untereinander und die Ähnlichkeit der irrelevanten Dokumente untereinander ist größer als die zwischen anderen (zufälligen) Teilmengen der Dokumentenkollektion.

Dies bedeutet, daß Dokumente auch gemeinsame Eigenschaften haben müssen, die sich als Basis für einen Vergleich eignen. Eine diese Eigenschaften ist das Vorkommen von Termen in Texten. Werden die gleichen Terme in ähnlicher Häufigkeit in verschiedenen Dokumenten benutzt, kann davon ausgegangen werden, daß die Dokumente einen gemeinsamen thematischen Bezug aufweisen. Eine weitere Eigenschaft, die zur Ähnlichkeitsbestimmung herangezogen werden kann, ist die Nachbarschaft von Dokumenten in einer Hypermedia-Umgebung oder einer bibliographischen Zitierungsstruktur.

Im Bereich des Information Retrieval kann die Technik der Cluster-Analyse dazu eingesetzt werden, um thematisch einander ähnliche Dokumente einer Dokumentenkol-

lektion in Gruppen (Cluster) aufzuteilen. Diese Gruppierung kann, muß aber nicht zwangsläufig, disjunkt sein. Es gibt keine vordefinierten Cluster und auch keine festgelegten Inhalte für einzelne Cluster. Des weiteren ist nicht vorherzusagen, wieviele Cluster mit welchen Inhalten sich durch einzelne Analyseverfahren bilden werden – deren Entstehung hängt ausschließlich von den Ähnlichkeiten der einzelnen Dokumente zueinander ab.

Beim Clustern von textuellen Dokumenten können Cluster und Dokumente als Vektoren von Termen repräsentiert werden. Somit können als Maße zur Bestimmung der Ähnlichkeit zwischen Cluster und Dokument dieselben Maße verwendet werden, wie sie bereits in Abschnitt 2.1.2 vorgestellt wurden (z. B. Kosinusmaß). Ein Cluster C wird dabei durch einen sogenannten Zentroiden repräsentiert. Dieser entspricht dem Mittelwert aller Punkte im Cluster C.

Existierende Cluster-Verfahren können grob in zwei Gruppen unterteilt werden: *nicht-hierarchische* und *hierarchische Methoden* (siehe beispielsweise [41] Kapitel 3 für eine vollständige Übersicht).

Nicht-hierarchische Cluster-Verfahren sind heuristischer Natur. Hierzu müssen eine Reihe von a priori Entscheidungen getroffen werden, z. B. über Anzahl und Größe der Cluster, Kriterien für die Clusterzugehörigkeit etc. Es muß versucht werden, Näherungslösungen zu finden, um n Dokumente auf m Cluster aufzuteilen. Für diese Lösungen werden die Dokumente partitioniert und dann durch Umverteilung optimiert, solange bis ein bestimmtes Kriterium erreicht ist. Die zur Cluster-Berechnung notwendige Laufzeit $O(nm)$ ist im Vergleich zu hierarchischen Verfahren deutlich geringer, sofern $m << n$ gilt. Ein bekannter Verteter von nicht-hierarchischen Verfahren ist K-Means.

Im Gegensatz zu den nicht-hierarchischen Verfahren, die nur eine einfache Partitionierung der Datenmenge erzeugen, werden beim hierarchischen Clustering die Daten in Form einer komplexeren Datenstruktur repräsentiert – einem sogenannten *Dendogramm*. Hierunter versteht man einen Baum, der eine hierarchische Zerlegung der Datenmengen in immer kleinere Datenmengen darstellt. Die Wurzel repräsentiert dabei einen einzigen Cluster, der die gesamte Datenmenge enthält, die Blätter des Baumes dahingegen identifizieren Cluster, die je ein einzelnes Objekt der Datenmenge enthalten. Die inneren Knoten repräsentieren die Vereinigung aller Kindknoten. Dabei werden die durch die inneren Knoten identifzierten Cluster durch ihre Zentroiden repräsentiert. Die Vereinigung von zwei Knoten zu einem Knoten in der nächsthöheren Hierarchieebene ergibt sich aus der Ähnlichkeit der Objekte zueinander. Die Techniken für hierarchisches Clustering können nach agglomerativen und divisiven Verfahren unterschieden werden. Die erstere Gruppe impliziert eine Bottom-Up-Vorgehensweise, d. h. jedes Objekt wird als einzelner Cluster behandelt und die Clus-

ter werden gemäß ihrer paarweisen Ähnlichkeit zueinander miteinander verschmolzen, bis sich alle Objekte in einem Cluster befinden. Bei divisiven Verfahren hingegen wird in der Wurzel mit einem einzigen Cluster begonnen, der von oben nach unten in immer kleinere Cluster partitioniert wird. In den letzten Jahren haben insbesondere die agglomerativen Verfahren zunehmend an Bedeutung gewonnen. Bekannte Vertreter dieser als HACM (= Hierarchical Agglomerative Clustering Methods) bezeichneten Gruppe sind die Verfahren Single Link, Complete Link und Group Average Link.

2.4.2.1. Clustering-Anwendungen im WWW

Die Übertragung von traditionellen Clustering-Verfahren auf die besonderen Anforderungen und Gegebenheiten des WWW wird durch praktische Erwägungen erschwert. So hat die Forderung nach kurzen Antwortzeiten für WWW-Suchdienste oberste Priorität, wodurch sich insbesondere der Einsatz von hierarchischen Clustering-Techniken aufgrund ihrer quadratischen Laufzeit verbietet. Auf der anderen Seite ergeben sich durch die Hyperlinkstruktur des WWW neue Möglichkeiten zur Berechnung von Ähnlichkeiten zwischen Dokumenten und zur Cluster-Generierung.

Heutzutage werden Cluster-Verfahren im WWW in verschiedenster Art- und Weise zur Unterstützung des Prozesses der Aufbereitung und Gewinnung von Informationen eingesetzt:

- Zur Strukturierung großer Antwortmengen: Die meisten Suchmaschinen liefern sehr große Antwortmengen mit Tausenden von Dokumenten zurück. Clustering-Techniken bieten sich hier an, um die verschiedenen Treffer nach ihren unterschiedlichen Bedeutungskontexten zu gruppieren und somit eine übersichtlichere Ergebnispräsentation zu gewährleisten. Dies kann auch zur Erhöhung der Präzision beitragen, weil unter Umständen irrelevante Dokumente in eigenen Clustern gruppiert werden und somit leichter aussortiert werden können. Beispiele hierfür finden sich in [30], [100] und [127].

- Bestimmung von gemeinsamen Informationen, um Mengen von Dokumenten zu charakterisieren: Die Kriterien, nach denen die Zusammenfassung von Dokumenten erfolgt, können für sich eine wertvolle Informationsquelle darstellen. Dies können beispielsweise Terme sein, die besonders häufig ko-zitiert werden. Diese erlauben als Deskriptoren eine textuelle Charakterisierung von Clustern und können vom Benutzer für eine Anfrageverfeinerung verwendet werden.

- Ermittlung von Communities: Werden beim Clustering hyperlinkbasierte Nachbarschaftsbeziehungen von Web-Dokumenten mit berücksichtigt, lassen sich

Communities im WWW identifizieren, die sich mit einer gemeinsamen Thematik beschäftigen.

- Ermittlung des Benutzerverhaltens: Clustering-Verfahren können auch auf die Log-Dateien der Web-Server angewendet werden. Hierdurch können verschiedene Benutzergruppen identifiziert werden, und es lassen sich Rückschlüsse über das Verhalten von Benutzergruppen bei der Navigation auf einer Web-Site ziehen.

Mittels des Verfahrens Suffix Tree Clustering (STC) [126] lassen sich große Ergebnismengen effektiv gruppieren. STC garantiert kurze Antwortzeiten, denn die dem Verfahren zugrundeliegende Datenstruktur des Suffix-Baums kann in linearer Laufzeit $O(n)$ gemäß der Anzahl n der zu gruppierenden Dokumente konstruiert werden. Der Suffix-Baum wird dabei inkrementell mit dem Einlesen der Dokumente aufgebaut. Eine weitere Reduzierung der Antwortzeiten wird dadurch erreicht, daß das Gruppieren nicht auf der Basis der vollständigen Dokumente durchgeführt wird, sondern lediglich mittels der von den Suchmaschinen zur Verfügung gestellten Dokument-Snippets. Das Verfahren gewährleistet somit eine strukturierte Ergebnispräsentation bei linearer Laufzeit und erscheint somit insbesondere für den Einsatz im Metasuchmaschinenbereich geeignet.

Erste Experimente zur Gruppierung von Web-Dokumenten durch Analyse der Hyperlinkstruktur finden sich beispielsweise in [75]. In [86] und [117] werden Clustering-Verfahren beschrieben, die zur Berechnung von Dokumentähnlichkeiten neben den in den Dokumenten enthaltenen textuellen Information auch existierende Linkstrukturen durch Berücksichtigung von Ko-Zitierungen miteinbeziehen. Dabei können bei der Ähnlichkeitsbestimmung der relative Einfluß von Termen, von eingehenden Links und von ausgehenden Links individuell gewichtet werden. In [86] wird hierzu eine Variante des K-Means Clustering verwendet, wohingegen die in [117] beschriebene Suchmaschine HyPursuit das Clustering nutzt, um den Suchraum hierarchisch zu strukturieren. Dementsprechend kommt hier ein hierarchisches Clustering-Verfahren (Complete Link) zum Einsatz.

Durch die beschriebenen Verfahren zum hyperlinkbasierten Clustering werden allerdings nur in begrenztem Umfang die Hub- und Authority-Eigenschaften der zugrundeliegenden Link-Struktur ausgenutzt. So kann das in Abschnitt 2.4.1.1 bereits erwähnte HITS-Verfahren ebenfalls dazu eingesetzt werden, um WWW-Dokumente gemäß ihrer Link-Struktur zu gruppieren. Dies kann durch die Berechnung der Eigenwerte und Eigenvektoren der zur Linkstruktur gehörigen Adjazenzmatrizen erreicht werden. Eine Vorstellung dieses Verfahrens, sowie eine Anwendung hiervon für Ressourcen des Unsichtbaren Webs finden sich in Abschnitt 5.1 dieser Arbeit.

2.4.3. Automatische Kategorisierung

Unter *automatischer Kategorisierung* (oder *Klassifikation*) versteht man die Zuordnung von Themengebieten oder Kategorien (oder Klassen) zu unstrukturiert vorliegenden Dokumenten. Im Gegensatz zum Clustering sind bei der automatischen Kategorisierung die einzelnen Kategorien bereits vorgegeben.

Formal läßt sich das Problem folgendermaßen ausdrücken: Gegeben ist eine Menge vordefinierter Kategorien $C = \{c_1, \ldots, c_n\}$ und eine Menge von Dokumenten $D = \{d_1, \ldots, d_m\}$. Die Kategorisierung besteht nun darin, die unbekannte totale Funktion $f : D \times C \to \{0, 1\}$ anzunähern, wobei gilt:

$$f(d_i, c_j) = \begin{cases} 1 & \text{falls gilt: das Dokument } d_i \text{ gehört der Kategorie } c_j \text{ an} \\ 0 & \text{falls gilt: das Dokument } d_i \text{ gehört der Kategorie } c_j \text{ nicht an} \end{cases}$$

Die für die Kategorisierung der Dokumente notwendigen Eigenschaften werden aus dem Inhalt der Dokumenttexte extrahiert. Dabei wird eine Kollektion von Dokumenten in eine Lern- und eine Teststichprobe aufgeteilt. Die Dokumente der Lernstichprobe sind in der Regel bereits kategorisiert, d. h. ihnen wurde bereits im Vorfeld eine oder mehrere Kategorien aus der Kategorienmenge C zugeordnet. Die Lernstichprobe wird dazu verwendet, um unter dem Einsatz von maschinellen Lerntechniken Repräsentanten für die einzelnen Kategorien zu erzeugen, z. B. in Form eines einzelnen Term-Vektors pro Kategorie.

Anschließend werden die Dokumente der Teststichprobe mit den Repräsentanten der Kategorien verglichen und hieraus eine Zuordnung der Dokumente zu den Kategorien abgeleitet. Dabei werden die Termgewicht-Vektoren der Dokumente mit den Termgewicht-Vektoren der Repräsentanten verglichen. Dies kann z. B. unter Verwendung von Ähnlichkeitsmaßen erfolgen, die auch im Clustering bzw. im Vektorraummodell Anwendung finden. Alternativ dazu können neu zu klassifizierende Dokumente auch direkt mit Dokumenten verglichen werden, die bereits zuvor kategorisiert wurden. Den Dokumenten wird dann jene Kategorie zugewiesen, der das jeweils ähnlichste Dokument bereits angehört. Diese Vorgehensweise erspart das explizite Erzeugen von Repräsentanten für die einzelnen Kategorien.

Häufig verwendete Techniken sind Support Vector Machines, Naive Bayes und k-nearest neighbour classification. Siehe [106], [123] und [124] für eine Übersicht.

2.4.3.1. Automatisierte Klassifikation von Web-Ressourcen

Wie in Abschnitt 2.3.1 dargestellt, stößt die manuelle Kategorisierung von Internetressourcen bei einem Aufkommen von täglich bis zu 4000 neuer WWW-Sites an ihre Grenzen. Deshalb besteht ein großes Interesse daran, den Prozeß der Kategorisierung von Web-Dokumenten weitestgehend zu automatisieren.

Automatische Kategorisierungsverfahren werden bislang bei der Katalogpflege noch relativ wenig eingesetzt. Zwar ermöglichen existierende Verfahren eine relativ zuverlässige Kategorisierung von Web-Seiten, dennoch reicht deren Qualität nicht an eine intellektuell vorgenommene Zuordnung heran. Nichtsdestotrotz bieten einige kommerzielle Anbieter heute bereits Kataloge an, die automatisiert erzeugt wurden. Hierzu gehört beispielsweise der Thunderstone Web Site Catalog[18].

Die von der Suchmaschine Northernlight eingesetzte Technik der *Custom Search Folders* ermöglicht zum Anfragezeitpunkt die dynamische Einordnung von Suchergebnissen in eine Hierarchie von themenspezifischen Ordnern. Hierzu werden allen Dokumenten zum Zeitpunkt der Indexierung zusätzlich Kategorien gemäß einer Themenhierarchie von 20.000 Begriffen zugeordnet.

Experimentelle Systeme zur Kategorisierung von Web-Dokumenten finden sich z. B. in [68], [74], [85] und [91]. Dabei fällt auf, daß viele derartige Verfahren immer wieder auf das Kategorisierungsschema von Yahoo zurückgreifen [68], [74] und [85]. Die Themenhierarchie von Yahoo stellt eine weltweit verbreitete und akzeptierte Klassifikation dar, und ist daher dazu geeignet, elektronische Dokumente einheitlich zu kategorisieren. Außerdem können verschiedene automatische Kategorisierungsverfahren für Web-Seiten miteinander verglichen werden, indem verifiziert wird, in welchem Maße die im Yahoo-Katalog erfaßten Seiten den richtigen Kategorien zugeordnet werden konnten.

Im Verfahren von [68] beispielsweise wird der komplette Unterbaum *Computer & Internet* des Yahoo-Katalogs mit insgesamt 2806 Kategorien und 18639 Dokumenten extrahiert und kategorisiert. Die Lernstichprobe besteht aus 12315 Dokumenten und die Teststichprobe aus 6324 Dokumenten. Zur Erzeugung eines Kategorie-Repräsentanten werden alle Dokumente einer Kategorie konkateniert und mittels der tf-idf Methode gewichtet.

Allerdings können die verschiedenen Verfahren zur Kategorisierung von Web-Ressourcen im Hinblick auf deren Qualität und Performanz nur schlecht miteinander verglichen werden. Zwar werden die Verfahren meist zu der manuellen Kategorisierung des Yahoo-Katalogs in Bezug gesetzt, jedoch werden in den einzelnen Experi-

[18] Thunderstone: http://search.thunderstone.com/texis/websearch/about.html [14. Nov. 2001]

mentbeschreibungen zum Teil nur relativ ungenaue Aussagen darüber getätigt, welcher Teilbaum des Yahoo-Katalogs zur Betrachtung herangezogen wurde bzw. zu welchem Zeitpunkt die Experimente durchgeführt wurden, was einen Vergleich der Verfahren untereinander erschwert. In den Experimenten wird für jedes Verfahren eine unterschiedliche Anzahl an Kategorien und Dokumenten berücksichtigt. Hinzu kommt die Auswahl der Kategorien im Hinblick auf deren thematische Diskrepanz: Werden beispielsweise nur thematisch weit auseinanderliegende Kategorien berücksichtigt, wie z. B. nur die Yahoo-Kategorien der obersten Hierarchiestufe, so ist eine Kategorisierung einfacher vorzunehmen, da zu erwarten ist, daß die Themengebiete und damit die enthaltenen Dokumente relativ disjunkt zueinander sind. Werden dagegen nur Kategorien aus einem relativ beschränkten Teilbereich, z. B. nur Unterkategorien aus dem Bereich Mikrobiologie berücksichtigt, fällt die Zuordnung ungleich schwerer.

2.4.3.2. Kategorisierung nach Kontext

Mit dem Aufkommen des WWW als verteiltem Hypermediasystem haben sich in jüngster Zeit verstärkt Ansätze etabliert, die eine Kategorisierung aus dem Kontext eines Dokumentes heraus vornehmen.

Dabei werden zusätzlich beschreibende Informationen über das zu kategorisierende Dokument aus benachbarten Dokumenten – z. B. in einer Hypertextumgebung – extrahiert. Hierzu kann jener Volltext herangezogen werden, der der Umgebung von Hyperlinks unterliegt, die auf das zu kategorisierende Dokument verweisen. Die Kategorisierung durch Kontext hat den Vorteil, daß auch Dokumente kategorisiert werden können, die selber keine textuellen Daten enthalten, wie Programme oder Multimediadaten. Allerdings kann eine Kategorisierung nur dann zuverlässig vorgenommen werden, wenn auf ein Dokument bereits eine ausreichende Zahl Links verweist.

In [6] wird ein solches Kategorisierungsverfahren beschrieben. Für jedes zu kategorisierende Dokument d_i wird eine kompakte Beschreibung $blurb(d_i)$ (blurb = Klappentext eines Buches) gebildet. Diese Beschreibung $blurb(d_i)$ wird aus Textfenstern um die Hyperlinks jener Seiten gebildet, die auf d_i verweisen. Die eigentliche Kategorisierung erfolgt unter Verwendung traditioneller Indexierungsmechanismen und Lerntechniken, mit dem Unterschied, daß $blurb(d_i)$ und nicht der Volltext von d_i berücksichtigt wird.

Verfahren, in denen eine Kategorisierung von Web-Ressourcen auf der Basis von Hypertext vorgenommen wird, finden sich auch in [29] und [45].

2.5. Metadaten

Ein wichtiges Konzept, um die Informationsvielfalt und die exponentiell anwachsende Datenmenge des WWW effizient erschließen zu können, stellt die Verwendung von Metadaten dar. Metadaten entstammen ursprünglich aus dem bibliothekarischen Umfeld und gewährleisten das gezielte Wiederauffinden von Informationen sowie deren intellektuelle und physikalische Nutzung. Sie bestehen aus einer Menge von Eigenschaften, die ein Objekt wie beispielsweise ein Buch beschreiben. Typische Metadaten sind *Autor, Titel, Erscheinungsdatum, Schlagworte* etc.

2.5.1. Eine Spezifikation von Metadaten

Metadaten werden allgemein oft als „Daten über Daten“ bezeichnet. Dabei gibt es keine eindeutige Unterscheidung zwischen Daten und Metadaten. Vielmehr stellen Metadaten selber wiederum eigenständige Ressourcen dar. Dieser Sachverhalt wird unter anderem durch die folgende Definition von Metadaten verdeutlicht, die sich im Wesentlichen an [51] orientiert:

Metadaten bilden die Summe der Aussagen, die über ein Informationsobjekt getroffen werden können. Dies schließt die verschiedenen Sichtweisen aller potentiellen Benutzergruppen mit ein (Laie, Spezialist, Suchender, Informationsanbieter etc.).

Ein Informationsobjekt ist in diesem Zusammenhang definiert durch all das, was durch einen Menschen oder ein System als diskrete Einheit adressiert und manipuliert werden kann. Dabei kann sich ein Informationsobjekt aus einem einzelnen Element oder einer Aggregation von Elementen zusammensetzen. Unabhängig von ihrer physikalischen oder intellektuellen Erscheinungsform besitzen Informationsobjekte drei Klassen von Merkmalen, die durch Metadaten ausgedrückt werden können:

1. *Inhaltsmerkmale* beziehen sich auf all das, was die im Informationsobjekt enthaltenen Daten inhaltlich charakterisiert. Inhaltsmerkmale sind intrinsischer Natur.

2. *Kontextmerkmale* kennzeichnen extrinsische Merkmale, die mit der Erzeugung des Objektes in Zusammenhang stehen, also Informationen über das Wer, Wo, Wann, Warum und Wie.

3. *Strukturmerkmale* beschreiben strukturelle Beziehungen die innerhalb einzelner Informationsobjekte oder zwischen veschiedenen Informationsobjekten bestehen können. Sie können sowohl extrinsisch als auch intrinsisch sein.

Metadaten können sowohl manuell (z. B. durch den Autor eines Dokumentes oder

einen Spezialisten) als auch computergestützt (z. B. durch automatische Indexierer) generiert werden. Die Erstellung und der Austausch von Metadaten in den Bibliotheken erfolgt unter Verwendung komplexer Formate wie MARC (MAchine-Readable Cataloging format) [49] oder das deutsche MAB[19] (Maschinelles Austauschformat für Bibliotheken). Diese definieren eine Menge von Feldern – den sogenannten Metadatensatz – die von den Bibliothekaren unter Berücksichtigung der vorgegebenen Syntax und Semantik mit konkreten Werten versehen werden. Derartig erzeugte bibliographische Metadaten können den Benutzern elektronisch über sogenannte *OPACs (Online Public Access Catalogs)* zugänglich gemacht werden.

2.5.2. Metadaten für das WWW

2.5.2.1. Allgemeine Anforderungen

Es zeigt sich, daß die Erstellung von Metadaten mit einem hohen Kostenaufwand verbunden ist. Berücksichtigt man ein geschätztes Wachstum von etwa 4000 neuer Web-Sites pro Tag[20], erscheint die systematische Auszeichnung von Web-Seiten mit Metadaten kaum praktikabel. Selbst Web-Kataloge wie Yahoo können nur einen Bruchteil des Webs erfassen – oft bleiben den menschlichen Indexierern nur wenige Sekunden, um eine neue Seite zu klassifizieren.

Deshalb zielen Metadaten-Techniken und Strategien für das WWW darauf ab, den Aufwand zur Auszeichnung von Web-Ressourcen mit Metadaten weg von den Suchmaschinenbetreibern hin zu den eigentlichen Informationserzeugern, den Autoren der Web-Seiten, zu verlagern. Dies erfordert das Vorhandensein eines allgemein akzeptierten Metadatensatzes für Web-Ressourcen. Im Einzelnen muß dieser den folgenden Kriterien genügen[115]:

- Er muß überschaubar und auch für Laien leicht verständlich sein.
- Die Metadaten müssen leicht erzeugt und gepflegt werden können.
- Die einzelnen Felder müssen einer allgemein verbreiteten und akzeptierten Semantik gehorchen.
- Es muß Konformität zu existierenden und zukünftigen Standards gewährleistet sein.
- Er muß auf internationaler Ebene anwendbar sein.

[19] MAB: http://www.ddb.de/cgi-bin/bermudix.pl?url=professionell/mab.htm [14. Nov. 2001]

[20] Bei einer durchschnittlichen Größe von 250 Web-Seiten pro Web-Site entspricht dies einem Wachstum von täglich um bis zu 1 Mio Web-Seiten.

- Er muß erweiterbar sein.
- Er muß die Interoperabilität zwischen verschiedenen Sammlungen und Suchsystemen gewährleisten.

2.5.2.2. Der Dublin Core

Diese Ziele führten zur Entwicklung des *Dublin Core (DC)* [115], einem Metadatensatz, der speziell zur Auszeichnung elektronischer Ressourcen geeignet ist. Der DC besteht aus 15 Elementen - 7 für Inhaltsmerkmale und 8 für Kontextmerkmale. Letztere können noch einmal unterteilt werden in je vier Merkmale zur Beschreibung von Eigentumsrechten und Merkmalen zur abstrakten Beschreibung der digitalen Objekte:

- Inhaltsmerkmale: *Title*, *Subject*, *Description*, *Source*, *Language*, *Relation*, *Coverage*
- Eigentumsrechte: *Creator*, *Publisher*, *Contributor*, *Rights*
- Abstrakte Beschreibung: *Date*, *Type*, *Format*, *Identifier*

Zur Einbindung von Metadaten in Web-Dokumente sieht der HTML-Standard das META-Element vor. Hierüber können Metadaten im Kopf eines HTML-Dokumentes spezifiziert werden [71]. Der HTML-Standard sieht für das META-Element die Attribute *name* und *content* vor, um den Namen eines Feldes und einen konkreten Wert zu spezifizieren. Zusätzlich besteht die Möglichkeit, über das *scheme*-Attribut existierende Kodierungsschemata mitanzugeben und auf diese Weise die korrekte Interpretation der Daten sicherzustellen.

Beispielsweise spezifiziert die HTML-Anweisung

```
<META name=''DC.Date'' scheme=''iso8601'' content=''1998-05-14'' >
```

eine Datumsangabe gemäß dem DC-Feld *Date*[21]. Um die syntaktisch korrekte Interpretation der Datumsangabe (JJJJ-MM-DD) zu gewährleisten, wird das entsprechende internationale Schema angegeben, nach dem die Kodierung der Datumsangabe erfolgt. Darüberhinaus können durch die Spezifikation eines kontrollierten Vokabulars auch semantische Interpretationshilfen mitangegeben werden, z. B. allgemeine oder fachspezifische Klassifikationsschemata wie der *Dewey Decimal Classification (DDC)* oder der *Medical Subject Headings (MESH)*[22] – einem Klassifikationsschema für medizi-

[21] *Date* ist spezifiziert durch das Datum, an dem die Ressource in der gegenwärtigen Form zugänglich gemacht wurde.
[22] MESH: http://www.nlm.nih.gov/mesh/meshhome.html [14. Nov. 2001]

nische Terminologie. Das folgende Beispiel zeigt die Verwendung von MESH für das DC-Feld Subject[23]:

```
<META name=''DC.Subject''scheme=''MESH'' content=''Myocardial Infarc-
tion; Pericardial Effusion'' >
```

Eingebundene Kodierungsschemata werden im DC- Sprachgebrauch[8] auch als *Qualifizierer* (engl.: Qualifier) bezeichnet (gekennzeichnet mit *dcq*, z. B. *dcq:iso8601* für DC-Feld *Date*). Daneben sind weitere Qualifizierer definiert, über die eine semantische Verfeinerung der 15 Elemente vorgenommen werden kann[24]. Beispielsweise stellt der Qualifizierer *dcq:revised* eine Verfeinerung des *Date*-Feldes dar und entspricht einem Revisionsdatum. Die Verarbeitung der DC-Qualifizierer erfolgt nach dem sogenannten *Dumb-Down-Prinzip*. Dieses besagt, daß es für einen Informations-Nutzer gemäß seinen individuellen Anforderungen möglich sein muß, Qualifizierer zu ignorieren und die verbleibende Information so nutzen zu können, als wäre sie ohne die Qualifizierer ausgezeichnet worden. Qualifizierer ermöglichen somit keine Erweiterung des semantischen Gültigkeitsbereiches, sondern ausschließlich eine Verfeinerung der bereits vorhandenen Informationen.

2.5.3. Probleme bei der Benutzung von Metadaten im WWW

Der Dublin Core erlaubt eine rudimentäre Charakterisierung von WWW-Ressourcen. Dabei ermöglichen die in ein HTML-Dokument eingebundenen DC-Metadaten den Betreibern von Suchmaschinen über die reine Volltext-Analyse hinaus eine kontextsensitive Indexierung der Dokumente, was eine Verbesserung der Retrievalqualität zur Folge hat. Dennoch stellt die regelkonforme Auszeichnung von Web-Ressourcen mit Metadaten durch die Web-Autoren jedoch eher die Ausnahme dar. In der Giles-Studie von 1999 wird der Anteil der mit dem Dublin Core ausgezeichneten Web-Sites mit nur 0,3% angegeben. Eine einfache Auszeichnung mit Schlüsselworten und Kurzbeschreibungen unter Verwendung des HTML-META-Tags[25] findet sich für 34% aller Web-Sites.

In [84] wird ein Wert für die kritische Masse von Web-Seiten angegeben, die mindestens mit Metadaten ausgezeichnet sein müssen, damit der Prozeß der Informationsgewinnung im WWW noch sinnvoll unterstützt werden kann. Dabei wird davon ausgegangen, daß die Metadaten einer bestimmten Seite bis zu einem gewissen Grad auch für jene Seiten Gültigkeit besitzen, die über einen Hyperlink-Pfad mit dieser verbunden sind – je weiter man sich von der Metadaten-Seite entfernt, desto stärker

[23] *Subject* ist spezifiziert durch das Thema der Ressource bzw. Schlagwörter oder Phrasen, die das Thema oder den Inhalt des Dokumentes beschreiben.
[24] Bislang wurden insgesamt 52 Verfeinerungs-Qualifizierer definiert.
[25] Der HTML-Standard definiert hierzu keinen spezifischen Metadatensatz als Vorgabe.

schwächt sich Bedeutung der Metadaten für die aktuelle Seite ab. Die Studie kommt dabei zu folgendem Ergebnis: Um noch für 50% aller Web-Seiten eine sinnvolle Zuordnung von Metadaten zu gewährleisten, müssen mindestens 16% aller Web-Seiten mit Metadaten ausgezeichnet werden.

Gründe für die ungenügende Verwendung von Metadaten im WWW sind, daß das Hinzufügen von Metadaten zu Web-Ressourcen nicht zwingend vorgeschrieben ist und einen arbeitstechnischen Mehraufwand bedeutet, dessen Notwendigkeit den meisten Autoren von Web-Seiten nur unzureichend bewußt ist.

Erschwerend kommt hinzu, daß das META-Element häufig mißbräuchlich verwendet wird, um für bestimmte Web-Seiten ein hohes Ranking durch generelle Suchmaschinen zu erreichen. Außerdem wird die Einbettung von standardisierten Metadaten durch die gängigen Web-Editoren und Web-Publishing-Programme technologisch noch nicht ausreichend unterstützt.

Die manuelle Erzeugung von Metadaten kann unterstützt werden durch Programme, wie dem MyMetaMaker for Thesis (MMMfT), der im Rahmen des DFG-Projektes *Dissertation Online* [37] entwickelt wurde. Dieser erlaubt die formularbasierte Eingabe von Metadaten (z. B. über eine WWW-Schnittstelle) und erzeugt automatisiert entsprechende HTML-Fragmente mit META-Elementen gemäß dem Dublin Core Standard, die dann wiederum in den Kopf von Web-Dokumenten eingefügt werden können. Der Einsatz des MMMfT ist jedoch auf ein sehr eingeschränktes bibliothekarisches und universitäres Umfeld begrenzt. Dabei wird der MMMfT primär genutzt, um Metadaten für die elektronischen Versionen wissenschaftlicher Dokumente, wie Dissertationen oder Diplomarbeiten zu erzeugen bzw. von den Autoren erzeugen zu lassen.

2.5.4. Wissensmanagement im WWW

2.5.4.1. Resource Description Framework (RDF)

Eine wichtige Voraussetzungen für die automatisierte Erschließung von Metadaten im WWW ist die Verfügbarkeit einer maschinenlesbaren Semantik, in der Metadaten einheitlich repräsentiert und ausgetauscht werden können. Die Daten müssen leicht verarbeitet werden können, auch wenn sie in großen Mengen vorliegen und jede unabhängige Organisation oder Person soll neue Metadaten definieren und spezifizieren können.

Dies erlaubt es, Anwendungen zu programmieren, die im WWW verteilt hinterlegte Metadaten zentral aufbereiten. Somit wird ein präziseres Auffinden von Informationen ermöglicht, als dies durch eine reine Volltextsuche gewährleistet werden kann.

Die genannten Forderungen führten zur Definition von RDF (Resource Description

Framework) [76]. RDF ist eine Empfehlung des W3C und stellt ein Rahmenwerk dar zur Beschreibung und zum Austausch von Metadaten im WWW. RDF basiert auf XML (eXtended Markup Language) [19] – einer Metasprache mit der Fähigkeit zur Definition neuer Auszeichnungselemente. Somit können RDF-Daten und Anwendungen in eine XML-Infrastruktur eingebunden werden, wodurch die Interoperabilität erleichtert wird.

Die Basis von RDF bildet ein Datenmodell zur semantischen Auszeichnung von Daten. Dieses Modell setzt sich aus drei Bestandteilen zusammen:

1. Ressourcen: Eine Ressource kann sowohl eine vollständige Web-Seite, ein Teil einer Web-Seite, als auch eine Kollektion von Web-Seiten sein. Auch ein Objekt, das nicht über das WWW verfügbar ist (z. B. ein gedrucktes Buch), stellt eine gültige Ressource dar. Ressourcen werden stets über ihren URI identifiziert und benannt.

2. Eigenschaften: Eigenschaften sind spezifische Aspekte, Relationen oder Attribute, die zur Beschreibung einer Ressource verwendet werden können.

3. Anweisungen: Eine Anweisung setzt sich zusammen aus einer spezifischen Ressource, einer benannten Eigenschaft und einem Wert, der die Eigenschaft für die Ressource aufweist.

Anweisungen bilden das grundlegende Strukturierungselement in RDF. Die drei Teile der Anweisungen (Ressource, Eigenschaft und Wert) werden als Subjekt, Prädikat und Objekt bezeichnet. Beispielsweise wird durch die folgende RDF-Anweisung festgelegt, daß die Ressource “http://www.w3.org” (Subjekt) einen Verleger besitzt (Prädikat) – nämlich das World Wide Web Consortium (Objekt). Das Beispiel zeigt auch die Verwendung von XML-Namensräumen [18], wobei das Dublin Core (Namensraum dc) Feld *Publisher* verwendet wird.

```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns:dc="http://purl.org/dc/elements/1.1/" >
   <rdf:Description about="http://www.w3.org" >
          <dc:Publisher> World Wide Web Consortium </dc:Publisher>
   </rdf:Description>
</rdf:RDF>
```

Subjekt und Objekt können beliebige Ressourcen sein, spezifiziert durch einen URI oder eine Zeichenkette. Dies erlaubt die Verkettung einzelner Anweisungen. Beispielsweise wird durch die verketteten Anweisungen

```
<rdf:RDF>
   <rdf:Description about="http://www.w3.org/Home/Lassila" >
          <Creator rdf:resource="http://www.w3.org/staffid/85740"/>
   </rdf:Description>

   <rdf:Description about="http://www.w3.org/staffid/85740" >
          <Email> lassila@w3.org </Email>
   </rdf:Description>
</rdf:RDF>
```

festgelegt, daß die Ressource mit dem URI ”http://www.w3.org/Home/Lassila” von dem Objekt ”http://www.w3.org/staffid/85740” (einem Angestellten mit der id 85740) erzeugt wurde. In der nächsten Anweisung fungiert dieses Objekt als Subjekt, und es wird festgelegt, daß dessen email-Adresse *lassila@w3.org* lautet.

Eine RDF-Anweisung bildet wiederum selbst eine Ressource. Dies ermöglicht die rekursive Einbettung von Anweisungen. Eine RDF-Anweisung kann auch als ein gerichteter, beschrifteter Graph angesehen werden: Subjekte und Objekte bilden die Knoten, Kanten sind immer von Subjekt zu Objekt gerichtet und mit dem Prädikat beschriftet.

Ressourcen mit gleichen Eigenschaften können in RDF in Containern zusammengefaßt werden. Hierzu sind verschiedene Typen von Containern definiert, wie beispielsweise ungeordnete Listen (bag) und geordnete Listen (seq).

Das RDF-Modell sieht keine Mechanismen vor, um sowohl Eigenschaften als auch Beziehungen zwischen Eigenschaften und Ressourcen zu definieren. Um diesbezüglich die Modellierungsmöglichkeiten von RDF zu erweitern, erfolgte die Spezifikation von *RDF-Schema (RDFS)* [21]. RDFS wird über den Namensraum *rdfs* identifiziert und ist eine Sammlung von RDF-Ressourcen, die dazu verwendet werden kann, um Eigenschaften für anwendungsspezifische RDF-Ressourcen zu definieren. In RDF/RDFS können Ressourcen als Instanzen einer oder mehrerer Klassen modelliert werden.

2.5.4.2. Das Semantische Web

In dem Buch „Weaving the Web“ [12] sowie einer Reihe von Artikeln [11], [13] beschreibt Tim Berners-Lee, der Erfinder des WWW, seine Vision vom Web der Zukunft, dem sogenannten „Semantischen Web“.

Das Semantische Web bildet in seiner Gesamtheit ein System, das einer globalen Datenbank ähnelt, und basiert auf einer einheitlichen und maschinenlesbaren Repräsentation von Daten und Wissen.

Die Darstellung von Wissen im Semantischen Web erfolgt nach demokratischen Ge-

sichtspunkten, d. h. es existiert keine von zentraler Stelle vorgegebene universelle Definition von Konzepten, vielmehr können Web-Autoren auf ihren Web-Seiten selbstdefinierte Konzeptspezifikationen zur Wissensrepräsentation bereitstellen. Verschiedene Konzepte können dabei über beliebige Relationen zueinander in Bezug gesetzt werden. Dabei können sich existierende Konzeptspezifikationen durchaus widersprechen. Welche Bedeutung für ein bestimmtes Konzept dann tatsächlich als allgemeingültig anerkannt wird, entwickelt sich dynamisch aus dem Grad der Akzeptanz und der übereinstimmenden Verwendung innerhalb von verschiedenen Web-Communities. Ähnlich wie sich gute Hubs und Authorities dynamisch aus der Struktur ihrer Verlinkung herauskristallisieren, ergibt sich die Bedeutung eines Konzeptes aus dem Grad seiner Referenzierung.

Bei der Realisierung des semantischen Webs kommt RDF/RDFS eine wichtige Rolle zu. So erfordert das semantische Web das Vorhandensein von Werkzeugen zur benutzerfreundlichen Spezifikation, Manipulation und Auswertung von Wissen. RDF und RDFS bilden die Grundlage für solche Werkzeuge, denn sie stellen das Vokabular bereit, um Wissen in einem maschinell auswertbaren und austauschbaren Format darzustellen.

2.5.4.3. Wissensrepräsentation durch Ontologien

Formal repräsentiertes Wissen basiert auf einer Konzeptualisierung. Diese besteht aus einer Menge von abstrakten Konzepten, über die Wissen ausgedrückt wird, sowie einer Menge von Beziehungen zwischen diesen. Die explizite Spezifikation einer solchen Konzeptualisierung wird als *Ontologie* bezeichnet.

Ontologien entstammen dem Bereich der künstlichen Intelligenz. Ihr Ziel ist es, Wissen einheitlich in konzeptualisierter Form zu repräsentieren und damit dessen Wiederverwendbarkeit zu gewährleisten (siehe z. B. [43], [55], [89]).

In bestimmten, klar abgegrenzt thematischen, Bereichen mit fest definierter Terminologie haben sich ontologiebasierte Wissensbasen bewährt (z. B. bei Expertensystemen). Demgegenüber konnten Projekte, in denen versucht wurde, eine globale Ontologie zu definieren (wie im CYC-Projekt [80]), die in sie gesetzten Erwartungen bislang nicht erfüllen [66].

Ontologien sind eine axiomatische Charakterisierung der Bedeutung eines logischen Vokabulars. In vielen Fällen drücken die Axiome einer Ontologie nur Unterordnungsbeziehungen bzw. Enthaltenseinsbeziehungen (IS-A Relation) zwischen einstelligen Prädikaten aus. Eine solche einfache Ontologie wird auch als Taxonomie bezeichnet. Oft ist es aber notwendig, eine detailliertere Axiomatisierung vorzunehmen, um ungewollte Interpretationen auszuschließen.

Mit dem Aufkommen des WWW wurden verstärkt Versuche unternommen, um ontologiebasiertes Wissen in Web-Infrastrukturen einzubetten. Prominente Ansätze hierzu sind OIL ([24], [63]), SHOE [60], DAML[26] und Ontobroker [109]. RDF/RDFS erlaubt die Definition von einfachen Relationen und Konzepten und somit die Erstellung von einfachen Ontologien. Allerdings sind keine Möglichkeiten vorhanden, um Axiome wie Reflexivität, Transitivität oder Symmetrie in RDF/RDFS auszudrücken, um komplexe Ontologien zu spezifizieren. Hieraus motiviert sich das Projekt *OIL (Ontology Inference Layer)*[24], [63]. In OIL werden entsprechende Erweiterungen des RDF/RDFS-Vokabulars bereitgestellt. Diese können unter Verwendung des Namensraummechanismus (*xmlns:oil*) zusammen mit RDF/RDFS Spezifikationen verwendet werden.

[26] DAML: http://www.daml.org/ [14. Nov. 2001]

3. Anforderungen an ein Web-basiertes Verteiltes IR-System

Dieses Kapitel beschäftigt sich mit den Eigenschaften von *Verteiltem Information Retrieval*. Zunächst erfolgt eine Diskussion der Eigenschaften des Unsichtbaren Webs. Anschließend werden existierende Systeme und Strategien für Verteiltes Information Retrieval vorgestellt und im Hinblick auf deren Eignung für das WWW untersucht. Hieraus werden konzeptionelle Anforderungen für ein System für Verteiltes Information Retrieval über nicht-kooperativen Suchservern im WWW abgeleitet.

3.1. Das Unsichtbare Web

3.1.1. Dynamische Inhalte

Betrachtet man die Entwicklung des WWW, läßt sich ein klarer Trend dahingehend erkennen, daß immer mehr Inhalte auf dynamisch erzeugten Seiten präsentiert werden.

Eine Seite p wird als *dynamisch* bezeichnet, wenn ein Teil oder der gesamte Inhalt von p erst zur Laufzeit des Web-Servers generiert wird z.B. nachdem eine Anforderung für p von einem Server empfangen wurde. Die Seite p wird dabei von einem Programm erzeugt, das entweder client- oder serverseitig ausgeführt wird. Eine *statische* Seite hingegen existiert bereits vollständig auf einem Server und kann sofort an einen Client übertragen werden, sobald eine entsprechende Anforderung im Server eintrifft.

Für die in dieser Arbeit vorgenommenen Betrachtungen sind nur solche dynamischen Seiten relevant, die auf der Basis einer konkreten Benutzereingabe erzeugt werden. Ein typisches Szenario hierfür ist, daß als Ergebnis einer clientseitig erzeugten, formularbasierten Suchanfrage relevante Datensätze serverseitig aus einer Datenbank ausgelesen, für die WWW-Präsentation aufbereitet und an den Client zurückversendet werden.

Eine weitergehende Klassifizierung von dynamischen Seiten – z. B. Seiten, die als das Ergebnis einer Cookie-basierten Personalisierung erzeugt werden, oder Seiten, die über Applets regelmäßig entfernte aktuelle Informationen wie Aktienkurse abfragen

und clientseitig aufbereiten – findet sich in [95]. Im folgenden werden unter dynamischen Seiten ausschließlich jene Seiten verstanden, die als Folge einer zur Laufzeit durchgeführten formularbasierten Benutzereingabe erzeugt werden.

Der dynamisch aufbereitete Teil des WWW, wird oft als das „Unsichtbare Web“ bezeichnet (im Englischen werden die Bezeichnungen „Invisible Web“, „Hidden Web“oder „Deep Web“ verwendet) [7]. Eine solche bedarfsorientierte Informationsaufbereitung erlaubt eine gezielte Recherche in umfangreichen Web-Sites oder Repositorien. Oft enthält das Unsichtbare Web spezialisierte Nischeninformationen von hoher Qualität.

Jedoch existieren in der Regel keine Links auf dynamisch erzeugte Seiten, was es den Crawlern der Suchmaschinenbetreiber so gut wie unmöglich macht, durch automatisierte Indexierungsverfahren eine systematische und vollständige Inhaltserschließung von dynamischen Web-Sites vorzunehmen. Die Informationen, die auf den statischen bzw. verlinkten Seiten (z. B. auf der Startseite) von Web-Sites mit versteckten Inhalten bereit gestellt werden, reichen meist nicht aus, um eine angemessene Gewichtung bei Verwendung einer generellen Suchmaschine zu gewährleisten.

Dennoch stellt der effiziente und vollständige Zugriff auf die Ressourcen des Unsichtbaren Webs eine wichtige Voraussetzung für zukünftige globale digitale Bibliotheken dar. Die Kenntnis der Struktur und des Umfangs des Unsichtbaren Webs kann helfen, um hieraus Strategien für eine verbesserte Erschließung abzuleiten.

3.1.2. Umfang des Unsichtbaren Webs

Von der Firma BrightPlanet wurde eine umfangreiche Studie über die Struktur des Unsichtbaren Webs angefertigt [10]. Diese wurde im Juli 2000 veröffentlicht und versucht den enormen Umfang der im Unsichtbaren Web vorhandenen Datenmenge abzuschätzen. Einige für die folgenden Betrachtungen wichtige Ergebnisse sind:

- Die insgesamt im Unsichtbaren Web verfügbare Datenmenge wird auf 7500 Terabyte geschätzt. Gemessen an der Datenmenge des indexierbaren Webs von insgesamt etwa 15-20 Terabytes ergibt sich somit für das Unsichtbare Web ein Umfang, der um den Faktor 400 bis 500 größer ist.

- Die Gesamtanzahl von Dokumenten, auf die über das Unsichtbare Web zugegriffen werden kann, wird auf 550 Mrd. geschätzt. Die Anzahl indexierbarer Web-Seiten beträgt dagegen nur 800 Mio bis 1 Mrd.

- Die Anzahl der Web-Sites, die versteckte Inhalte enthalten, wird auf 100.000 geschätzt.

- Im Durchschnitt wird auf Web-Sites mit versteckten Inhalten 50% häufiger zugegriffen als auf indexierbare Web-Sites. Dennoch haben sie einen geringeren Bekanntheitsgrad und sind weniger stark durch Hyperlinks verknüpft.

- Die Datenmenge des Unsichtbaren Webs wächst in einem schnelleren Umfang als die des indexierbaren Webs.

- 95% des Unsichtbaren Webs ist öffentlich und kostenfrei zugänglich, eine vorhergehende Registrierung ist nicht notwendig.

3.1.3. Klassifizierung von Ressourcen des Unsichtbaren Webs

Eine Klassifizierung von Web-Sites mit versteckten Inhalten – diese werden im folgenden als *Suchserver* bezeichnet (Eine detaillierte Definition von Suchservern findet sich in Abschnitt 3.2.2) – läßt sich naturgemäß aufgrund der Heterogenität der individuell enthaltenen Daten nur sehr schwer vornehmen. Die Bandbreite reicht von generellen Suchmaschinen wie AltaVista oder Google, über Bibliotheks-OPACs bis hin zu sehr spezialisierten Informationsdiensten wie Zugfahrplanauskünften oder Online-Übersetzungshilfen. In [10] wird eine Einteilung von Suchservern in zwölf verschiedene Klassen vorgenommen.

Einen Anhaltspunkt für mögliche Klassifikationsmerkmale bietet der Grad der thematischen Geschlossenheit (Kohäsion) der Dokumente untereinander, die über einen spezifischen Suchserver recherchiert werden können. Suchserver, deren recherchierbare Dokumente sich z. B. ausschließlich mit altägyptischer Geschichte der sechsten Dynastie befassen, weisen beispielsweise eine hohe Kohäsion auf, wohingegen generelle Suchmaschinen, deren suchbarer Index sich über einen Großteil aller im WWW verfügbaren Dokumente erstreckt, sicherlich nur eine sehr niedrige Kohäsion aufweisen. Um den Grad der Kohäsion zu bestimmen, kann beispielsweise das in [128] beschriebene Verfahren angewendet werden.

Unternehmen wie Lycos[1], The Big Hub[2] oder Direct Search[3] bieten redaktionell aufbereitete Kataloge von thematisch spezialisierten Suchservern an. Diese werden – wie aus herkömmlichen Web-Katalogen bereits bekannt (siehe Abschnitt 2.3.1) – manuell klassifiziert und in eine vorgegebene Hierarchie von Kategorien eingeordnet. Neben der Kategorie und einer kurzen inhaltlichen Beschreibung der Suchserver als Ganzes sind in der Regel keine weiteren Eigenschaften aufgeführt. Die Anzahl der pro Katalog erfaßten Suchserver reicht von etwa 1.500 bis hin zu 30.000.

[1] Lycos Searchable Databases: http://dir.lycos.com/Reference/Searchable_Databases [14. Nov. 2001]
[2] The Big Hub: http://www.thebighub.com [14. Nov. 2001]
[3] Direct Search: http://gwis2.circ.gwu.edu/~gprice/direct.htm [14. Nov. 2001]

3.2. Verteilte Suche im WWW

3.2.1. Problematik

Eine verteilungstransparente Integration heterogener Suchdienste muß mit dem Wachstum des Unsichtbaren Webs Schritt halten können. Dies kann nur von einer verteilten Architektur wie z. B. einer Metasuchmaschine (siehe Abschnitt 2.3.4) geleistet werden.

Die Implementierung einer Metasuchmaschine für große Mengen spezialisierter Suchserver wird allerdings durch einige Randbedingungen erschwert. So wünschen viele Informationsanbieter nicht, daß automatisierte Anfragen an ihre Server gestellt werden. Gründe hierfür sind neben der Ausnutzung fremder Ressourcen für eigene Zwecke auch kommerzielle Aspekte. So finanzieren sich viele Web-Sites über Banner-Werbung, die im Falle eines automatisierten Zugriffs ungelesen bleibt.

Im folgenden soll allerdings nur auf die technischen Aspekte eingegangen werden. Die meisten Metasuchmaschinen im WWW verfolgen eine einfache Broadcast-Strategie, bei der eine Benutzeranfrage an alle zugrundeliegenden Suchserver weitergeleitet wird. Neben langen Antwortzeiten für die Benutzer ergibt sich hierdurch eine hohe und unangemessene Belastung von Kommunikationsnetz und Server.

Es ist also notwendig, eine Selektion der potentiell relevantesten Server vorzunehmen und die Anfragen gezielt an diese weiterzuleiten. Betrachtungen hierzu finden sich in Abschnitt 3.3. Des weiteren ist es erforderlich eine automatisierte Anfrageverarbeitung an den zugrundeliegenden Suchservern, sowie die Erzeugung einer global gewichteten Resultatliste vorzusehen.

In den folgenden Abschnitten erfolgt eine Diskussion der Aufgaben und der Komponenten von Verteiltem Information Retrieval (VIR).

3.2.2. Eine Spezifikation von Verteiltem Information Retrieval

Um den Begriff des *Verteilten Information Retrieval* zu konkretisieren, erfolgt zunächst eine Spezifikation der Komponenten und Aufgaben eines Verteilten IR - (VIR) Systems. Die Darstellung orientiert sich an der in [33] gegebenen Definition und wurde um den Begriff der Dokumente nach [40] erweitert.

Ein VIR-System besteht aus folgenden Komponenten:

- *Digitale Dokumente:* Ein digitales Dokument ist eine in sich abgeschlossene Informationseinheit, die eindeutig adressiert werden kann und deren Inhalt digital codiert und auf einem elektronischen Datenträger gespeichert ist, so daß er

mittels eines Rechners genutzt werden kann[4].

- *Dokumentserver:* Dokumentserver stellen die eigentlichen Dokumente bereit. Im WWW können Dokumentserver z. B. durch HTTP-basierte Web-Server repräsentiert werden, deren Aufgabe darin besteht, auf die Anforderung eines Dokumentes durch einen HTTP-Request mit der Auslieferung des Dokumentes in einer HTTP-Response zu reagieren.

- *Suchserver:* Aufgabe der Suchserver ist die Indexierung und das Durchsuchen von Dokumenten, die auf einem oder mehreren Dokumentservern aufliegen. Wird eine Suchanfrage q an einen Suchserver s übermittelt, so beantwortet er diese mit der Rückgabe einer Resultatliste $R = (D, o)$. Dabei ist D eine Menge von Dokumenten, die durch eine Ordnung o nach Relevanz zu q gewichtet ist. Suchserver können genereller Natur sein und eine große Dokumentenmenge abdecken (z. B. Suchmaschinen wie AltaVista oder Google) oder aber sehr spezialisierte Inhalte erschließen (z. B. Web-Sites wie den *1848 Flugschriften-Server*).

- *Mediator:* Ein Mediator (oder Broker) ermöglicht die Suche über eine Menge von Suchservern. Sei S eine Menge von Suchservern und q eine Benutzeranfrage. Die Arbeitsweise des Mediators setzt sich aus den folgenden aufeinander aufbauenden Arbeitsschritten zusammen:

  Die Aufgaben der Teilschritte sind:

  1. $Selektion : S, q \rightarrow S'$
     Aus einer Menge von Suchservern S wird in Abhängigkeit von einer Suchanfrage q eine Teilmenge $S'(S' \subseteq S)$ ausgewählt. Die Menge S' enthält jene Suchserver, die unter Berücksichtigung von systemabhängigen Faktoren wie maximal tolerierbarer Antwortzeit, Netzauslastung und Rechenzeit durchsucht werden können, um eine Resultatmenge R_m von maximal hoher Qualität zu produzieren.
  2. $Retrieval : S', q \rightarrow R_1, \ldots, R_{|S'|}$
     Die Anfrage q wird an alle Suchserver $s_i \in S'$ gestellt, wobei jeder eine Resultatliste R_i mit $R_i = (D_i, o_i)$ zurückliefert. Dies schließt die Weiterleitung von q an jeden Suchserver $s_i \in S'$ über entsprechende Protokolle mit ein sowie die Übersetzung von q in die Abfragesprache von s_i und die Analyse der Ergebnisseite zur Erzeugung der Resultatliste R_i.

[4] Wenn im folgenden von einem Dokument die Rede ist, so ist damit immer ein digitales Dokument gemeint.

3. $Merging : R_1, \ldots, R_{|S'|}, q \rightarrow R_m$
   Aus den einzelnen Resultatlisten R_i wird eine integrierte Resultatliste $R_m = (D_m, o_m)$ erzeugt, wobei gilt: $D_m = D_1 \cup D_2 \cup \ldots \cup D_{|S'|}$ und o_m ist ein effektives Ranking, so daß relevante Dokumente vor weniger relevanten Dokumenten gerankt werden.

Die Abbildung 3.1 stellt die Aufgaben und Komponenten von Verteilten Information Retrieval graphisch dar: Aus einer Menge von Suchservern $S = \{s_1, \ldots, s_n\}$ werden einige Suchserver selektiert, angefragt und aus den erhaltenen Resultatlisten eine integrierte Resultatliste R_m gebildet. Die Speicherkomponenten symbolisieren die durch einen Suchserver erschlossenen Dokumentserver.

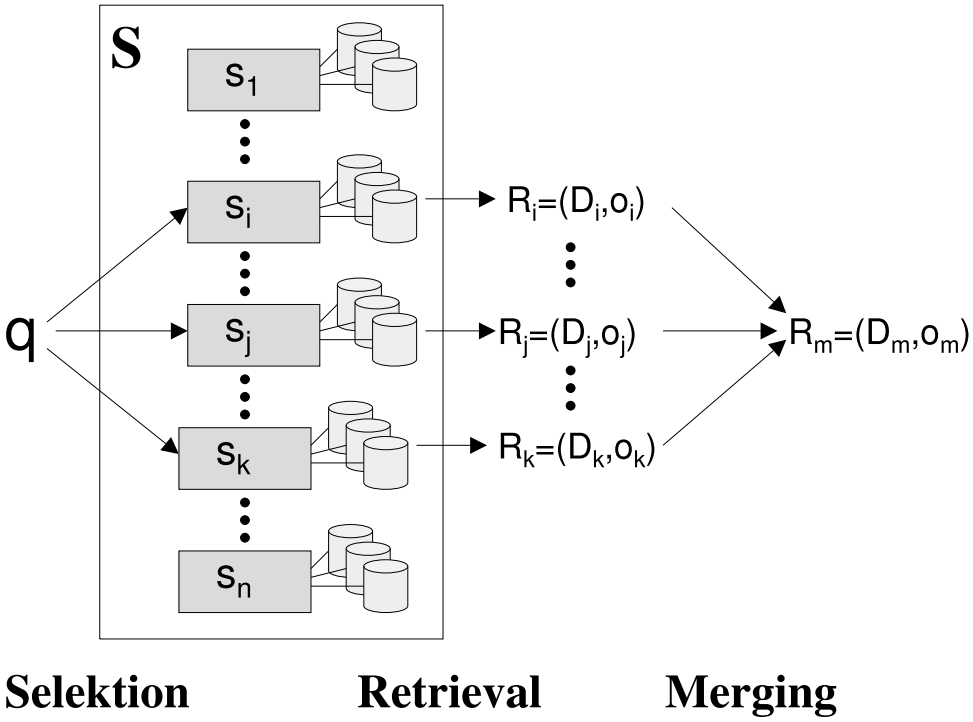

Abbildung 3.1.: Verteiltes Information Retrieval

Man beachte, daß auch dynamisch erzeugte Web-Seiten durch einen eindeutigen URL adressiert werden können und somit gültige Dokumente im Sinne der Definition darstellen. Eine Resultatliste stellt selbst wiederum ein dynamisch erzeugtes Dokument dar, somit kann ein Suchserver auch als ein spezieller Dokumentserver angesehen werden. Suchserver und Dokumentserver unterscheiden sich nach außen hin lediglich dadurch voneinander, daß ein Suchserver ein Resultat zurückliefert, dessen Inhalt eine definierte und interpretierbare Struktur aufweist, nämlich eine geordnete Liste von

Dokumentverweisen.

Wo genau die Grenze zwischen Suchservern und Dokumentservern gezogen wird, hängt davon ab, was als Dokument, also als die kleinste zu suchende Informationseinheit, angenommen wird. Man denke beispielsweise an ein Online-Wörterbuch, das nach Eingabe eines deutschen Begriffs in ein Web-Formular eine Liste der möglichen englischen Übersetzungen ausgibt. Interpretiert man die Datensätze der möglichen Übersetzungen in der zugrundeliegenden Datenbank als eigenständige Dokumente, so erfüllt eine solche Web-Site die Kriterien eines Suchservers – es wird eine geordnete Liste von relevanten Dokumenten erzeugt und zurückgeliefert. Interpretiert man jedoch die dynamisch erzeugte Webseite aller gefundenen Übersetzungen als ein einziges Dokument, so stellt diese Web-Site lediglich einen Dokumentenserver dar – auf eine bestimmte Anforderung wird das zugehörige Dokument zurückgeliefert.

3.3. Selektion relevanter Suchserver

Mit dem Anwachsen des Unsichtbaren Webs spielen zunehmend Strategien eine Rolle, die eine Menge S von verteilten Suchservern im Hinblick auf eine Suchanfrage q gewichten. Grundlage der Gewichtung bilden Algorithmen, die für jeden der Suchserver $s \in S$ die Wahrscheinlichkeit abschätzen, Dokumente zu enthalten, die für die Anfrage relevant sein könnten. Im Anschluß an die Selektion der n relevantesten Suchserver kann die Anfrage an die ausgewählten Server weitergeleitet werden.

Der Prozeß der selektiven Anfrageweiterleitung (selective query routing) ist notwendig, weil die Weiterleitung einer Anfrage an alle Suchserver der Menge S, wie bei einfachen Metasuchmaschinen üblich, lange Antwortzeiten zur Folge hat. Außerdem stellt dies einen verschwenderischen Umgang mit Bandbreiten sowie mit eigenen und fremden Systemressourcen dar. Ein solcher Ansatz skaliert nicht mit der Anzahl der zugrundeliegenden Suchserver und ist insbesondere dann nicht akzeptabel, wenn die Menge der zu durchsuchenden Suchserver ein gewisses Maß übersteigt. Deshalb reduzieren die meisten der im Web betriebenen Metasuchmaschinen die Anzahl der integrierten Systeme auf höchstens 15 bis 20, je nach Kapazität der einzubindenden Web-Server, um einen akzeptablen Kompromiß zwischen Antwortzeit und Informationsabdeckung zu erreichen.

Insbesondere für Suchserver, deren Dokumente nur einen thematisch spezialisierten Teilbereich abdecken, macht die Einbindung durch einen generellen Anfrage-Broadcast keinen Sinn. Deshalb ist es wichtig, bereits vor dem Versenden der Anfrage mittels geeigneter Strategien eine kleine Menge potentiell relevanter Suchserver zu identifizieren und die Anfrage selektiv an diese weiterzuleiten.

Die Selektion relevanter Suchserver stellt einen zentralen Aspekt dieser Arbeit dar. Deshalb erfolgt in diesem Abschnitt eine eingehende Betrachtung und Diskussion existierender Selektionsstrategien.

3.3.1. Untersuchung von Verfahren zur Selektion von Suchservern

Das Problem der Selektion von verteilten Datenquellen bzgl. ihrer Relevanz zu Suchanfragen wird eingehend in der Literatur beschrieben (siehe z. B. [26], [33], [73] und [125]). Es wird dabei u. a. als *collection selection*, *database selection*, *source selection*, *server selection* oder *resource discovery problem* bezeichnet. In [47] wird zusätzlich zwischen Selektion auf Systemebene und Selektion auf Inhaltsebene unterschieden. Dabei bezieht sich ersteres auf die physikalische Lokation eines Server, unabhängig von dem Inhalt der gespeicherten Information. Bei der inhaltsbasierten Selektion wird also für alle Server angenommen, daß sie die gleichen Suchkosten aufweisen; systemrelevante Faktoren wie Antwortzeit, Bandbreite etc. bleiben dementsprechend unberücksichtigt

Im folgenden werden zunächst zwei bekannte Verfahren vorgestellt - *CORI* und *GlOSS*. Die Effizienz beider Verfahren wurde bereits in umfangreichen Experimenten in künstlichen Testumgebungen wie INSPEC- aber auch in WWW-basierten Testumgebungen analysiert. Nach einem Vergleich der beiden Verfahren (dieser erfolgt vor allem im Hinblick auf deren Einsatz in Web-Szenarien) wird das System *QPilot* vorgestellt, ein Selektionsverfahren, das speziell auf die realen Anforderungen des WWW zugeschnitten ist.

3.3.1.1. GlOSS

GlOSS (Glossary of Servers Server) [54], [73], ist ein System, um eine Menge von Suchservern $S = \{s_1, \ldots, s_n\}$ bzgl. ihrer Relevanz zu einer Suchanfrage q zu gewichten. Die Suchserver enthalten dabei ausschließlich textuelle Dokumente. GlOSS wendet in der aktuellen Version das Vektorraummodell (vGlOSS) an und verwendet Term-Statistiken zur Berechnung des Relevanzgewichtes. Dabei speichert GlOSS die Statistiken über die Suchserver in zwei Matrizen $F(f_{i,j})$ und $W(w_{i,j})$, wobei i über die Anzahl der Suchserver und j über die Anzahl verschiedener Terme iteriert. F und W enthalten für jeden Suchserver s_i und jeden Term t_j jeweils einen Eintrag:

- $f_{i,j}$: entspricht der Anzahl der Dokumente aus s_i, die den Term t_j enthalten.
- $w_{i,j}$: entspricht der Summe der normalisierten Gewichte des Termes t_j über alle Dokumente des Suchservers s_i. $w_{i,j}$ kann berechnet werden, indem man

die Werte in den Spalten der Term/Dokument-Matrix von s_i für den Term t_j aufaddiert. Dabei wird davon ausgegangen, daß alle Suchserver den gleichen Algorithmus zur Berechnung eines Relevanzgewichtes zwischen Term und Dokument verwenden.

Die Gesamtfunktionalität wird in zwei Schritten erbracht. Zuerst müssen die Werte für die F- und W-Matrix für jeden Suchserver erstellt werden. Die Werte müssen dann zu einem zentralen Server exportiert und dort in die entsprechenden Zeilen der beiden Matrizen eingetragen werden. Anschließend wird unter Auswertung der Statistiken zu einer gegebenen Anfrage eine Rangliste der potentiell relevantesten Suchserver berechnet.

Im Vergleich zur vollständigen Datenmenge eines Suchservers enthalten die beiden Matrizen ein wesentlich geringeres Datenvolumen und können somit effizient zum zentralen Server versendet werden. Außerdem müssen Null-Häufigkeiten nicht explizit gespeichert bzw. exportiert werden.

Damit der zentrale GlOSS-Server stets über aktuelle Term-Häufigkeiten verfügt, müssen die externen Suchserver in periodischen Abständen ein Kollektorprogramm ausführen, das die notwendigen Informationen aus den lokalen Indizes extrahiert und an den zentralen Server versendet. Alternativ wurde auch das STARTS-Protokoll [53] (siehe auch Abschnitt 3.5.1) eingesetzt, um Zusammenfassungen der verteilten Suchserver zu versenden.

Um die Relevanz eines Suchservers s zu einer Benutzeranfrage q zu bestimmen, wird das Gütemaß $Goodness$ eingeführt:

$$Goodness(l, q, s) = \sum_{d \in Rank(l,q,s)} sim(q, d)$$

Dabei bestimmt $sim(q, d)$ die Ähnlichkeit zwischen der Anfrage q und einem Dokument d. $Rank$ bestimmt eine Menge von Dokumenten aus s, deren Ähnlichkeit über einem benutzerdefinierten Schwellwert l liegt, also $Rank(l, q, s) = \{d \in s \mid sim(q, d) > l\}$. Die $Goodness$ von q zu s ergibt sich somit aus der Summe der Ähnlichkeitswerte aller Dokumente, deren Ähnlichkeitswert zu q über l liegt.

Die Suchserver werden dann in der Reihenfolge der absteigenden $Goodness$-Werte sortiert, um das Ranking zu produzieren. Man muß allerdings beachten, daß das ideale Ranking $Ideal(l)$ von GlOSS mit den zur Verfügung stehenden Informationen nur angenähert werden kann, d. h. der Wert $sim(q, d)$ kann nicht exakt berechnet werden, weil die verfügbaren Daten in Form der beiden Matrizen F und W nur Abstraktionen der tatsächlichen Häufigkeitsverteilungen in den einzelnen Suchservern darstellen.

3.3.1.2. CORI

Der CORI-Ansatz (COllection Retrieval Inference network) [26] basiert auf Inferenz-Netzwerken, einem probabilistischen Ansatz für Information Retrieval [112], allerdings werden im sogenannten CORI-Netz nicht Dokumente, sondern vollständige Suchserver als Blätter aufgefaßt.

In CORI werden einzelne Suchserver in einem *database selection index* repräsentiert. Dieser enthält alle vorkommenden Terme und ihre Dokumenthäufigkeiten, d. h. dies entspricht der Information in der F-Matrix aus dem GlOSS-Ansatz.

Die Annahme (belief) $p(t_j \mid s_i)$, daß der Suchserver s_i bzgl. des Anfrageterms t_j relevant ist, berechnet sich durch:

$$T = \frac{df_{i,j}}{df_{i,j} + 50 + 150 \cdot sw_i/\overline{sw}}$$

$$I = \frac{log(\frac{|S|+0.5}{sf_j})}{log(\mid S \mid +1.0)}$$

$$p(t_j \mid s_i) = 0.4 + 0.6 \cdot T \cdot I$$

wobei:

- $df_{i,j}$: Anzahl der Dokumente im Suchserver s_i, die den Anfrageterm t_j enthalten,
- sf_j: Anzahl der Suchserver, die Dokumente mit t_j enthalten,
- $\mid S \mid$: Anzahl der Suchserver für das Ranking,
- sw_i: Anzahl verschiedener Terme im Suchserver s_i,
- $\overline{sw}$: Durchschnitt der sw-Werte über alle Suchserver.

In Anlehnung an das $tf \cdot idf$-Gewicht (siehe Abschnitt 2.1.3.3) kann der CORI-Ansatz zusammenfassend als $df \cdot isf$- Gewicht beschrieben werden, wobei isf für *inverse server frequency* steht.

3.3.1.3. CORI und GlOSS im Vergleich

In [44] wurden verschiedene Experimente mit CORI und GlOSS durchgeführt. Dabei wurde im wesentlichen festgestellt, daß CORI gegenüber GlOSS eine effektivere Selektion durchführt. GlOSS neigt dazu, jene Suchserver hoch zu gewichten, die eine große Anzahl von Dokumenten aufweisen. Dies wird offenbar durch die einfache

Aufaddierung der Einzelgewichte $sim(q, d)$ in der $goodness$-Formel bewirkt. In CORI bleibt das produzierte Ranking von der Größe des Suchservers unbeeinflußt. Es liegt die Vermutung nahe, daß dieser Effekt durch die Berücksichtigung der Termanzahl pro Suchserver, also mittels des Wertes $sw/\overline{sw}$, kompensiert wird. Jedoch wird gezeigt, daß dies offenbar nur einen geringen Einfluß auf die Gesamtperformanz hat. Die Frage, woher diese günstige Eigenschaft von CORI resultiert, bleibt offen.

Ein weiterer Nachteil von GlOSS besteht in der Abhängigkeit von serverspezifischen Gewichtungsmethoden, wie sie in der Matrix W von GlOSS repräsentiert sind. Wenn sich ausschließlich auf Term/Dokument-Häufigkeiten verlassen wird (also die F-Matrix von GlOSS oder die *df*-Werte von CORI), so erreicht man eine Unabhängigkeit der Server-Selektion von serverspezifischen IR-Modellen, Gewichtungs- und Indexierungsstrategien.

Die in [44] untersuchten $df \cdot isf$ basierten Selektionsstrategien, von denen CORI einen Repräsentanten darstellt, berücksichtigen keine Server-spezifischen Gewichtungsverfahren und erscheinen somit als geeignetere Kandidaten für Server-Selektionen in heterogenen Umgebungen wie dem WWW.

Dies wird durch eine neue Untersuchung von [35] bestätigt, in der vGlOSS, verschiedene Varianten von CORI sowie ein weiteres Selektionsverfahren – das Verfahren CVV (Cue Validity Variance) [125] – in einer simulierten Web-Umgebung gegeneinander ausgetestet werden. Da die Betreiber von Web-Suchservern im allgemeinen keine Häufigkeitsstatistiken bereitstellen, wurden die Term/Dokument-Häufigkeiten für die Experimente durch das Versenden von Probeanfragen ermittelt. Eine Diskussion von Probeanfragen findet sich in Abschnitt 3.5.3 der Arbeit.

3.3.1.4. QPilot

Ein System, das im stärkerem Maße als CORI und GlOSS versucht, die realen Anforderungen des Unsichtbaren Webs zu berücksichtigen, ist das System QPilot [110]. QPilot steuert die Selektion von themenspezifischen Suchservern und basiert auf einer Technik, die von den Autoren als „Themen-zentriertes Anfrage-Routing“ (topic-centric query routing) bezeichnet wird. In QPilot wird ein Suchserver nicht durch den vollständig indexierten Volltext seiner enthaltenen Dokumente beschrieben, wie dies in den Systemen CORI oder GlOSS der Fall ist. Vielmehr wird ein Suchserver nur durch eine relativ kleine Anzahl abstrakter thematischer Begriffe charakterisiert. QPilot verwendet unterschiedliche Strategien, um ausgehend von dem URL der Start-Seite eines spezialisierten Suchservers diese Begriffe automatisch zu generieren:

- Die *Front-Page Methode*: Auslesen der Begriffe auf der Front-Page, d. h. die

Web-Seite, die die Suchschnittstelle zum Suchserver enthält (z. B. über ein HTML-Formular). Jeder vorkommende Begriff wird zusammen mit seiner Häufigkeit in einem Index gespeichert.

- Die *Back-Link Methode*: Hierzu werden Web-Seiten untersucht, die Links auf die Front-Page von Suchservern enthalten. Diese werden als Backlink-Seiten bezeichnet. Von den Links der Backlink Seiten werden die Begriffe aus dem unterliegenden Volltext extrahiert und zusammen mit deren Häufigkeiten in dem Index abgespeichert.

- Die *Database-Sampling-Methode*: Es werden Probeanfragen an die Suchschnittstelle der Suchserver versendet, und die Terme der zurückgelieferten Resulate werden samt ihrer Häufigkeiten im Index gespeichert.

Wird der Index mit einem oder einer Kombination dieser Verfahren erzeugt, so weist dieser immer noch ein wesentlich geringeres Volumen auf als ein Index, der den vollständigen Volltext aller zugrundeliegenden Datenbanken berücksichtigt, wie z. B. bei CORI oder GlOSS. Deshalb müssen weitere Anstrengungen unternommen werden, um die Wahrscheinlichkeit von Übereinstimmungen zwischen den Anfragetermen und dem Index zu erhöhen. Dies gilt insbesondere, wenn man berücksichtigt, daß reale Anfragen im Web größtenteils nur aus ein oder zwei Termen bestehen (hierzu existieren eine Reihe von Studien über die Benutzung von Suchmaschinen, siehe z. B. [67]).

QPilot verwendet hierzu die Technik der Anfrageerweiterung (Query Expansion): Bei dieser Technik wird die Suchanfrage über eine disjunkte Verknüpfung um zusätzliche Terme erweitert, die thematisch mit der Suchanfrage in Beziehnung stehen. Meist werden die zusätzlichen Terme aus einem themenspezifischen oder einem allgemeinen Thesaurus ausgelesen, wobei häufig Synonyme oder Oberbegriffe zu den ursprünglichen Suchanfragetermen verwendet werden.

Im Gegensatz zu einer solchen statischen thesaurusbasierten Abfrageerweiterung verwendet QPilot eine Technik, bei der dynamisch zum Abfragezeitpunkt thematisch verwandte Terme automatisch generiert werden. Diese werden aus dem Web selbst gewonnen und zwar unter Ausnutzung einer generellen Suchmaschine wie AltaVista. Hierzu wird der Suchbegriff an die Suchmaschine versendet und die gefundenen Dokumente[5] werden nach häufig vorkommenden Termen durchsucht. Übersteigt die Ko-Zitierungs-Häufigkeit der gefundenen Terme mit dem Anfragterm einen gewissen Schwellwert, werden diese in einer Menge W gespeichert. Zu jedem $w_i \in W$ wird auch die Anzahl c_i der Ko-Zitierungen mit q gespeichert. Anschließend wird die Originalanfrage q um die Terme w_i erweitert, und die Suchmaschinen s werden bzgl.

[5] Um die Antwortzeiten auf einem akzeptablen Niveau zu halten, werden eigentlich nur deren Snippets analysiert.

ihrer Relevanz zu der erweiterten Suchanfrage gewichtet. Zur Erzeugung einer solchen Rangliste von Servern wird das folgende *Goodness*-Maß verwendet. Dabei wird die Relevanz zwischen einem Server s und einer Menge $W = \{w_1, w_2, \ldots, w_n\}$ von Anfrageerweiterungstermen, die für q ermittelt wurden, bestimmt. f_i bezeichnet die Häufigkeit, mit der ein Erweiterungsterm w_i im Serverindex auftaucht:

$$Goodness(s, W) = \sum_{w_i \in W} f_i \cdot c_i$$

Das Verfahren sieht auch vor, die Erweiterungsterme w_i gemäß evtl. verschiedenen Bedeutungen der Anfrage q in Cluster zu gruppieren. In diesem Fall werden gemäß der identifizierten Cluster verschiedene Mengen W generiert und für jeden Cluster ein individuelles Ranking erzeugt.

3.4. Integration von Suchservern

3.4.1. Wrapper-Programmierung

Die Suchserver des Unsichtbaren Webs sind nach außen hin meist nur durch deren Suchschnittstellen sichtbar. Diese sind häufig durch ein HTML-Formular repräsentiert. Um eine Anfrage automatisiert auf Seiten der Suchserver durchzuführen und auszuwerten, sind eine Reihe von Vorbereitungen notwendig. So muß für jeden Suchserver ein entsprechendes Wrapperprogramm (siehe auch Abschnitt 2.3.4) bereitstehen. Dessen Aufgaben bestehen im einzelnen aus den folgenden Teilschritten:

1. Entgegennahme einer Anfrage,
2. Übersetzung der Anfrage in die Abfragesprache des Suchservers,
3. Initiierung der eigentlichen Abfrage über die Suchschnittstelle des Suchservers,
4. Entgegennahme der vom Suchserver zurückgelieferten Resultatseite,
5. Erkennung und Extraktion der einzelnen Trefferdokumente aus der Resultatseite,
6. Rückgabe der Liste der Trefferdokumente in einheitlich aufbereiteter Form.

Das Programmieren von Wrappern ist ein sehr aufwendiger Prozeß, der sich nur schwer automatisieren läßt. Das Hauptproblem besteht darin, daß die Suchschnittstellen und die Resultatseiten in erster Linie für Menschen entworfen wurden. Das Layout

der Web-Seite läßt kaum Rückschlüsse auf die Semantik der dargestellten Informationen zu. Dementsprechend stellt das „Trainieren“ von Programmen zum automatisierten Zugriff über eine Suchschnittstelle auf einen nicht-kooperativen Server ein nicht triviales Problem dar. Selbst einfachste Informationen, wie die Anzahl der gefundenen Dokumente, lassen sich aus einer HTML-Resulatseite nur äußerst schwer extrahieren (sofern diese Information auf den Resultatseiten überhaupt dargestellt ist) und deren Erkennung muß von Wrapper zu Wrapper stets neu angepaßt werden. Entsprechend weist auch nicht jeder Link auf der Resultatseite unbedingt auf ein gefundenes Dokument hin.

Als problematisch erweist sich auch die Tatsache, daß man in der Regel keinen Einfluß auf das Layout der Resultatseite hat, von der die Treffer extrahiert werden sollen. Bereits kleinste Änderungen im Layout, die von den Server-Betreibern vorgenommen werden, können u. U. zur Folge haben, daß die im Wrapper spezifizierten Muster zur Erkennung einzelner Treffer nicht mehr greifen und so eine Reimplementierung von einzelnen Wrappern unumgänglich machen.

Verschiedene Metasuchsysteme sowie spezialisierte Crawler-Programme verwenden Werkzeuge zur automatisierten bzw. semiautomatisierten Generierung von Wrapperprogrammen. Ein Beispiel hierfür ist das Ariadne-System [3], das einzelne Teilanfragen gemäß ihrer Semantik auf verschiedene spezialisierte Web-basierte Suchserver verteilt, um hieraus integrierte Resultate auf komplexe Anfragen zu generieren.

Das System *HiWE (Hidden Web Exposer)* [95] analysiert die HTML-Formulare der Suchschnittstellen von Web-basierten Suchservern, um hieraus Wrapperprogramme zur Anfragedurchführung zu generieren. Die Wrapper werden verwendet, um Probeanfragen an die Suchserver zu versenden und aus dem Volltext der Resultatseiten einen Index zu generieren. Eine tiefergehende Analyse der Resultatseiten findet nicht statt. Diese werden nur unter Verwendung von verschiedenen Heuristiken dahingehend untersucht, ob evtl. gar keine Dokumente gefunden wurden oder Fehlermeldungen enthalten sind.

Das Werkzeug *WysiWyg Web Wrapper Factory (W4F)* [101] erlaubt die Spezifikation von Wrappern über eine graphische Benutzeroberfläche. Hierzu werden sogenannte *Extraction Rules* deklariert, über die die strukturelle Analyse von Web-Seiten gesteuert wird. Extrahierte Daten können des weiteren über spezifizierte Abbildungsvorschriften auf eine Zielstruktur abgebildet werden. W4F basiert allerdings auf proprietären Deklarationssprachen. Das Werkzeug ANDES [87] kann ebenfalls zur Extraktion von Web-Daten eingesetzt werden, verwendet hierfür allerdings die offenen Standards XML und XSLT zur Deklaration von strukturellen Erkennungsregeln und den Prozeß der anschließenden Wrappergenerierung. Um die Wrapper robuster gegenüber potentiellen Layout-Änderungen zu machen, werden erkannte HTML-Stukturen hierfür auf

äquivalentes und syntaktisch korrektes XHTML abgebildet.

Für zukünftige Anwendungen kann erwartet werden, daß, bedingt durch die verstärkte Durchdringung des Webs mit semantischen Auszeichnungen, wie sie durch den Einsatz von XML zumindest teilweise gewährleistet wird, sich der Prozeß der automatischen Wrappergenerierung vereinfachen wird.

3.4.2. Eigenschaften von Web-basierten Suchschnittstellen

Eine grundlegende Forderung bei der Zusammenfassung unterschiedlicher Suchschnittstellen ist, daß die Suchoptionen der einzelnen Suchserver weitestgehend vollständig erhalten bleiben sollen.

Dies stellt ein schwieriges Unterfangen dar, zumal sich die einzelnen Suchschnittstellen in Funktionalität und Aufbau stark voneinander unterscheiden können. Im folgenden sind Kriterien aufgeführt, in denen sich Web-basierte Suchschnittstellen voneinander unterscheiden:

- *Boolesche Operatoren zur Verknüpfung von Suchbegriffen:* Die meisten Suchschnittstellen unterstützen die Booleschen Operatoren AND, OR bzw. AND NOT. Als weiterer boolescher Operator steht oft eine NEAR- Funktion zur Verfügung, die über die AND-Operation hinaus den Wortabstand zweier Begriffe zueinander in den Dokumenten berücksichtigt, was sich zur Reduzierung großer Treffermengen als sehr nützlich erweisen kann. Dabei kann sich die Größe des zu berücksichtigenden Wortabstandes wiederum von Suchserver zu Suchserver unterscheiden (bei AltaVista beträgt dieser beispielsweise 10 Worte).

- *Formulierung und Verarbeitung des Suchbegriffes:* Die Suchbegriffe stellen den eigentlichen, frei formulierbaren Anteil einer Suchanfrage dar. Einige Suchserver unterstützen Rechts- oder Links-Trunkierung durch die Verwendung von Wildcards (oft mittels des *-Zeichens) bzw. die explizite Spezifikation von Wortketten als eigener Suchbegriff (z. B. durch Setzen von Anführungszeichen). Auf die Verarbeitung von einzelnen Suchbegriffen durch die Suchmaschine hat jedoch der Anfragende oft keinen Einfluß, z. B. die automatische Entfernung von Stopworten, die zusätzliche Berücksichtigung syntaktischer Varianten (Groß- und Kleinschreibung, Wortstammreduktion, n-grams, Behandlung von Sonderzeichen, etc.).

- *Vorgegebene Suchfelder zur Charakterisierung des Suchbegriffes:* Suchbegriffe können oft mit vorgegebenen Suchfeldern kombiniert werden, durch die die Semantik des Suchbegriffs angenähert werden kann. Hierdurch wird dessen Interpretation durch den zugrundeliegenden Suchserver festgelegt und der Suchraum

eingeschränkt. Die häufigsten Beispiele für Suchfelder finden bei der Suche über bibliographische Metadaten Anwendung, wie z. B. *Autor*, *Titel*, *Stichwort* etc. Viele generelle Suchmaschinen sehen auch Felder zur Spezifikation von Suchen über Internet-relevante Konzepte, wie email-Adressen, URLs oder Hyperlinks vor.

Um zu gewährleisten, daß Metasuchmaschinen in der Lage sind, die volle Vielfalt der Abfragemöglichkeiten aller zugrundeliegenden Suchserver auf einer integrierten Suchschnittstelle zu bewahren, müssen die Optionen einheitlich beschrieben werden können.

Um diese Forderung zu erfüllen, ist es notwendig, ein Spezifikationsmodell für Web-basierte Suchschnittstellen zu verwenden, mit dessen Hilfe die zur Anfrageformulierung verfügbaren Optionen einer jeden Suchschnittstelle eindeutig und einheitlich beschrieben werden können. Ein solches Modell muß leicht für neue Suchschnittstellen adaptiert werden können und skalierbar im Hinblick auf neu hinzukommende Anfragemöglichkeiten sein.

Hierzu kann beispielweise das im *ADMIRE-System* verwendete Modell von [65] verwendet werden. Das Modell basiert auf verschiedenen Definitionen von Basiselementen, die direkt aus den Optionen von Web-basierten Suchschnittstellen abgeleitet werden können. Wichtige Basiselemente des Modells sind beispielsweise:

- *Terme*, die in einen Eingabebereich eingegeben werden können.
- *Suchfelder*, mit denen der Suchbereich für einen Term eingeschränkt werden kann (Typische Beispiele sind *Titel*, *Autor*, *Volltext*).
- *Logische Operatoren* zur Kombination zweier Terme, z. B. UND, ODER.

Über die Basiselemente hinaus stellt das Modell weitere komplexe Definitionen bereit, die durch die Kombination von einfacheren Definitionen zusammengesetzt sind, z. B. Logische Ausdrücke, Boolesche Ausdrücke bis hin zu sogenannten Anfrage-Ausdrücken, die eine Suchoberfläche vollständig spezifizieren.

Des weiteren stellt jede Suchschnittelle unterschiedliche Optionen mit unterschiedlicher Semantik bereit, manche erlauben z. B. die Trunkierung von Suchbegriffen oder die Verknüpfung von Suchbegriffen durch boolesche Operatoren, andere wiederum nicht.

3.4.3. Erzeugung einer integrierten Resultatliste

Jeder individuelle Suchserver gewichtet seine Resultate nach unterschiedlichen Methoden. Zwar verwenden die meisten Suchserver Varianten der in Abschnitt 2.1 vorgestellten Modelle (Boolesches Modell und Vektorraummodell), der konkrete Algorithmus, wie die Gewichtung im Detail vorgenommen wird, ist in der Regel jedoch nicht öffentlich zugänglich[6].

Eine Suchanfrage q, die an n verschiedene Suchserver weitergeleitet wird, liefert somit auch n verschiedene Resultatlisten, über deren Gewichtungskriterien keine Informationen vorhanden sind. Hieraus muß eine integrierte Liste erzeugt werden, die die gefundenen Dokumente verschiedener Suchserver nach ihrer Relevanz zu q global einheitlich gewichtet. Eventuell existiert auch eine Vorgabe, wieviel Dokumente die integrierte Resultatliste maximal enthalten darf, dementsprechend muß in Abhängigkeit von diesem Wert für jeden Suchserver auch die Anzahl von Dokumenten bestimmt werden, die von ihm in die integrierte Resultatliste eingebracht werden dürfen.

Das Problem der globalen Zusammenfassung lokal erzeugter Resultatlisten wird in der Literatur auch als *Collection Fusion Problem* bezeichnet. Betrachtungen hierzu finden sich in [114] und [122].

3.5. Kooperation von Suchservern

Die Erfassung von dynamischen Inhalten im WWW kann durch die Suchserver unterstützt werden, indem diese ein kooperatives Verhalten aufweisen.

In einem Verbund von kooperativen Suchservern erfolgt der Austausch von Anfragen und Resultaten in einem vordefinierten Format, das automatisiert interpretiert werden kann, z. B. durch die Verwendung eines gemeinsamen Protokolls, wie dem in der bibliothekarischen Welt gebräuchlichen *Z39.50 Protokoll* [7].

Darüber hinaus kann die Kooperation aber auch den bedarfsgesteuerten Austausch von vollständigen oder abstrahierten Term-Indizes miteinschließen, wie dies z. B. bei CORI und GlOSS der Fall ist. Einige Protokolle (z. B. das im folgenden Abschnitt beschriebene STARTS-Protokoll) unterstützen neben inhaltsbezogenen Daten auch den Austausch von serverspezifischen Metadaten wie die Gesamtanzahl der über den Suchserver recherchierbaren Dokumente. So steigt die Qualität einer verteilten Suche in dem Maße an, in dem sich die Suchserver kooperativ gegenüber einem hierauf aufsetzenden Metasuchsystem verhalten. Je umfangreicher die Datenmenge ist, die ei-

[6] Abhilfe können hier Verfahren schaffen, die eine nachträgliche Bestimmung der Gewichtungsmethoden von Suchmaschinen ermöglichen ([82]).

[7] Z39.50: http://lcweb.loc.gov/z3950/agency/ [14. Nov. 2001]

nem Metasuchsystem über jeden integrierten Suchserver zur Verfügung steht, desto zuverlässiger lassen sich im Verteilten Information Retrieval die Schritte Selektion, Retrieval und Merging vornehmen.

Suchserver können auch gegenüber den Crawlern genereller Suchmaschinen ein kooperatives Verhalten aufweisen. Hierzu müssen die Server entsprechend konfiguriert werden, damit diese bei Erkennung eines Crawler-Zugriffs auf deren Web-Site einen Export des gesamten Datenbestandes in einem indexierbaren Format veranlassen (z. B. durch die temporäre Bereitstellung von statischen HTML-Seiten), und so dessen Erfassung durch einen zentralen Suchmaschinenbetreiber veranlassen. Abgesehen von einigen wenigen E-Commerce Web-Sites wird eine derartige Strategie von den meisten Web-Servern jedoch nicht unterstützt.

Unterstützen dahingegen verschiedene kooperative Suchserver ein einheitliches Retrieval-Protokoll zum Austausch von Anfragen und Datensätzen, erleichtert dies die Implementierung von hierauf aufsetzenden Wrapperprogrammen, da dem Metasuchsystem eine einheitliche Sicht auf die zu integrierenden Datenquellen ermöglicht wird. So müssen Wrapperprogramme nicht mehr für jeden neuen Suchserver angepaßt werden, und deren Implementierung läßt sich weitestgehend automatisiert bewerkstelligen.

Im folgenden Abschnitt wird ein Protokoll vorgestellt, das zur Standardisierung der Kooperation von Suchservern entwickelt wurde. Anschließend werden Architekturen vorgestellt und diskutiert, in denen Suchserver in einem Verbund kooperieren.

3.5.1. STARTS

STARTS (Stanford Protocol Proposal for Internet Retrieval and Search) [53] ist ein Protokoll für Internet Retrieval, das die Universität Stanford in Zusammenarbeit mit verschiedenen Suchmaschinenbetreibern definiert hat. STARTS ermöglicht es, Anfragen an textbasierte Suchmaschinen und die zurückgelieferten Ergebnismengen einheitlich zu beschreiben, ähnlich wie dies mit Z39.50 für bibliographische Datenbanken vorgenommen wird. Dabei definiert STARTS eine Reihe von Metadaten, die es erlauben, den verfügbaren Datenbestand einer Internet Suchmaschine als Ganzes zu beschreiben, sogenannte *Content Summaries*. Diese enthalten z. B. Informationen über die Gesamtanzahl der enthaltenen Dokumente sowie alle enthaltenen Terme mit deren document-frequency-Werten.

STARTS unterstützt Verteiltes Information Retrieval somit für alle der in Abschnitt 3.2.2 beschriebenen Eigenschaften: Server-Selektion (durch die Content Summaries), Anfragebearbeitung (durch die vereinheitlichte Client-Server Interaktion) und die Integration der verschiedenen Treffermengen (durch die einheitliche Darstellung der

gefundenen Dokumente). Im STARTS-Protokoll werden die Informationen festgelegt, die zwischen den Internetsuchmaschinen und den hierauf aufsetzenden Metasuchern ausgetauscht werden müssen. Nicht festgeschrieben wird jedoch die Art und Weise der Formatierung oder des Transportes der ausgetauschten Daten[8].

3.5.2. Kooperative Architekturen

3.5.2.1. Beispielarchitekturen

Im Stanford Digital Library Projekt [9] werden Konzepte und Techniken entwickelt, die es ermöglichen, autonome Dienste in eine skalierbare Infrastruktur – den sogenannten Stanford Information Bus – einzubetten. Ein wichtiges Ziel ist es, die Interoperabilität von heterogenen Informationssystemen zu gewährleisten. Das bereits vorgestellte GlOSS-System (siehe Abschnitt 3.3.1.1) sowie das STARTS-Protokoll wurden im Rahmen dieses Projektes entwickelt. Sie ermöglichen sowohl die Erstellung als auch den Austausch von Metadaten und Indizes der in die Architektur integrierten Suchserver.

Das *Harvest-System* [25] ermöglicht den Aufbau einer verteilten Indexstruktur. Hierzu stellt Harvest Werkzeuge bereit, die es Servern erlauben ihre Dokumente lokal zu indexieren und die erzeugten Indexlisten an übergeordnete Server in einem Harvest-Verbund zu exportieren[9]. Dies erlaubt den Aufbau von hierarchischen, themenspezifischen Indizes. Harvest ist vor allem im wissenschaftlichen und universitären Umfeld relativ stark verbreitet und wird beispielsweise im Rahmen des Projektes *Dissertation Online* eingesetzt, um die universitätsübergreifende Recherche von Dissertationen in verschiedenen Fachgebieten zu gewährleisten.

Das System *Pharos* [38] sowie das von Zhu et. al. [129] entwickelte System sind ebenfalls auf die vollständige Verfügbarkeit der Volltexte aller Dokumente eines Suchservers angewiesen[10]. Mittels Verfahren zur automatischen Kategorisierung werden alle Dokumente eines Suchservers kategorisiert und die am häufigsten vergebenen Kategorien zur Beschreibung der Suchserver herangezogen. Das System von Zhu verwendet hierzu die Kategorienhierarchie eines prominenten Web-Katalogs, wohingegen Pharos auf das Klassifikationsschema der Library of Congress zurückgreift. Neben einer themenspezifischen Klassifikation werden in Pharos zusätzlich die Suchserver nach räumlicher und zeitlicher Abdeckung klassifiziert (z. B. welche Orte und Zeiträume werden in den Dokumenten erwähnt). Wie in Harvest werden die einzelnen Serverbe-

[8] In den aufgezeigten Beispielen wurden die Daten im Summary Object Interchange Format (SOIF) von Harvest spezifiziert. Bzgl. des Transportes ist in einer Web-Umgebung zu erwarten, daß der Transport primär über HTTP erfolgen wird

[9] Dabei wird auch die Indexierung von Postscript bzw. PDF-Dokumenten unterstützt.

[10] Beide Systeme wurden nur in experimentellen Umgebungen evaluiert.

schreibungen zum Aufbau von übergeordneten Servern herangezogen und damit eine mehrstufige Anfrageweiterleitung ermöglicht, was wiederum die Skalierbarkeit des Gesamtsystems erhöht. Erst werden die Anfragen mit den abstrahierten Indizes der übergeordneten themenspezifischen Server abgeglichen und anschließend werden diese an die eigentlichen Suchserver weitergeleitet.

3.5.2.2. Diskussion der vorgestellten Architekturansätze

Die Integration von Protokollen wie STARTS in einen Suchserver erfordert einen zusätzlichen Implementierungsaufwand von Seiten der Informationsanbieter. Insbesondere die Betreiber von spezialisierten Suchservern sind jedoch oft nicht bereit, diesen Aufwand zu leisten oder sind an einer Offenlegung ihrer internen Daten gar nicht erst interessiert.

Strategien zur Server Selektion wie GlOSS oder Systeme wie Pharos lassen sich allerdings nur dann erfolgreich auf das WWW übertragen, wenn umfangreiche Term-Häufigkeitsstatistiken über die enthaltenen Dokumente der einzelnen Suchserver vorhanden sind. Auch das Harvest-System erfordert eine clientseitige Installation von proprietärer Software.

Tatsächlich verhalten sich im realen Anwendungsszenario des WWW die Suchserver in der Regel jedoch nicht kooperativ gegenüber Indexierungs-Crawlern von Suchmaschinenbetreibern oder Metasuchsystemen, weshalb nach alternativen Strategien zur Erzeugung von Suchserver-Beschreibungen und zur automatisierten Erzeugung von Metadaten Ausschau gehalten werden muß.

Eine alternative Strategie, die nicht auf die Kooperation von Suchservern angewiesen ist, stellt das automatisierte Versenden von Probeanfragen an die Suchserver dar, um hieraus Term- bzw. Dokument-Häufigkeiten zu interpolieren. Eine Diskussion dieser Vorgehensweise erfolgt im folgenden Abschnitt.

3.5.3. Erzeugung von Inhaltsbeschreibungen durch Probe-Anfragen

Eine Möglichkeit zur automatisierten Erzeugung von Inhaltsbeschreibungen von Suchservern besteht im gezielten Versenden von Probe-Anfragen. Dabei werden Anfragen aus einer vorgegebenen Menge an den Suchserver versendet und die zurückgelieferten Resultatseiten bzw. die gefundenen Dokumente indexiert und hieraus eine Inhaltsbeschreibung erstellt (siehe hierzu [35],[95],[114]).

Nachteile dieser Vorgehensweise sind:

- Es müssen Wrapper vorhanden sein, die die Probe-Anfrage auf die Abfragesprache des Suchservers abbilden und evtl. die Resultatseite auswerten. Der Prozeß der Wrapper-Generierung läßt sich nur schwer automatisieren.
- Das erstellte Inhaltsmodell ist stark abhängig von der Auswahl der Probeanfragen, die gestellt wurden, und die extrahierten Terme spiegeln den eigentlichen Inhalt u. U. nur unzureichend oder sogar verfälscht wider.
- Das gehäufte Versenden von Probe-Anfragen an ein- und denselben Suchserver bedeutet eine starke Belastung einzelner Web-Server.
- Suchserver, die primär Multimediadokumente enthalten, lassen sich inhaltlich nur mit zusätzlichem Aufwand erschließen.

Das in Abschnitt 3.3.1.4 vorgestellte System QPilot führt in der Database-Sampling-Methode eine Versendung von Probe-Anfragen durch. Dabei erwies sich diese Strategie jedoch als unterlegen gegenüber der Front-Page und der Backlink-Methode im Hinblick auf die Qualität der Server-Selektion.

3.6. Anforderungen an eine VIR-Architektur für nicht-kooperative Suchserver im WWW

Ziel der Arbeit ist die Herleitung einer integrierten Architektur für Verteiltes Information Retrieval zur Erschließung von Ressourcen des Unsichtbaren Webs. Dabei sollen Metasuchtechniken zum Einsatz kommen, wobei die Funktionalität einer traditionellen Metasuch-Architektur erweitert wird, um zusätzlich die besonderen Gegebenheiten des Unsichtbaren Webs zu berücksichtigen.

So hat die in Abschnitt 3.1.2 vorgestellte Studie deutlich gemacht, daß die Anzahl von Web-basierten Suchservern, die themtisch spezialisierte Dokumente enthalten, exponentiell anwächst. Im Hinblick auf die Skalierbarkeit eines hierauf aufsetzenden Metasuchsystems kommt somit der Selektion von relevanten Suchservern eine besondere Bedeutung zu.

Eine weitere Eigenschaft des Unsichtbaren Webs besteht darin, daß der Großteil der Web-basierten Suchserver sich nicht kooperativ gegenüber Metasuchsystemen oder automatisierten Indexierungsversuchen verhält. Zwar existieren eine Reihe von Vorschlägen, wie beispielsweise das STARTS-Protokoll, um eine solche Kooperation zu unterstützen, jedoch sind deren Akzeptanz und aktuelle Verwendung im WWW als eher gering einzuschätzen. Dennoch ist gerade die Selektion von relevanten Suchservern (wie z. B. bei CORI und GlOSS) in starkem Maße abhängig von der Qualität der

über jeden Suchserver lokal vorhandenen Metadaten.

Im folgenden werden wichtige Anforderungen an eine VIR-Architektur für nichtkooperative Suchserver zusammengefaßt und gleichzeitig ein Ausblick auf die weiteren Abschnitte dieser Arbeit gegeben. In diesen werden Lösungen zu den identifizierten Teilbereichen erarbeitet, um hieraus abschließend eine integrierte Architektur abzuleiten.

Ein wichtige Anforderung besteht in der Generierung von Metadaten über nichtkooperative Suchserver. Die Metadaten sollen weitestgehend automatisiert unter Verwendung spezialisierter Crawler generiert werden. Hierzu müssen explizite Crawling-Strategien identifiziert und angewendet werden. Diese sollen aufgrund der in Abschnitt 3.5.3 genannten Nachteile ohne die Verwendung von Probeanfragen auskommen. Vielmehr sollen die Metadaten durch die systematische Auswertung von Informationen aus dem sichtbaren Teil des WWW gewonnen werden, indem die verteilte Indexierungsarbeit, die kollektiv im Sichtbaren Web geleistet wird, systematisch ausgewertet wird (z. B. durch die inhaltliche und strukturelle Analyse von Hyperlinks mittels HITS oder Pagerank).

Um eine einheitliche Verwendung und Interpretation von Metadaten für eine große Menge von verschiedenen Suchservern zu gewährleisten, ist es notwendig ein Schema bereitzustellen, mittels dessen Web-basierte Suchserver beschrieben werden können. Ein geeigneter Metadatensatz muß definiert werden, der Felder für sämtliche potentiell erreichbaren Informationen vorsieht, die dazu geeignet sind, einen Suchserver als Ganzes zu charakterisieren. Dieser soll sowohl inhaltlich beschreibende Terme als auch qualitätsorientierte Informationen umfassen, wie die Anzahl der Zugriffe in einem bestimmten Zeitraum oder die Anzahl der Backlinks. In Kapitel 4 wird ein Metadatensatz vorgestellt, der diesen Anforderungen genügt – dieser wird als Frankfurt Core bzw. FC bezeichnet. Darüber hinaus werden für einzelne Felder des Frankfurt Core Strategien identifiziert, um die spezifizierten Metadaten aus dem öffentlichen Teil des WWW automatisiert zu extrahieren. Der Frankfurt Core unterstützt auch eine Systematik zur wissensbasierten Darstellung von Suchfeldern (z. B. *Autor*, *Titel* etc.) von verschiedenen Suchservern. Dies kann für die Konstruktion einer integrierten Suchschnittstelle verwendet werden, indem die syntaktischen und semantischen Unterschiede der einzelnen Suchfelder durch Abbildung auf ein einheitliches Modell zur Wissensrepräsentation überwunden werden.

Die inhaltliche Auszeichnung von Suchservern kann auch durch Web-Mining Strategien unterstützt werden, indem beispielweise Web-Ressourcen mittels automatischer Kategorisierungsverfahren auf eine vordefinierte Menge von Kategorienbezeichnern abgebildet werden. In Kapitel 5 wird ein auf dem HITS-Algorithmus basierendes Clustering- Verfahren für spezialisierte Suchserver vorgestellt, sowie ein Verfahren

zur automatisierten Kategorisierung von Web-Ressourcen. Beide Verfahren werden durch Experimente evaluiert.

In Kapitel 6 wird eine Methode zur Selektion von Suchservern im WWW vorgestellt. Dabei wird auf der Basis der gesammelten Metainformationen in Abhängigkeit von einer Suchanfrage eine Selektion der relevantesten Suchserver durchgeführt. Die Qualität der Selektionsmethode wird hierzu in einer geeigneten Experimentumgebung evaluiert.

Die in den Kapiteln 4 – 6 gewonnenen Erkenntnisse werden dazu genutzt, um hieraus eine Architektur für spezialisierte Web-basierte Suchserver mit versteckten Inhalten herzuleiten. Dies setzt zum einen voraus, daß ein geeignetes Rahmenwerk geschaffen wird, um Suchserver ausreichend zu charakterisieren (Frankfurt Core, spezialisierte Metadatencrawler), und integriert zum anderen die im Rahmen dieser Arbeit identifizierten notwendigen Werkzeuge für Web-Mining und Server-Selektion in einer komponentenbasierten Metasuchmaschinen-Architektur. Die prototypische Realisierung des Gesamtsystems – dies beinhaltet sowohl die Darstellung der Einzelkomponenten als auch deren Zusammenspiel – erfolgt in Abschnitt 7.

4. Ein Metadatensatz zur Beschreibung von Suchservern

In diesem Kapitel wird ein neuer Metadatensatz für Web-basierte Suchserver vorgestellt – der *Frankfurt Core*. Zunächst werden konkrete Anforderungen an einen solchen Metadatensatz identifziert und überprüft, inwieweit diese von existierenden Metadatensätzen erfüllt werden. Anschließend erfolgt in Abschnitt 4.3 die Spezifikation der einzelnen Felder des Frankfurt Core. Die Verwendung des Frankfurt Core wird anhand eines vollständig ausgezeichneten Suchservers demonstriert. Darüber hinaus werden für einzelne Felder Strategien vorgestellt, um Metadaten automatisiert aus dem sichtbaren Teil des WWW zu extrahieren. Der Frankfurt Core unterstützt auch eine Systematik zur wissensbasierten und einheitlichen Repräsentation von Suchfeldern (z. B. *Autor*, *Titel* etc.) verschiedener Suchserver. Diese wird in Abschnitt 4.5 vorgestellt und deren Anwendung anhand eines Beispiels demonstriert.

4.1. Einleitung

Metadaten spielen eine wichtige Rolle im Verteilten Information Retrieval, insbesondere um vergleichende Aussagen über verschiedene Suchserver abzuleiten. Metadaten sind bedeutsam für alle drei der in Abschnitt 3.2.2 genannten Teilschritte Selektion, Retrieval und Merging:

1. Der Prozeß der Selektion von geeigneten Suchservern zu einer Suchanfrage ist in erheblichem Maße von der Qualität der insgesamt hierüber vorhandenen Informationen abhängig.

2. Um im Retrieval-Schritt eine Anfrage an den selektierten Suchservern durchzuführen, müssen die Suchoptionen der individuellen Suchschnittstellen bekannt sein (d. h. welche Suchfelder werden unterstützt, welche Operatoren können zur Anfrageformulierung herangezogen werden etc.), um wrappergestützt eine Übersetzung der Suchanfrage in die Abfragesprache des Suchservers vorzunehmen.

3. Merging, das Erzeugen einer integrierten Resultatliste, kann unterstützt werden, wenn es möglich ist, die Suchserver, aus denen die einzelnen Dokumente extrahiert wurden, über entsprechende Suchserver-spezifische Metadaten vergleichend zueinander in Bezug zu setzen.

Insbesondere für die Selektion gilt: Je mehr Metadaten über die Suchserver vorhanden sind, desto flexibler lassen sich verschiedene Selektionsstrategien implementieren. So können verschiedene Metasuchsysteme, die dieselbe Menge an Suchservern einbinden, die vorhandenen Metadaten als unterschiedlich bedeutsam erachten und dementsprechend eine individuelle Gewichtung der einzelnen Suchserver vornehmen. Soll beispielsweise eine bestimmte Metasuchmaschine verstärkt qualitätsorientierte Merkmale berücksichtigen, so können derartige Metadaten, wie die Anzahl der Backlinks eines Suchservers oder dessen Pagerankgewicht, entsprechend hoch gewichtet werden. Durch das umfangreiche Auszeichnen von Suchservern mit Metadaten erreicht man also eine Unabhängigkeit von der anzuwendenden Selektionsstrategie.

Im Hinblick auf die Skalierbarkeit des Gesamtsystems ist es wünschenswert, den Prozeß der Metadatengenerierung weitestgehend zu automatisieren. Im folgenden Abschnitt werden Anforderungen an Metadaten für Suchserver diskutiert. Anschließend wird für zwei Metadatensätze für Dokumentkollektionen überprüft, inwieweit die Anforderungen von diesen erfüllt werden.

In Abschnitt 4.3 wird ein neuer Metadatensatz für Web-basierte Suchserver – der sogenannte Frankfurt Core (FC) – abgeleitet. Dieser erfüllt die spezifizierten Anforderungen und ist dazu geeignet Suchserver einheitlich zu beschreiben. Es werden die einzelnen Felder des Frankfurt Core vorgestellt und anschließend wird dessen Verwendung anhand eines Beispiels illustriert. In Abschnitt 4.4 werden Strategien vorgestellt, um Metadaten für einige der Felder des Frankfurt Core automatisiert zu extrahieren. Darüber hinaus unterstützt der Frankfurt Core eine Systematik zur wissensbasierten Repräsentation von Suchfeldern verschiedener Suchserver. Diese wird in Abschnitt 4.5 vorgestellt.

4.2. Anforderungen an einen Metadatensatz für Web-basierte Suchserver

In Abschnitt 2.5.2.1 wurden allgemeine Anforderungen an einen Metadatensatz für Web-Ressourcen formuliert, gemäß derer die initiale Spezifikation des Dublin Core (DC) sowie dessen Weiterentwicklung erfolgte[1].

[1] Der Einsatz des DC ist allerdings nicht zwangsläufig allein auf Web-Ressourcen beschränkt, vielmehr kann dieser auch zur Auszeichnung von nicht-elektronischen Ressourcen verwendet werden.

Web-basierte Suchserver sind über den URL ihrer Suchschnittstelle eindeutig adressierbar und stellen eigenständige Web-Ressourcen dar. Somit besitzen die Anforderungen ebenfalls Gültigkeit für einen Metadatensatz zur Beschreibung von Suchservern. Insbesondere die Forderung nach Interoperabilität ist innerhalb des WWW-Umfeldes von hoher Bedeutung und kann durch das Abstützen auf offene Standards für Metadaten (wie beispielsweise RDF) gewährleistet werden. Allerdings läßt sich mit den 15 Elementen des Dublin Core und dessen Qualifizierern nur eine rudimentäre Auszeichnung von Suchservern vornehmen. Der DC ist nur im unzureichendem Maße dazu geeignet, um Verteiltes Information Retrieval effektiv zu unterstützen.

Um beispielsweise eine inhaltliche Selektion von Suchservern auf der Basis einer CORI- oder GlOSS-Gewichtung durchzuführen, müssen Termstatistiken für einzelne Suchserver verfügbar sein. Werden Suchserver dahingegen ausschließlich unter Verwendung von DC-Feldern ausgezeichnet, so erlauben lediglich die Felder *Title*, *Coverage*, *Description* und *Subject* eine inhaltliche Einschätzung der über den Suchserver verfügbaren Dokumente. Es ist also zu erwarten, daß die durch den Dublin Core verfügbare Informationsmenge nicht ausreicht, um eine faire automatisierte Selektion zu gewährleisten. Des weiteren erlaubt der Dublin Core auch keine Spezifikation der Optionen einer Suchschnittstelle. Im folgenden werden deshalb zusätzliche Anforderungen an einen Metadatensatz für Suchserver identifiziert und vorgestellt. Diese sind im einzelnen:

- Beschreibung von Optionen der Suchschnittstelle.
- Einbinden von Metadaten, die von kooperativen Suchservern zur Verfügung gestellt werden (z. B. Termstatistiken, Zugriffsstatistiken, etc.). Diese werden im folgenden als private Metadaten bezeichnet.
- Einbinden von Metadaten, die im sichtbaren Teil des WWW vorhanden sind (z. B. Linktexte, existierende Kategorisierungen etc.). Diese werden im folgenden als öffentliche Metadaten bezeichnet.
- Berücksichtigung von Metadaten, die eine objektive qualitative Einschätzung von Suchservern erlauben.

4.2.1. Private Metadaten

Unter privaten Metadaten verstehen wir Informationen, die von kooperativen Suchservern zur Verfügung gestellt werden. Diese werden aus Quellen extrahiert, die im allgemeinen nur den Betreibern von Web-Servern zur Verfügung stehen.

So besitzen nur die Betreiber eine vollständige Sicht auf die recherchierbaren Rohdaten, wie z. B. die Volltexte von Dokumenten, die in einem Dateisystem aufliegen oder Datensätze, die in einer internen Datenbank abgespeichert sind. Hieraus lassen sich Suchserver-spezifische Metadaten, wie die Gesamtanzahl der recherchierbaren Dokumente sowie vollständige Termstatistiken generieren, um diese (z. B. über das STARTS-Protokoll) einer hierauf aufsetzenden Metasuchmaschine zur Verfügung zu stellen.

Eine weitere wichtige Quelle, aus der sich private Metadaten extrahieren lassen, stellen die Protokoll-Dateien der Web-Server dar. Hieraus lassen sich Zugriffsstatistiken ableiten, die Rückschlüsse über die Popularität eines Servers oder das allgemeine Benutzerverhalten erlauben. Weiterhin lassen sich aus den Protokolldateien auch häufig gestellte Suchbegriffe auslesen, die als Grundlage für eine indirekte inhaltliche Charakterisierung herangezogen werden können.

4.2.2. Öffentliche Metadaten

Im realen Szenario des Unsichtbaren Webs verhalten sich Suchserver jedoch nicht kooperativ. Deshalb kommt den öffentlichen Metadaten eine umso größere Bedeutung zu. Hierunter werden im folgenden Suchserver-spezifische Metadaten verstanden, die gewonnen werden können, ohne auf interne Daten wie Dateistruktur, Datenbankinhalte oder Protokolldateien zuzugreifen. Vielmehr lassen sich öffentliche Metadaten weitestgehend automatisiert aus dem sichtbaren und öffentlich zugänglichen Teil des WWWs extrahieren, indem dieses systematisch nach bereits existierenden Informationen über einen Suchserver durchforstet wird.

Ausgehend von der URL der Startseite eines Suchservers lassen sich bereits eine Fülle von Metadaten automatisiert extrahieren. Neben dem vorhandenen Volltext kann auch die vorhandene HTML-Auszeichnung ausgewertet werden und so zusätzlich spezifischere Daten wie der Titel des Suchservers, bereits hinterlegte Metadaten oder die Optionen der Suchschnittstelle extrahiert werden.

Darüber hinaus können auch die Informationen aus der Nachbarschaft der Startseite, z. B. über Backlink-Seiten, berücksichtigt werden, um Struktur- und weitere Volltextinformationen zu gewinnen.

Weitere öffentlich zugängliche Quellen stellen spezialisierte Web-Sites dar, wie beispielsweise existierende Web-Kataloge für Web-Ressourcen z. B. Kataloge von spezialisierten Suchservern, wie The Big Hub oder Direct Search, als auch generelle wie der Yahoo-Katalog. Aus diesen lassen sich vorhandene Kategorisierungen automatisiert extrahieren, vorausgesetzt, der Suchserver wurde bereits in den Katalogen erfaßt.

4.2.3. Qualitätsorientierte Metadaten

Eine Information, die bei der Berechnung von Relevanzgewichten von Suchservern zu Suchanfragen von erheblicher Bedeutung ist und die vom DC nicht unterstützt wird, stellt die Größe der über einen Suchserver verfügbaren Datenmenge dar[2]. Ebenso wie die Anzahl der auf einen Suchserver verweisenden Backlinks oder dessen Pagerank-Gewicht, kann die Information der verfügbaren Gesamtdatenmenge als qualitative Information gewertet werden: Je mehr Dokumente über einen Suchserver verfügbar sind, desto höher ist auch die Wahrscheinlichkeit einzuschätzen, daß der Suchserver relevante Dokumente enthält und desto höher ist seine generelle Qualität einzuschätzen. Eine wichtige Anforderung an einen Metadatensatz für Suchserver stellt somit auch die Berücksichtigung von qualitätsorientierten Metadaten dar.

So spielen neben rein inhaltsorientierten Metadaten zur Beschreibung von Suchservern vermehrt auch Metadaten eine immer größere Rolle, die eine qualitätsorientierte Einschätzung von Suchservern erlauben. Web-basierte Suchserver besitzen oft zahlreiche Backlinks von externen Servern, die für qualitative Aussagen herangezogen werden können (z. B. Backlinks auf die Startseite eines Servers oder auf dessen Suchschnittstelle). In Kapitel 2.4.1 wurde bereits diskutiert, inwieweit Hyperlinks dazu geeignet sind, um qualitätsorientierte Aussagen über Web-Ressourcen herzuleiten. Dabei wurden verschiedene qualitative Kriterien – Hubgewicht, Authoritygewicht, Pagerankgewicht und Anzahl der Backlinks – vorgestellt. Diese erlauben eine objektive Einschätzung der Qualität von Web-Ressourcen, denn zumindest die ersten drei genannten Gewichte verhalten sich relativ robust gegenüber Manipulationen, da sie auch die Qualität der Backlink-Seiten mitberücksichtigen. Wird der komplette Web-Graph berücksichtigt, spiegeln sie eine kollektiv erzeugte und einheitlich meßbare Wertschätzung der Web-Communities wider.

Es lassen sich noch weitere qualitätsorientierte Metadaten identifizieren, insbesondere im Hinblick auf Web-basierte Suchserver mit spezialisierten Inhalten (siehe [128] für eine Übersicht). Beispiele sind die Häufigkeit, mit der ein Suchserver aktualisiert wird, die Häufigkeit der Zugriffe, die mittlere Wartezeit für eine Anfrage oder der Grad der thematischen Geschlossenheit der recherchierbaren Dokumente.

Eine wichtige Anforderung an einen Metadatensatz für Suchserver ist es deshalb, qualitative Metadaten mitzuberücksichtigen. Dabei sollen allerdings nur solche Metadaten aufgenommen werden, die tatsächlich auch objektiv meßbar sind und einheitlich determiniert werden können, um hieraus vergleichbare Aussagen über die Suchserver abzuleiten.

[2] Im CORI-Gewicht wird eine derartige Information beispielweise über den cw-Wert mitberücksichtigt, also der Gesamtanzahl verschiedener Terme in einem Suchserver.

Qualitative Metadaten können sowohl private Metadaten (z. B. Anzahl der recherchierbaren Dokumente, Zugriffsstatistiken) als auch öffentliche Metadaten (z. B. Anzahl der Backlinks, Linkterme) sein.

4.2.4. Übersicht über Metadaten

In Tabelle 4.1 sind Beispiele für öffentliche und private Metadaten zusammengestellt. Die aufgeführten Metadaten werden in der Tabelle nach weiteren Kriterien gruppiert:

1. nach der privaten oder öffentlichen Quelle, aus der das spezifizierte Metadatum extrahiert werden kann (Protokolldatei des Web-Servers, benachbarte Seiten etc.)

2. gemäß des Typs des Metadatums (Inhaltsmerkmal oder Qualitäts- / Kontextmerkmal)

	Quelle	Qualitäts-/Kontextmerkmal	Inhaltsmerkmal
privat	Protokoll-Datei	Anzahl Zugriffe	häufige Suchbegriffe
	Rohdaten	Anzahl Dokumente Aktualisierungsfrequenz	Dokumentterme
öffentlich	Start-/Suchseite	Suchoptionen	Titel des Suchservers Terme auf Startseite Kurzbeschreibungen
	Benachbarte Seiten	Pagerankgewicht Hub- Authority Gewicht Anzahl Backlinks	Linkassoziierte Terme Titel benachbarter Seiten
	Spezialseiten (z. B. Kataloge)		Kategorienbezeichner Kurzbeschreibungen

Tabelle 4.1.: Beispiele für Suchserver-beschreibende Metadaten

4.2.5. Untersuchung von existierenden Metadatensätzen für Dokumentkollektionen

In diesem Abschnitt soll untersucht werden, inwieweit die genannten Anforderungen an einen Metadatensatz für Web-basierte Suchserver von zwei Metadatensätzen für spezialisierte Kollektionen erfüllt werden. Untersucht werden hierzu die *Research Support Library Programm (RSLP) Collection Description* und die *STARTS Content Summaries*.

4.2.5.1. RSLP Collection Description

Im Rahmen des britischen Forschungsprogramms Research Support Library Programm (RSLP) wurde ein Metadatensatz für Kollektionen von Dokumenten (RSLP Collection Description, RSLP-CD) [93] entwickelt, um die Kollektionen verschiedener RSLP-Projekte einheitlich in maschinenlesbarer Form zu beschreiben. Die Spezifikation und Implementierung des RSLP-CD Metadatensatzes erfolgt unter Verwendung des RDF-Standards, wodurch die Maschinenlesbarkeit und die Austauschbarkeit der Daten zwischen verschiedenen Sammlungen erleichtert wird.

Die Felder des Dublin Core sowie einige DC-Qualifizierer werden unter Verwendungen des XML-Namensraummechanismus' in eine RSLP-CD Kollektionsbeschreibung miteingebunden. Darüber hinaus werden in einer RSLP-CD spezifischere Felder von den DC-Feldern mittels *sub-property* abgeleitet. Beispielsweise sind zwei verschiedene Felder für Datumsangaben vorgesehen, die beide von *dc:date* abgeleitet werden: ein Erfassungsdatum sowie ein Entstehungsdatum der in der Kollektion enthaltenen Objekte.

Das Metadatenschema soll auch zur Beschreibung von nicht-elektronischen Kollektionen (z. B. Museumsbeständen) verwendet werden können. Die Eingabe der Metadaten erfolgt in der Regel manuell über eine formularbasierte Web-Schnittstelle. Die Forderung nach leichter Pflege und Erzeugbarkeit der Metadaten ist somit nur unzureichend erfüllt. Demzufolge sieht der Metadatensatz keine Felder zur Spezifikation von Suchschnittstellen vor. Eine Integration von Termstatistiken sowie der Gesamtanzahl der enthaltenen Dokumente ist ebenfalls nicht vorgesehen. Das Feld *strength* erlaubt zumindest die Angabe von qualitativen Eigenschaften zu einer Kollektion, indem nach subjektiver Einschätzung die Stärken einer spezifischen Kollektion formuliert werden können.

Der RSLP-CD ist stark an bibliothekarischen Anforderungen orientiert, damit er auch zur Beschreibung von nicht Web-basierten bzw. nicht-elektronischen Kollektionen angewendet werden kann und erscheint zusammenfassend als nicht ausreichend für die Erfordernisse des Verteilten Information Retrievals.

4.2.5.2. STARTS Content Summaries

Die bereits in Abschnitt 3.5.1 erwähnte Content Summary des STARTS- Protokolls (STARTS-CS) [53] erlaubt die Beschreibung von Kollektionen von Textdokumenten durch die Angabe von Server-weiten Termlisten mit den entsprechenden tf- und df-Werten. Dabei werden auch strukturelle Informationen mitberücksichtigt, also ob ein Term im Titel eines Dokumentes oder nur im Volltext vorkommt. Auch die Ge-

samtanzahl der verfügbaren Dokumente kann spezifiziert werden. Darüber hinaus erlaubt die STARTS-CS eine Spezifikation der Suchschnittstelle durch Angabe der unterstützten Suchfelder. Zusätzlich können über die Felder *ScoreRange* und *RankingAlgorithmID* Informationen über die verwendeten Gewichtungsalgorithmen erfaßt werden. Daneben sieht die STARTS-CS einige allgemeine Felder vor, wie verwendete Sprachen, Kontaktadressen oder Kurzebeschreibung des Suchservers. Allerdings ist die STARTS-CS primär zur Beschreibung von kooperativen Suchservern vorgesehen. Felder zur Integration von automatisiert generierten öffentlichen Metadaten sind in der STARTS-CS nicht vorgesehen. Abgesehen von den Informationen über die Gesamtdatenmenge sieht die STARTS-CS auch keine qualitätsorientierten Metadaten vor.

Die STARTS-CS kann als Gegenstück zum RSLP-CD gesehen werden und orientiert sich in erster Linie an den Anforderungen des Verteilten Information Retrievals und weniger an bibliothekarischen Anforderungen. Die Tabelle 4.2 zeigt zusammenfassend, inwieweit die beiden vorgestellten Metadatensätze die genannten Anforderungen erfüllen (+ bedeutet: die Anforderung wird erfüllt, 0 bedeutet: die Anforderunge wird zum Teil erfüllt, - bedeutet: die Anforderung wird gar nicht erfüllt).

Anforderung	RSLP-CD	STARTS-CS
Für Laien verständlich	+	0
Leichte Erzeugung und Pflege	-	+
Verbreitete und akzeptierte Semantik	+	0
Konformität zu existierenden Standards	+	0
Internationale Anwendbarkeit	+	+
Erweiterbarkeit	+	+
Interoperabilität	+	0
Beschreibung der Suchschnittstelle	-	+
Private Metadaten	0	+
Öffentliche Metadaten	-	-
Qualitätsorientierte Metadaten	0	0

Tabelle 4.2.: Vergleich von RSLP-CD und STARTS-CS hinsichtlich der Anforderungen an einen Metadatensatz für Web-basierte Suchserver

4.3. Spezifikation des Frankfurt Core

In diesem Abschnitt erfolgt die Darstellung eines neuen Metadatensatzes zur einheitlichen Beschreibung von Web-basierten Suchservern. Dieser wird in Anlehnung an den Dublin Core als Frankfurt Core (FC) bezeichnet und setzt sich aktuell aus 27 Feldern zusammen.

Die Spezifikation des FC erfolgt unter Verwendung von XML/RDF (siehe Abschnitt

2.5.4.1). Die Benutzung dieses offenen Standards erleichtert die Verwendbarkeit von FC-Metadaten in verschiedenen heterogenen Metasuchsystemen. Auch lassen sich durch die Verwendung des XML-Namensraummechanismus weitere Metadatensätze, wie beispielsweise der Dublin Core, einbinden.

In Anlehnung an die in Abschnitt 4.2.5 durchgeführten Betrachtungen enthält der FC sowohl Felder für öffentliche und private als auch für qualitative Metadaten. Eine vollständige Spezifikation des FC befindet sich im Anhang der Arbeit.

Die Darstellung der Felder des FC erfolgt getrennt nach Kontextmerkmalen und Inhaltsmerkmalen, sowie einzelnen Metadaten zur Auszeichung von Suchoptionen.

Inhaltsbasierte Metadaten

- *fc:title* (sub-property von dc:title)
  Der vom Verfasser, Urheber oder Verleger vergebene Name eines Suchservers.

- *fc:abstract* (sub-property von dc:description)
  Eine textliche Beschreibung des Inhalts eines Suchservers.

- *fc:times* (sub-property von dc:coverage)
  Der Zeitbereich, auf den die einzelnen Dokumente des Suchservers sich inhaltlich beziehen, z. B. eine Epoche oder ein Zeitintervall. Identifizierte Zeitangaben werden in einem RDF-Bag Container abgelegt.

- *fc:places* (sub-property von dc:coverage)
  Die regionale Abdeckung, der die Dokumente des Suchservers inhaltlich zugeordnet werden können, z. B. Orte , Länder. Identifizierte Ortsangaben werden in einem RDF-Bag Container abgelegt.

- *fc:startpage_terms* (sub-property von dc:subject)
  Terme, die aus dem BODY-Element der Startseite extrahiert werden. Stopworte und Tags werden entfernt. Die Terme werden sortiert nach Häufigkeit in einem RDF-Seq Container abgelegt.

- *fc:category_terms* (sub-property von dc:subject)
  Kategorienbezeichner von existierenden Web-Katalogen, unter denen der Suchserver eingeordnet wurde, bzw. Kategorienbezeichner, die durch eine automatisierte Kategorisierung zugeordnet werden konnten. Identifizierte Kategorienbezeichner werden in einem RDF-Bag Container abgelegt.

- *fc:catalog_abstracts* (sub-property von dc:abstract)
  Kurzbeschreibungen von existierenden Web-Katalogen, die den Suchserver be-

schreiben. Die Kurzbeschreibungen werden in einem RDF-Bag Container abgelegt.

- *fc:backlinkpage_terms* (sub-property von dc:subject)
  Terme, die von den Ankertexten evtl. vorhandener Backlinks auf die Startseite extrahiert werden können. Stopworte und Tags werden entfernt. Die Terme werden sortiert nach Häufigkeit in einem RDF-Seq Container abgelegt.

- *fc:backlinkpage_titles* (sub-property von dc:title)
  Dokument-Titel der Backlink-Seiten. Die Titel werden aus dem HTML-TITLE-Tag extrahiert. Die erhaltenen Titel werden in einem RDF-Bag Container abgelegt.

- *fc:query_terms* (sub-property von dc:subject)
  Terme, die am häufigsten über die Suchschnittstelle des Suchservers abgefragt wurden. Über den Zeitraum von einem Jahr werden alle eingegebenen Suchbegriffe aus der Protokolldatei des Web-Servers extrahiert, die mindestens einen Treffer erzeugten. Hieraus werden die am häufigst gestellten Suchterme ausgewählt.[3]. Stopworte werden entfernt und die Terme werden gemäß ihrer absteigenden Vorkommenshäufigkeit in einem RDF-Seq Container abgelegt.

- *fc:title_terms* (sub-property von dc:subject)
  Terme, von den Titeln der Dokumente, die über den Suchserver recherchiert werden können. Stopworte werden entfernt. Die Terme werden gemäß ihres absteigenden df-Wertes sortiert und in einem RDF-Seq Container abgespeichert. (df_i =Anzahl der über den Suchserver recherchierbaren Dokumente, die den Term t_i im Dokument-Titel enthalten)

- *fc:content_terms* (sub-property von dc:subject)
  Inhaltsbeschreibende Terme über die Dokumente, die über den Suchservern recherchiert werden können. Diese können aus evtl. vorhandenen Kurzbeschreibungen oder dem Volltext der Dokumente entnommen werden. Stopworte werden entfernt. Die Terme werden gemäß ihres absteigenden df-Wertes sortiert und in einem RDF-Seq Container abgespeichert (df_i =Anzahl der über den Suchserver recherchierbaren Dokumente, die den Term t_i in Kurzbeschreibungen oder dem Volltext enthalten).

[3] Die Identifikation von Suchbegriffen kann über die Erkennung der Zeichenketten von Suchfeldern und Skripten in den Protokolldateien erfolgen. Da deren Bezeichnung von Server zu Server individuell erfolgen kann, muß eine solche Untersuchung an jede Protokolldatei immer wieder neu angepasst werden. Außerdem lassen sich die Suchbegriffe aus der Protokolldatei nur extrahieren, wenn diese unter Verwendung von HTTP-GET übergeben wurde, also an den URL des Suchskriptes angehängt wurden.

Kontextorientierte/Qualitative Metadaten

- *fc:language* (sub-property von dc:language)
  Die Sprache in der die meisten Dokumente recherchiert werden können. Dies muß nicht zwangsläufig die Sprache sein, in der die Dokumente verfaßt wurden. Auch existieren Multimediadokumenten denen keine Sprache zugeordnet werden kann. Derartige Dokumente sind über vorhandene Metadaten repräsentiert und können über diese recherchiert werden.

- *fc:publisher* (sub-property von dc:publisher)
  Die Einrichtung, die verantwortet, daß der Suchserver in der gebenenen Form zur Verfügung steht, wie z. B. ein Verleger, ein Herausgeber, eine Universität oder ein Unternehmen.

- *fc:format* (sub-property von dc:format)
  Das datentechnische Format, dem die meisten Dokumente angehören, die über den Suchserver recherchiert werden können, z. B. HTML, ASCII, Postscript-Datei, JPEG-Bilddatei etc.

- *fc:date_created* (sub-property von dc:date)
  Zeitpunkt an dem Suchserver erstmals im WWW verfügbar gemacht wurde.

- *fc:date_modified* (sub-property von dc:date)
  Zeitpunkt an dem der Inhalt des Suchserver zuletzt modifiziert wurde.

- *fc:number_of_backlinks*
  Anzahl der Backlinks auf die Startseite des Suchservers, ermittelt über die Backlink-Schlüsselwortsuche einer generellen Suchmaschine.

- *fc:authority_weight*
  Gewicht für die Autorität eines Suchservers unter Berücksichtigung der vollständigen Linkstruktur des Sichtbaren Webs bzw. eines geeignet großen Teilbereiches hieraus, z. B. HITS-Authority-Gewicht oder Pagerankgewicht.

- *fc:number_of_documents*
  Anzahl unterschiedlicher Dokumente, die über die Suchschnittstelle der Suchserver recherchiert werden können.

- *fc:number_of_accesses_per_day*
  Anzahl der einmaligen Zugriffe von unterschiedlichen Rechneradressen an einem Tag. Dieser Wert soll für mindestens 30 aufeinanderfolgende Tage ermittelt (also etwa dem Zeitraum eines Monats) und hieraus ein Mittelwert der Zugriffe pro Tag ermittelt werden.

Metadaten zur Beschreibung der Suchoptionen

- *fc:search_interface* (sub-property von dc:relation)
  Adresse der Suchschnittstelle des Suchservers (Oft befinden sich Suchschnittstelle und Sartseite einzelner Suchserver auf unterschiedlichen Seiten)

- *fc:search_fields*
  Bezeichner von Suchfelder über die der Gültigkeitsbereich einzelner Suchbegriffe eingeschränkt werden kann (z. B. *Autor*, *Titel*, etc.). Die Suchfelder werden in einem RDF-Bag Container abgelegt.

- *fc:search_operators*
  Operatoren über die zwei Suchbegriffe logisch miteinander verknüpft werden können z. B. AND, OR, AND NOT, NEAR (siehe Abschnitt 3.4.2). Die Operatoren werden in einem RDF-Bag Container abgelegt.

- *fc:search_qualifiers*
  Qualifizierer über die der Interpretationsspielraum von einzelnen Suchbegriffen verändert werden kann, z. B. Rechts-/Linkstrunkierung, Verknüpfung von Termen zu Wortketten, Berücksichtigung von Groß- bzw. Kleinschreibweise, etc. Die Qualifizierer werden in einem RDF-Bag Container abgelegt.

- *fc:search_field_mapping*
  Namen der Suchfelder, die von der Suchschnittstelle unterstützt werden, gefolgt von einem „:“-Zeichen und den Namen von der semantisch äquivalenten Suchkonzepten einer generellen Konzept-Taxonomie, auf die die Suchfelder abgebildet werden (siehe Abschnitt 4.5 für Einzelheiten über die Konzept-Taxonomie). Die Taxonomie kann über das Feld *fc:concept_descriptions* eingebunden werden.

- *fc:concept_descriptions*
  Spezifikationen von Konzeptbeschreibungen, zur zusätzlichen Integration von wissens- bzw. ontologiebasierten Darstellungen. Die Konzeptbeschreibungen werden in einem RDF-Bag Container abgelegt.

4.3.1. Ein Beispiel für den Frankfurt Core

In diesem Abschnitt soll der Frankfurt Core anhand eines vollständig ausgezeichneten Suchservers illustriert werden. Hierzu stand ein spezialisierter Suchserver für historische Schriften zur Verfügung, der an der Professur ABVS[4] der Johann Wolfgang

[4] Architektur und Betrieb Verteilter Systeme

Goethe-Universität in Frankfurt im Rahmen eines von der Deutschen Forschungsgemeinschaft geförderten Digitale Bibliotheken-Projektes [59] entwickelt wurde.

Der Server mit dem Titel „1848 - Flugschriften im Netz“ [5] [96] erscheint zur Illustrierung des FC als geeigent, da zum einen die Protokoll-Datei des Web-Servers sowie die Rohdaten verfügbar sind und zum anderen der Suchserver eine ausreichende Anzahl von Backlinks besitzt (meist von Web-Seiten mit historischen Inhalten).

Über den Suchserver lassen sich Bilddateien digitalisierter Originale von Flugschriften, Zeitungsartikeln, Poster etc. der deutschen Revolution von 1848/1849 recherchieren. Die einzelnen Original-Dokumente der Sammlung werden systematisch von einem Historiker gesichtet und inhaltlich erschlossen. Dieser versieht die Dokumente mit Metadaten wie *Autor*, *Angesprochene Personen*, *Angesprochene Orte*, *Titel* oder *Kurzbeschreibung*. Insgesamt besteht die Sammlung aus etwa 80.000 digitalisierten Dokumenten, von denen derzeit etwa 1000 Dokumente inhaltlich erschlossen vorliegen. Über die Suchschnittstelle kann nur auf die inhaltlich erschlossenen Dokumente zugegriffen werden.

Private und öffentliche Metadaten wurden für den Suchserver weitestgehend automatisiert unter Auswertung der Protokolldateien und der Rohdaten sowie der Linkstruktur des Sichtbaren Webs generiert [118]. Strategien zur Gewinnung von öffentlichen Metadaten werden detailliert in Abschnitt 4.4.1 besprochen.

Im aufgezeigten Beispiel werden nur die Anzahl der Backlinks (FC-Feld *number_of_backlinks*) aufgeführt. Dies liegt daran, daß zuverlässige und vergleichbare Pagerank-, Hub- und Authoritygewichte nur unter Berücksichtigung des gesamten Web-Graphen berechnet werden und deshalb nur mit extrem großen Aufwand erstellt werden können. Außerdem ist in diesem Zusammenhang auch auf den Artikel von Amento et al zu verweisen [4]. Dieser zeigt anhand von Experimenten auf, daß die Anzahl der Backlinks ebenso gut die Qualität von Web-Sites vorhersagt, wie Hub-/ Authority- und Pagerankgewichte. Dabei wurden für die Experimente menschliche Experten und Laien aufgefordert, die Qualität der Web-Sites zu beurteilen, um vergleichende Aussagen ableiten zu können.

Im folgenden wird die FC-Auszeichnung des Suchservers „1848 - Flugschriften im Netz“ in RDF dargestellt. Anhand des Beispiels wird auch die Einbindung von Dublin Core Metadaten demonstriert. So wird mit dem Feld *dc:subject* das Sachgebiet gemäß dem DDC Klassifikationsschemata identifziert (hier: *History of Germany and Austria*). Die Identifikation von Kodierungsschemata erfolgt unter Verwendung von Dublin Core Qualifizierern (Namensraum *dcq*).

[5] 1848 Flugschriften-Server: http://dbib.uni-frankfurt.de/1848 [14. Nov. 2001]

```
<?xml version="1.0"?>
<rdf:RDF
 xmlns:rdf="http://www.w3.org/TR/PR-rdf-syntax/"
 xmlns:fc="http://dbib.uni-frankfurt.de/fc/schema/"
 xmlns:dc="http://purl.org/dc/elements/1.1/"
 xmlns:dcq="http://purl.org/dc/qualifiers/1.0/">

    <rdf:Description about="http://dbib.uni-frankfurt.de/1848">

                <fc:title xml:lang="de">1848 - Flugschriften im Netz</fc:title>

                <fc:abstract xml:lang="de">
                    In dem Projekt 1848 - Flugschriften im Netz wurden
                    historische Flugblaetter, Aufrufe und Plakate aus
                    der Revolution von 1848 im WWW zugaenglich gemacht.
                    Diese wurden zuvor sicherheitsverfilmt und
                    anschliessend digitalisiert.
                </fc:abstract>

                <dc:subject>
                  <rdf:Description>
                    <dcq:scheme>
                        DDC
                    </dcq:scheme>
                    <rdf:value>
                        943 History of Germany and Austria
                    </rdf:value>
                  </rdf:Description>
                </dc:subject>

                <fc:format>
                  <rdf:Description>
                    <dcq:formatScheme>IMT</dcq:formatScheme>
                        <rdf:value>img/gif</rdf:value>
                  </rdf:Description>
                </fc:format>

                <fc:language>
                  <rdf:Description>
                    <dcq:languageScheme>RFC1766</dcq:languageScheme>
                    <rdf:value>de</rdf:value>
                  </rdf:Description>
                </fc:language>

                <fc:times>
                  <rdf:Bag>
                        <rdf:li>1848 - 1849</rdf:li>
```

```
    </rdf:Bag>
</fc:times>

<fc:places>
    <rdf:Bag>
        <rdf:li> Berlin </rdf:li>
        <rdf:li> Wien </rdf:li>
        <rdf:li> Frankfurt </rdf:li>
    </rdf:Bag>
</fc:places>

<fc:date_created>
  <rdf:Description>
      <dcq:dateScheme> WTN8601 </dcq:dateScheme>
      <rdf:value> 1998-10-01 </rdf:value>
  </rdf:Description>
</fc:date_created>

<fc:date_modified>
  <rdf:Description>
      <dcq:dateScheme> WTN8601 </dcq:dateScheme>
      <rdf:value> 1999-05-18 </rdf:value>
  </rdf:Description>
</fc:date_modified>

<fc:publisher>
      Johann Wolfgang Goethe-Universitaet Frankfurt
      Graefstr. 38 D-60486 Frankfurt am Main
</fc:publisher>

<fc:backlink_terms>
      <rdf:Seq>
          <rdf:li xml:lang="en"> 20 history </rdf:li>
          <rdf:li xml:lang="en"> 18 bibliography </rdf:li>
          <rdf:li xml:lang="en"> 18 germany </rdf:li>
          <rdf:li xml:lang="en"> 17 university </rdf:li>
          ..............
      </rdf:Seq>
</fc:backlink_terms>

<fc:category_terms>
      <rdf:Bag>
          <rdf:li xml:lang="de"> Deutsche Geschichte </rdf:li>
          <rdf:li xml:lang="de"> Geisteswissenschaften </rdf:li>
          <rdf:li xml:lang="de"> Geschichte </rdf:li>
          <rdf:li xml:lang="de"> 19. Jahrhundert </rdf:li>
      </rdf:Bag>
```

```
</fc:category_terms>

<fc:catalog_abstracts>
        <rdf:Bag>
                <rdf:li xml:lang="de">
                    1848 Flugschriften im Netz – praesentiert
                    historische Flugblaetter, Aufrufe und
                    Plakate aus der Revolution von 1848.
                </rdf:li>
        </rdf:Bag>
</fc:catalog_abstracts>

<fc:backlinkpage_titles>
        <rdf:Bag>
                <rdf:li xml:lang="de">
                  UB Marburg – Fachinformation Geschichte
                </rdf:li>
                <rdf:li xml:lang="de">
                  Fachinformation Geschichte – Revolution
                  1848/49
                </rdf:li xml:lang="de">
                <rdf:li>
                  DVB, Geschichte
                </rdf:li>
                <rdf:li xml:lang="en">
                  German Studies Web: Library Catalogs and
                  Other Library Information
                </rdf:li>
                ..............
        </rdf:Bag>
</fc:backlinkpage_titles>

<fc:number_of_documents>
        1000
</fc:number_of_documents>

<fc:number_of_accesses_per_day>
        86
</fc:number_of_accesses_per_day>

<fc:number_of_backlinks>
        82
</fc:number_of_backlinks>

<fc:query_terms>
        <rdf:Seq>
                <rdf:li> 92 gagern </rdf:li>
```

```
                <rdf:li> 77 berlin </rdf:li>
                <rdf:li> 60 friedrich </rdf:li>
                <rdf:li> 57 revolution </rdf:li>
                <rdf:li> 51 blum </rdf:li>
                ..............
        </rdf:Seq>
</fc:query_terms>

<fc:document_title_terms>
        <rdf:Bag>
                <rdf:li> 273 reich </rdf:li>
                <rdf:li> 230 nationalgarde </rdf:li>
                <rdf:li> 284 wien </rdf:li>
                <rdf:li> 276 buerger </rdf:li>
                <rdf:li> 176 wahl </rdf:li>
                ..............
        </rdf:Bag>
</fc:document_title_terms>

<fc:content_terms>
        <rdf:Seq>
                <rdf:li> 512 wien </rdf:li>
                <rdf:li> 118 berlin </rdf:li>
                <rdf:li> 99 preussische </rdf:li>
                ..............
        </rdf:Seq>
</fc:content_terms>

<fc:search_interface  rdf:resource=
        "http://dbib.uni-frankfurt.de/1848/suchen.html"/>

<fc:search_fields>
        <rdf:Bag>
                <rdf:li> Person </rdf:li>
                <rdf:li> Stichwort </rdf:li>
        </rdf:Bag>
</fc:search_fields>

<fc:search_field_mapping>
        <rdf:Bag>
                <rdf:li> Person: person </rdf:li>
                <rdf:li> Stichwort: keyword </rdf:li>
        </rdf:Bag>
</fc:search_field_mapping>

<fc:concept_descriptions>
        <rdf:Bag>
```

```
                    <rdf:li resource=
                        "http://dbib.uni-frankfurt.de/knowledge.lisp"/>
                </rdf:Bag>
            </fc:concept_descriptions>

    </rdf:Description>

</rdf:RDF>
```

4.4. Automatisierte Erzeugung von Frankfurt Core Metadaten

4.4.1. Ein Metadatenkollektor für öffentliche Metadaten

Die Generierung von Metadaten für einige Felder des Frankfurt Core kann durch Bereitstellung entsprechender Werkzeuge automatisiert werden. Hierzu wurde ein Metadaten-Kollektor entwickelt, der systematisch Backlink-Seiten, Frontseiten und Web-Kataloge analysiert, um öffentliche Metadaten zu erzeugen. Hierin werden spezialisierte Strategien angewendet, die ausgehend von der URL der Startseite eines Suchservers Informationen über diesen automatisiert aus dem Sichtbaren Web extrahieren. Die gewonnenen Metadaten werden in den entsprechenden Feldern des FC abgelegt. Die Identifizierung von Backlink-Seiten zu einem spezifizierten Suchserver erfolgt unter Ausnutzung einer generellen Suchmaschine und der in Abschnitt 2.4.1 beschriebenen *link*-Schlüsselwortsuche.

Im folgenden werden die Vorgehensweisen zur automatisierten Erzeugung von Backlink-Termen (FC-Feld: *backlink_terms*), von Kategorienbezeichnern und Kurzbeschreibungen aus existierenden Web-Katalogen (FC-Feld *category_terms* und *catalog_abstracts*) dargestellt. Dabei sei ein Suchserver ausschließlich über den URL u seiner Startseite identifziert.

Terme, die Hyperlinks von Backlink-Seiten unterlegt sind (FC-Feld: *backlink_terms*):

1. Anfrage an eine generelle Suchmaschine (*link:<u>*) zur Identifikation von Backlink-Seiten.

2. Erzeugen einer Menge B von Backlink-Seiten mit $B = \{b_1, \ldots, b_m\}$, wobei b_1 bis b_m die URLs der ersten m Backlink-Seiten sind, die aus der Resultatseite extrahiert werden können.

3. Für alle $b_i \in B$: Auslesen der Terme t_j auf der Seite b_i , die sich zwischen den Anker-Tags <A> und </A> des Hyperlinks auf u befinden.

4. Entfernen von Stopworten.

5. Sortierung der verbliebenen Terme nach Vorkommenshäufigkeit und Eintragen in *fc:backlink_terms*.

Terme und Kurzbeschreibungen, die von Katalogbetreibern zur Beschreibung der Kollektion verwendet wurden (FC-Feld: *category_terms* und *catalog_abstracts*):

1. Gegeben ist eine Menge $K = \{k_1, \ldots, k_n\}$ von URLs von Web-Servern, wobei jede k_i den Start-URL eines Web-Katalogs (z. B. Yahoo, Brightplanet etc.) spezifiziert.

2. Für alle $k_i \in K$: Anfrage an eine generelle Suchmaschine mit der Anfrage: *+link:*$< u >$ *+url:*$< k_i >$. Die Schlüsselwortsuche url schränkt die Menge der Backlink-Seiten auf jene ein, deren URL den Präfix k_i aufweist.

3. Die Anfrage liefert als Antwort Seiten zurück, auf denen der Suchserver u bereits kategorisiert vorliegt. Die von den verschiedenen Betreibern verwendeten Kategorienterme können je nach Katalog aus den Snippets oder dem Lokalteil der URLs der Treffer-Seiten ausgelesen[6] und in das FC-Feld *category_terms* eingetragen werden.

4. Auf den hierdurch gefundenen Backlink-Seiten finden sich meist manuell erstellte Kurzbeschreibungen der Suchserver, die von Katalogbetreibern erstellt wurden. Falls vorhanden werden diese über entsprechende Wrapperprogramme von den Originalseiten extrahiert und in das FC-Feld *catalog_abstracts* eingetragen.

4.4.2. Strategien zur Generierung von privaten Metadaten für nicht-kooperative Suchserver

Auch für einige private Metadaten lassen sich Strategien identifzieren, um diese weitestgehend automatisiert zu erzeugen, auch wenn die Rohdaten und Protokoll-Dateien nicht vorliegen.

[6] Meist werden die enthaltenen Unterverzeichnisse der Kataloge nach den verwendeten Kategoriebezeichnern benannt.

Manchmal befindet sich die Angabe der Gesamtanzahl der recherchierbaren Dokumente (*fc:number_of_documents*) bereits auf den statischen Seiten der Suchserver. Jedoch für einige Suchserver läßt sich diese Anzahl auch durch gezieltes Anfragen an ihren Suchschnittstellen ermitteln, z. B. durch alleinige Angabe eines Wildcard-Qualifizieres (meist das *-Zeichen) als Suchanfrage – dies liefert idealerweise alle Dokumente zurück, und die Anzahl der Treffer identifiziert die gesuchte Gesamtanzahl. Eine weitere Möglichkeit, eine Resultatliste über alle Dokumente zu erzeugen, besteht in der Formulierung einer Suchanfrage der Form NOT, gefolgt von einer zufälligen Zeichenkette, die wahrscheinlich nicht in einem Dokument vorkommt (z. B. NOT *hjjhfdajh*).

Die privaten Metadaten *document_title_terms*, *content_terms* und *query_terms* eignen sich des weiteren dazu, um die Resultate von Probeanfragen zu hinterlegen. Im Feld *query_terms* werden dann jene Probeanfragen, die Treffer erzeugt haben, hinterlegt – geordnet nach der Größe der jeweils erzeugten Treffermenge. Aus den Dokumenttiteln und den Snippets der verschiedenen Resultatseiten lassen sich die Term-Listen für *document_title_terms* und *content_terms* erzeugen.

4.5. Eine wissensbasierte Repräsentation von Suchfeldern

Der Frankfurt Core unterstützt eine einheitliche Verwendung verschiedener Suchfeldern durch Abbildung der individuellen Suchfelder auf allgemeingültige Suchkonzepte. Hierdurch wird die Zusammenfassung verschiedener Suchfelder auf der integrierten Suchschnittstelle einer Metasuchmaschine ermöglicht.

4.5.1. Ansätze zur Integration von Suchfeldern

Zur Integration von Suchfeldern verschiedener Suchmaschinen in einer Metasuchmaschine lassen sich in einer ersten Näherung zwei gegensätzliche Strategien identifizieren. Bei den folgenden Betrachtungen wird davon ausgegangen, daß syntaktisch gleichbedeutende Suchfelder bereits auf gemeinsame Bezeichner abgebildet wurden.

1. Vereinigung der Suchfelder: Alle unterschiedlichen Suchfelder der zu integrierenden Suchserver werden in die Suchoberfläche der Metasuchmaschine übernommen.

2. Schnittmenge der Suchfelder: Nur die Suchfelder, die allen zu integrierenden Suchservern gemeinsam sind, werden übernommen.

Bei Anwendung der ersten Strategie besteht das Problem, wie von den Wrappern eine Anfrage umgesetzt werden soll, wenn der Suchserver das ausgewählte Suchfeld nicht unterstützt. Triviale Lösungsstrategien, wie das Ignorieren einer solchen Anfrage auf Seiten des Wrappers oder die Umsetzung der Anfrage auf alle unterstützten Suchfelder, sind unpraktikabel. Deshalb verfolgen die meisten Metasuchmaschinen den zweiten Ansatz, d. h. die Bildung einer Schnittmenge aller gemeinsamen Suchfelder („Least Common Denominator"-Strategie). Dieser „kleinste gemeinsame Nenner" besteht im Falle der Suchfelder jedoch meist nur aus einer generellen Suche, die nicht näher eingeschränkt werden kann. Eine solche Anfrage wird meist auf alle Felder der zugrundeliegenden Suchserver umgesetzt, wodurch deren individuelle Suchmöglichkeiten verloren gehen.

Die Erhaltung der Suchserver-spezifischen Felder zur Anfrageformulierung stellt eine wichtige Anforderung an den Entwurf von Metasuchmaschinen für spezialisierte Suchserver dar [65].

4.5.2. Abbildung der Suchfelder auf einheitliche Konzepte

Um die verschiedenen Suchfelder in einer gemeinsamen Suchschnittstelle zu integrieren, müssen die syntaktischen Unterschiede der individuellen Suchfelder überwunden werden.

So beziehen sich viele Suchfelder auf identische oder zumindest semantisch ähnliche Suchkonzepte, weisen allerdings unterschiedliche Bezeichner auf. Beispielsweise können die Suchfelder mit den Bezeichnungen *Autor*, *Author*, *author*, *Writer* oft synonym verwendet werden und beziehen sich auf ein identisches Konzept. Der erste Schritt zur Vereinheitlichung von Suchfeldern besteht somit darin, für semantisch gleichbedeutende Suchfelder einen systemweit eindeutigen Bezeichner für das referenzierte Suchkonzept zu verwenden.

Im folgenden bezeichnet $K = \{k_1, \ldots, k_n\}$ die Menge aller Suchkonzepte, die von der integrierten Suchschnittstelle einer Metasuchmaschine unterstützt werden. Um einen Suchserver s, dessen Suchschnittstelle die Menge $F = \{f_1, \ldots, f_m\}$ von Suchfeldern unterstützt, in die Metasuchmaschine zu integrieren, muß jedes einzelne Suchfeld $f_i \in F$ auf jenes $k_j \in K$ abgebildet werden, das der Bedeutung von f_i am nächsten kommt.

Der Frankfurt Core sieht hierzu ein Feld *searchfield_mapping* vor. Hierin werden die von individuellen Suchservern unterstützten Suchfelder auf einheitliche Bezeichner von Suchkonzepten abgebildet. Für das Beispiel des 1848-Suchservers werden durch die Anweisung

```
<fc:searchfield_mapping>
   <rdf:Bag>
      <rdf:li> Person:person </rdf:li>
      <rdf:li> Stichwort:keyword </rdf:li>
   </rdf:Bag>
</fc:searchfield_mapping>
```

die beiden unterstützten Suchfelder (*Person*, *Stichwort*) auf die entsprechenden Konzeptbezeichner für personenbezogene (*person*) und stichwortbezogene (*keyword*) Suche abgebildet. Über das Feld *concept_descriptions* lassen sich die verwendeten Konzeptspezifikationen in eine Frankfurt Core Beschreibung einbinden.

4.5.3. Eine Taxonomie für Suchfelder

Die Bedeutungen der verschiedenen Suchkonzepte $k_i \in K$ sind nicht scharf voneinander getrennt, und es lassen sich Beziehungen zwischen einzelnen Elementen identifzieren, die für eine Strukturierung der Menge K herangezogen werden können. So unterscheiden sich individuelle Suchkonzepte oft in ihrem Grad an Generalität, wodurch sich eine hierarchische Organisationsstruktur anbietet.

Ein typisches Suchkonzept stellt beispielsweise das Konzept *Creator* dar. Im Sinne des Dublin Cores bezeichnet dies „die Person(en) oder Organisation(en), die den intellektuellen Inhalt einer Ressource verantworten“. Ein Creator (*creator*) kann somit sowohl den Autor eines Buches (*author*), einen Komponisten (*composer*) oder auch einen Photographen identifizieren. Dieser Zusammenhang kann leicht durch eine hierarchische Beziehung im Sinne einer Spezialisierung modelliert werden.

Eine einfache Taxonomie für Suchkonzepte könnte demnach aussehen wie in Abbildung 4.1 dargestellt: Am Anfang der Hierarchie steht das generellste Konzept *thing*. Von diesem sind alle spezifischeren Konzepte abgeleitet. Diese sind grob nach abstrakten (*abstract*) und greifbaren Konzepten (*tangible*) aufgeteilt.

Im folgenden wird ein Beispiel für das Abbilden der Suchfelder einer konkreten Web-Kollektion auf die Suchkonzepte der Taxonomie dargestellt. Als Beispiel fungiert die Internet Movie Database (IMDB)[7]. Diese stellt u. a. zwei verschiedene Felder bereit, die sich auf Individuen beziehen. So bezieht sich das Suchfeld *People* auf Personen, die in die Herstellung eines Filmes involviert waren oder sind – dies können sowohl Regiesseure, Schauspieler, Kameraleute etc. sein. Dementsprechend wird *People* auf das Konzept *person* abgebildet, weil dies das spezifischste Feld in der Taxonomie dar-

[7] IMDB: http://www.imdb.com [14. Nov. 2001]

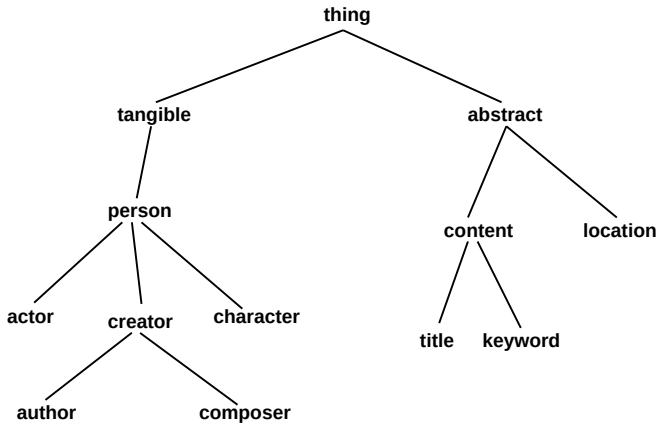

Abbildung 4.1.: Ausschnitt aus einer Taxonomie für Suchkonzepte

stellt, das alle Bedeutungen des *People*-Suchfeldes abdeckt[8]. Das Suchfeld *Character* hingegen dient der Suche nach Individuen, die in einem Film porträtiert werden, bezeichnet also reale als auch nicht reale Individuen wie *Richard III*, *Luke Skywalker*, *Robin Hood* etc. und wird auf das eigens hiefür vorgesehene Suchkonzept *character* abgebildet.

4.5.4. Verwendung der Taxonomie in einer Metasuchmaschine

Die Taxonomie kann dazu genutzt werden, um anhand eines ausgewählten Suchkonzeptes von Seiten einer Metasuchmaschine jene Konzepte zu identifizieren, die gleich spezifisch oder spezifischer als das ausgewählte sind. Hieraus können Anfragen an die individuellen Suchschnittstellen der angeschlossenen Suchserver generiert werden, ohne die Aussagekraft der kollektionsspezifischen Suchfelder einzuschränken.

Eine Anfrage in der Metasuchmaschine gliedert sich in die folgenden Teilschritte auf:

1. Auswahl der Suchkonzepte: Über die Suchschnittstelle der Metasuchmaschine können Anfragen formuliert werden, wobei alle in der Taxonomie enthaltenen Konzepte als Suchfelder zur Einschränkung des Gültigkeitsbereiches einer Anfrage herangezogen werden können. Wird kein Suchkozept spezifiziert, so wird automatisch das allgemeinste Suchkonzept *thing* zugewiesen.

2. Auswahl der abzufragenden Suchfelder: Unterstützt die Suchschnittstelle eines integrierten Suchservers eines oder mehrere der identifizierten Suchkonzepte, kann die Metasuchmaschine entsprechende Abfragen gemäß der in der FC-Beschreibung spezifizierten Abbildung vorbereiten. Es werden nur jene Felder

[8] Man beachte, daß z. B. ein Schauspieler bzw. ein Kameramann keinen Creator im Sinne des Dublin Core Feldes darstellen.

für eine Anfrage ausgewählt, die gemäß der Taxonomie gleich oder spezifischer als die ausgewählten Felder sind.

3. Anfrageversendung: Die erzeugten Anfragen werden an die Suchschnittstellen der Kollektionen versendet.

Im folgenden wird ein ausführliches Beispiel für den Ablauf einer Anfrage dargestellt. Über die Schnittstelle einer Metasuchmaschine können Anfragen spezifiziert werden, wobei jedes der Konzepte der in Abbildung 4.1 dargestellten Taxonomie zur Anfrageformulierung herangezogen werden kann.

Die Verwendung der Taxonomie in einer Metasuchmaschine soll am Beispiel von drei heterogenen Suchservern demonstriert werden und ist in Abbildung 4.2 dargestellt. Für jeden der eingebundenen Suchserver ist zusätzlich die im FC spezifizierte Abbildungsvorschrift der Suchfelder auf die Konzepte der Taxonomie dargestellt. Im einzelnen werden betrachtet:

1. Ein Bibliotheks-OPAC:
   Der (deutschsprachige) OPAC besitzt Suchfelder für Autorensuche und Titelsuche. Die Suchfeldbezeichner lauten entsprechend *Autor* und *Titel* und werden auf die Taxonomie-Konzepte *author* und *title* abgebildet.

2. Eine Film-Datenbank:
   Die Film-Datenbank sieht Suchfelder für die Suche nach dargestellten Personen (Feldname: *Character*, Suchkonzept:*character*) und nach Schauspielern (Feldname:*Actor*, Suchkonzept:*actor*) vor.

3. Eine generelle Suchmaschine:
   Die generelle Suchmaschine sieht keine Suchfelder zur Anfrageeinschränkung vor. In diesem Falle wird im FC nur ein einziger Feldbezeichner (genannt *Query*) spezifiziert, der auf das allgemeinste Suchkonzept *thing* abgebildet wird.

Im Beispiel wird nun über die Suchoberfläche der Metasuchmaschine eine Anfrage *person:"spielberg"* formuliert. Für jeden der angeschlossenen Suchserver wird nun gemäß der im FC hinterlegten Abbildungsvorschrift überprüft, ob die unterstützten Suchkonzepte gemäß der Taxonomie spezifischer als oder gleich dem *person*-Konzept sind.

Das Suchfeld *Autor* vom OPAC wird auf das Konzept *author* abgebildet. Dies stellt ein gültiges Subkonzept von *person* dar, *title* jedoch ist kein Subkonzept von *person*, deshalb wird nur die Anfrage *Autor:"spielberg"* an den OPAC versendet.

Für die Film-Datenbank gilt, daß die unterstützten Suchkonzepte *actor* und *character* ebenfalls gültige Subkonzepte von *person* sind, *title* jedoch nicht. Dementsprechend werden zwei Anfragen an die Film-Datenbank versendet – *Character:"spielberg"* und *Actor:"spielberg"*.

An die generelle Suchmaschine dahingegen wird keine Suchanfrage versendet, denn eine personenbezogene Suche wird von dessen Suchschnittstelle nicht unterstützt.

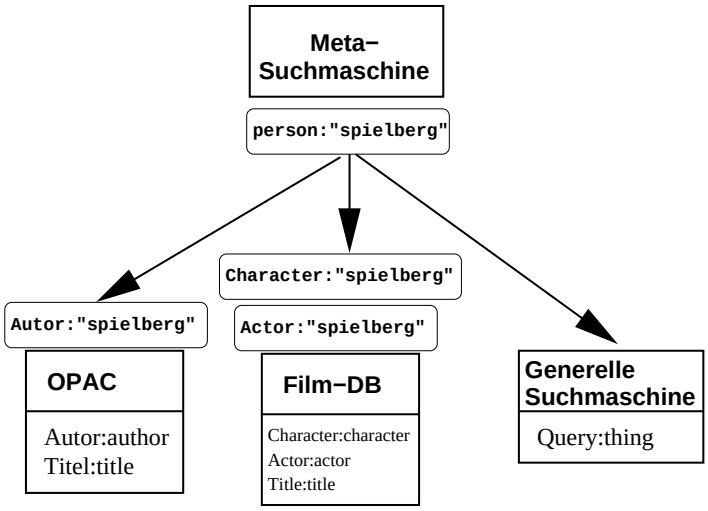

Abbildung 4.2.: Beispiel für die Verwendung der Konzeptaxonomie

Es zeigt sich, daß durch die dargestellte Vorgehensweise das Versenden von überflüssigen Anfragen reduziert und somit die Server der eingebundenen Suchmaschinen entlastet werden kann.

4.5.5. Wissensbasierte Spezifikation von Suchkonzepten mit Description Logic

Die semantisch einheitlich interpretierbare Repräsention von Wissen bildet eine der wesentlichen Grundvoraussetzungen für die Realisierung des von Tim Berners-Lee skizzierten Semantischen Webs (siehe Abschnitt 2.5.4.2).

Entsprechendes gilt für die Realisierung der Suchkonzept-Taxonomie. Für diese muß die eindeutige Interpretierbarkeit der einzelnen Suchkonzepte gewährleistet werden, um eine exakte Abbildung von Suchfeldern auf Suchkonzepte vorzunehmen zu können und so eine mißverständliche Interpretation von Suchkonzepten auszuschließen.

Um die Wiederverwendbarkeit einer solchen Taxonomie und damit die Langlebigkeit von hierauf basierenden Anwendungen zu gewährleisten, erscheint es notwendig, eine Spezifikationstechnik zu wählen, die eine formale Beschreibung von Suchkonzepten

erlaubt. Dies ermöglicht es einzelne Suchkonzepte über komplexere Beziehungen einander in Bezug zu setzen, als es eine einfache Enthaltensein-Relation zuläßt.

Um diese Anforderungen zu berücksichtigen, muß eine Beschreibungstechnik gewählt werden, die den Aufbau einer solchen ontologiebasierten Wissensrepräsentation zuläßt. Deshalb erfolgt die Spezifikation der Taxonomie unter Verwendung von Description Logic.

Description Logic (DL) [121] ist eine leistungsfähige Klasse von Sprachen zur Wissensrepräsentation in der Tradition von semantischen Netzen und objektorientierten Repräsentationen. DLs erlauben die formale und strukturierte Spezifikation von Wissen in einem Anwendungsgebiet, wobei DLs im besonderen zur Modellierung von Konzepten und Konzepthierarchien geeignet sind. Der DL-Formalismus unterscheidet dabei zwischen zwei fundamentalen Bausteinen, aus denen sich komplexe Konzeptbeschreibungen sukzessive zusammensetzen lassen: Konzepte und Rollen. Ein Konzept ist eine Beschreibung, die gemeinsame Eigenschaften von eigenständigen Instanzen (Objekten) zusammenfaßt. Rollen dahingegen beschreiben die Eigenschaften von Instanzen, die einer bestimmten Klasse angehören. Konzepte können als unäre Prädikate und Rollen als binäre Relationen zwischen Objekten aufgefaßt werden. Jede DL definiert eine bestimmte Menge von Sprachkonstrukten (z. B. Schnittmenge, Vereinigung, Rollen-Quantifizierung) über die neue Konzepte und Rollen definiert werden können.

DLs sind expressive Sprachen und garantieren die Terminierung von Schließalgorithmen mit einer korrekten Antwort, wobei auch Objekte, über die nur unvollständig konzeptionelle Informationen spezifiziert wurden, adäquat behandelt werden können. Dabei werden automatisiert aus den gegebenen spezifizierten Rollen und Konzepten Klassifikations-Taxonomien abgeleitet.

Diese Eigenschaften führten insbesondere in den letzten Jahren zu einem gesteigerten Interesse an DLs. So erscheinen sie als geeignete Kandidaten zur Repräsentation einer globalen Wissensbasis gemäß der Vorstellung des Semantischen Webs. Zum Beispiel verwendet die auf RDF/RDFS basierende Ontologiebeschreibungssprache OIL (Ontology Inference Layer) [24] Description Logic, um sowohl formale Semantik als auch effizientes Schließen zu gewährleisten.

Die implementierungstechnische Realisierung der in dieser Arbeit verwendeten Suchkonzept-Taxonomie in Description Logic erfolgte unter Verwendung des auf LISP basierenden *CLASSIC-Systems* [97]. Eine Beschreibung des CLASSIC-Systems und der hiermit realisierten Konzept-Taxonomie befindet sich im Implementierungsteil dieser Arbeit (Abschnitt 7.4).

5. Hyperlinkbasierte Verfahren für Clustering und Kategorisierung von Suchservern

Zur Ergänzung der inhaltlichen Auszeichnung von Suchservern werden in diesem Kapitel zwei neue Web-Mining Verfahren für Suchserver vorgestellt und anhand von Experimenten evaluiert. Die Verfahren beruhen auf der strukturellen und inhaltlichen Analyse von Hyperlinks. In Abschnitt 5.1 wird ein auf dem HITS- (Hyperlink Induced Topic Search) Algorithmus basierendes Clustering Verfahren behandelt, wohingegen Abschnitt 5.2 sich mit einem hyperlinkbasiertem Kategorisierungsverfahren befaßt.

5.1. HITS-basiertes Clustering von Suchservern

5.1.1. Der HITS-Algorithmus

Der HITS-Algorithmus (Hyperlink Induced Topic Search) erlaubt die Berechnung von Hub- und Authority-Gewichten für Web-Ressourcen (siehe Abschnitt 2.4.1.1) und wurde von Jon Kleinberg während eines Aufenthalts am IBM Almaden Research Lab entwickelt [69]. Der Algorithmus wurde im Rahmen des CLEVER-Projektes[1] weiterentwickelt.

Ausgehend von einer Suchanfrage werden von HITS die besten Hubs und Authorities zurückgeliefert. Für jede Web-Seite, die für eine Anfrage berücksichtigt wird, berechnet der HITS-Algorithmus ein Hub- und ein Authority-Gewicht. Je höher ein Gewicht ist, desto ausgeprägter ist die Hub- bzw. Authority-Eigenschaft einer Seite für die Anfrage. Der HITS-Algorithmus analysiert den Verlinkungsgrad einzelner Seiten in einem ausreichend großen Teilausschnitt des Web-Graphen. Der berücksichtigte Graph-Ausschnitt enthält Seiten, die in einem thematischen Bezug zur initialen Suchanfrage stehen. Um den Teilgraph aufzubauen, werden die folgenden Schritte angewendet:

[1] CLEVER-Projekt: http://www.almaden.ibm.com/cs/k53/clever.html [14. Nov. 2001]

1. Bestimmung einer initialen Grundmenge von Web-Seiten zum Aufbau des Web-Graphen
2. Erweiterung der Grundmenge
3. Eliminierung interner Links
4. Berechnung von Hubs und Authorities

Bestimmung der Grundmenge Zur Bestimmung der Grundmenge wird eine Suchanfrage q an eine textbasierte generelle Suchmaschine (z. B. AltaVista) versendet. Von der zurückgelieferten Trefferliste werden die n am höchsten gewichteten URLs extrahiert (ein typischer Wert für n ist 200). Diese n durch ihre URLs repräsentierten Seiten bilden die Grundmenge R. Um aus der Grundmenge R einen Web-Graphen aufzubauen, werden die p Seiten als Knoten aufgefaßt. Existiert von einer Seite $p \in R$ ein Link auf eine andere p' mit $p' \in R$, so induziert dieser Link eine Kante (p, p') im zugehörigen Web-Graphen.

Erweiterung der Grundmenge In der Praxis zeigt sich, daß die Seiten in der Grundmenge R nur sehr schwach untereinander verlinkt sind. Hierauf können noch keine aussagekräftigen Berechnungen durchgeführt werden. Außerdem ist nicht gewährleistet, daß mit der Berücksichtigung der ersten n Treffer der generellen Suchmaschine tatsächlich auch genug qualitativ hochwertige Seiten gefunden wurden. Oft werden potentiell gute Hubs oder Authorities von den Suchmaschinen nur auf sehr niedrigen Rängen platziert. Deshalb wird die Grundmenge R erweitert, und zwar zu der Basismenge B. B besteht aus den Seiten von R, und zusätzlich aus Seiten die in R hinein verweisen, und aus Seiten, auf die von einer Seite aus R heraus verwiesen wird. Damit B nicht zu groß wird, wird ein Begrenzungsparameter d eingeführt: Jede Seite aus R darf maximal d Seiten in B miteinbringen, die auf sie verweisen.

Eliminierung interner Links G_0 bezeichnet den Graphen, der sich aus der Basismenge B ergibt. Existiert ein Hyperlink zwischen zwei Seiten, deren URL denselben Domain-Namen aufweist, so wird dieser als interner Link bezeichnet. Interne Links dienen meist nur der Verbesserung der Navigation innerhalb einer Web-Site, d. h. zwischen den beiden verbundenen Seiten besteht meist kein echter thematischer Bezug. Deshalb werden interne Links aus G_0 entfernt. Der resultierende Graph G bildet einen gerichteten Teilgraphen des WWW.

Berechnung von Hubs und Authorities Für eine Seite p kann das Hub-Gewicht $H(p)$ berechnet werden, indem die Authority-Gewichte $A(u)$ aller Seiten

u aufsummiert werden, auf die von p ausgehend ein Link gesetzt wurde ($p \rightarrow u$) – im Graph G existiert also eine gerichtete Kante von p nach u.

$$H(p) = \sum_{u \in B : p \rightarrow u} A(u)$$

Entsprechend wird das Authority-Gewicht $A(p)$ bestimmt, indem die Hub-Gewichte aller Seiten v aufaddiert werden, die einen Hyperlink auf p enthalten – im Graph G existiert also eine gerichtete Kante von v nach p.

$$A(p) = \sum_{v \in B : v \rightarrow p} H(v)$$

Die Berechnung der Hub- und Authority-Gewichte erfolgt iterativ. Zu Beginn werden die Hub- und Authority Gewichte aller Seiten in B auf den Wert eins gesetzt. In jedem Iterationsschritt werden die Gewichte für jede Seite neu bestimmt und anschließend normalisiert.

Man kann zeigen, daß die Gewichte konvergieren. Dies ist in der Regel bereits nach wenigen Iterationsschritten der Fall. Sei A die Adjazenzmatrix des zugehörigen Graphen G. Man kann nun zeigen, daß die Hub-Gewichte gegen den größten Eigenvektor der Matrix AA^T und die Authority-Gewichte gegen den größten Eigenvektor der Matrix $A^T A$ konvergieren (siehe hierzu [69]). Der größte Eigenvektor ist jener Vektor, dessen zugehöriger Eigenwert den höchsten Absolutwert aufweist.

5.1.1.1. Clustering mittels HITS

Hubs und Authorities verstärken einander und bilden sogenannte Communities. Durch die Links auf den Hub-Seiten werden die Authority-Seiten zu einem bestimmten Themengebiet gewissermaßen zusammengeklammert. Im Web-Graph werden Communities durch Teilgraphen identifiziert, die einen dichteren Grad an Verknüpfung aufweisen, als andere Teilbereiche des Graphen.

Der HITS-Algorithmus identifiziert jenen Teilgraph, der den dichtesten Verknüpfungsgrad aufweist. Zu bestimmten Suchanfragen können jedoch Seiten zurückgeliefert werden, die unterschiedlichen Bedeutungszusammenhängen bzw. unterschiedlichen Communities zugeordnet werden können. Für den allgemeinen Suchbegriff *jaguar* beispielsweise werden bei Verwendung einer generellen Suchmaschine Seiten gefunden, die thematisch vollkommen unterschiedlich voneinander sind. Die ausgeprägtesten Bedeutungen sind die Automarke, eine amerikanische Footballmannschaft (Jacksonville Jaguars) sowie ein Software-Programm (Atari Jaguar). Der Al-

gorithmus berechnet allerdings nur für jene Seiten die höchsten Hub- und Authority-Gewichte, die der ausgeprägtesten Community angehören. Da sich die meisten der *jaguar*-Web-Seiten auf die Automarke beziehen, bilden diese den dichtesten Teilgraph, was zur Folge hat, daß die weniger dichten Communities unberücksichtigt bleiben.

Kleinberg zeigt auf, daß sich auch die Hubs und Authorities der weniger dichten Communities bestimmen lassen. Diese werden durch die weiteren Eigenwerte der Matritzen AA^T bzw. A^TA und deren zugehörige Eigenvektoren bestimmt. Es gilt: AA^T und A^TA besitzen dieselben Menge von Eigenwerten. Das Vektoren-Paar $(\vec{x}_i, \vec{y}_i)$ bildet die zu einem Eigenwert λ_i gehörigen Eigenvektoren der Matrizen A^TA bzw. AA^T. Eine Community wird durch den Eigenwert λ_i bestimmt, je größer λ_i desto dichter ist der assoziierte Teilgraph. Die Koordinaten des Vektors $\vec{x}_i$ bestimmen die Hub-Gewichte der Seiten aus B und der Vektor $\vec{y}_i$ bestimmt die zugehörigen Authority-Gewichte. Je höher der Wert der Koordinate, desto höher ist das Gewicht der Seite, die durch die Koordinate identifiziert wird.

Werden also alle Eigenwerte und Eigenvektoren der Matrizen AA^T und A^TA berechnet, können hieraus die Hub- und Authority-Gewichte für alle Communities bestimmt werden.

5.1.2. Clustering von spezialisierten Suchservern

Um die Erschließung von spezialisierten Suchservern im WWW zu verbessern, erscheint die Anwendung von HITS in zweierlei Hinsicht als geeignet:

Die Analyse der Hyperlinkstruktur, in die die Suchserver eingebettet sind, erlaubt es, diese bzgl. ihrer thematischen Nähe zu gruppieren, ohne auf das Vorhandensein von beschreibendem Volltext angewiesen zu sein. Ein Suchserver wird dabei nur durch den URL seiner Startseite identifiziert. Die durch das HITS-Verfahren identifzierten Communites bilden somit Cluster von Suchservern. Diese Idee ist in Abbildung 5.1 graphisch dargestellt. Gleichzeitg wird durch das Authority-Gewicht eine qualitative Gewichtung aller Suchserver vorgegeben, die in einem einzelnen Cluster enthalten sind. Durch die Höhe des Community-spezifischen Eigenwertes werden die Cluster zusätzlich untereinander geordnet: Je höher der Eigenwert einer Community, desto höher ist deren Ausgeprägtheit bzw. deren Popularität im WWW einzuschätzen.

Im folgenden wird ein im Rahmen dieser Arbeit entwickeltes Verfahren aufgezeigt, das eine Menge von spezialisierten Suchservern unter Anwendung der HITS-basierten Community-Berechnung in Cluster aufteilt [56]. Im Anschluß werden eine Reihe von Experimenten durchgeführt, um die Qualität des Clusterings einschätzen zu können. Diese wurden unter Verwendung von Suchservern aus dem Lycos-Searchable Databases Katalog durchgeführt. In dem Katalog liegen die Suchserver bereits kategori-

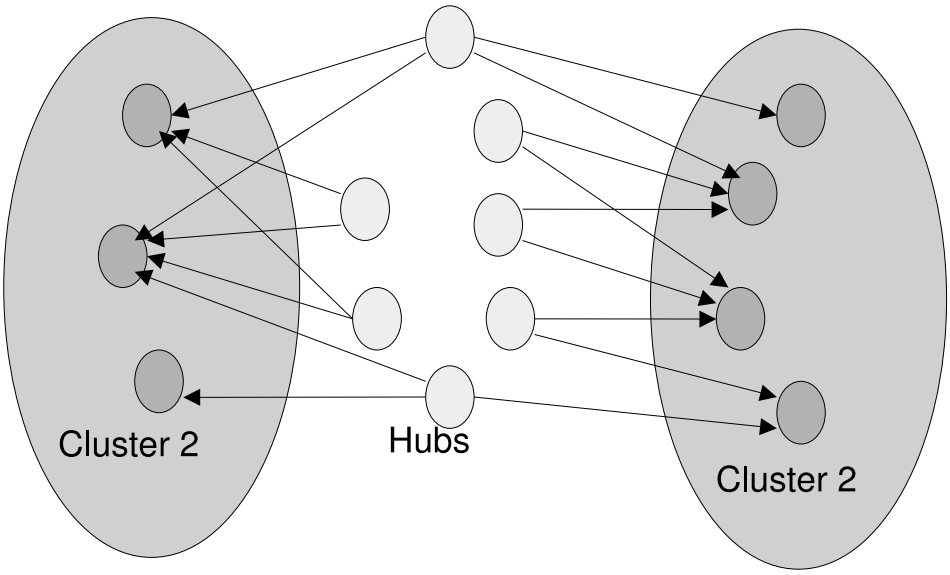

Abbildung 5.1.: Clustering von Suchservern

siert vor. Für die Experimente wurden Suchserver aus verschiedenen Kategorien ausgewählt, mittels HITS in Cluster aufgeteilt und überprüft, inwieweit Suchserver, die den gleichen Lycos-Kategorien angehören, auch durch das Cluster-Verfahren einander zugeordnet werden.

5.1.2.1. Beschreibung des Verfahrens zum Clustering von Suchservern

Zu Beginn wird manuell eine Menge S aus m Suchservern zusammengestellt, um einen ausreichend großen Web-Graphen aufzubauen. Jeder Suchserver $s_i \in S$ wird dabei ausschließlich durch den URL u_i repräsentiert. Dabei adressiert der URL u_i die Seite der Suchschnittstelle eines Suchservers s_i mit spezialisierten Inhalten.

In HITS wird eine einzelne Web-Seite gleichzeitig als Hub und Authority gewichtet, vorausgesetzt, sie enthält sowohl ein- als auch ausgehende Links. Die Suchschnittstellen von Suchservern enthalten in der Regel jedoch kaum ausgehende Links. Sind doch ausgehende Links enthalten, so handelt es sich meist um interne Links zur Unterstützung der Navigation auf der Web-Seite. Dementsprechend werden alle $s_i \in S$ ausschließlich als Authorities gewichtet.

Der erste Schritt besteht im Auffinden von Hub-Seiten, die auf möglichst viele Ele-

mente in S gleichzeitig verweisen. Hierzu wird eine generelle Suchmaschine zu Hilfe genommen, die das Identifizieren von Backlink-Seiten unterstützt.

Für jedes Element $s_i \in S$ werden unter Verwendung der Backlink-Suche m Seiten identifziert, die einen Hyperlink auf s_i enthalten. Alle derart identifzierten potentiellen Hubs h_j werden in eine Menge H eingefügt. Anschließend wird über die Elemente $h_j \in H$ iteriert und all jene h_j wieder entfernt, die nicht mindestens auf r Elemente in S verweisen. Damit liegt eine Menge von Authorities und Hubs in getrennten Mengen vor (S respektive H).

Aufbauend auf S und H wird die Adjazenzmatrix A des resultierenden Graphen gebildet, mit

$$A(i,j) = \begin{cases} 1 & \text{falls gilt: Es existiert ein Link von einer Seite } i \text{ zu einer Seite } j \\ 0 & \text{sonst} \end{cases}$$

Die Dimension n von A ergibt sich aus der Größe der zugrundeliegenden Mengen S und H, d. h. $n = size(S) + size(H)$. Durch das Variieren der Parameter r und m kann die Größe der Menge H und damit die Dimension von A beeinflußt werden.

Anschließend werden die Eigenwerte und Eigenvektoren der Matrix $A^T A$ berechnet und hierdurch die Cluster abgeleitet. Jeder Eigenwert identifziert einen Cluster, und jede Koordinate $\neq 0$ des zugehörigen Eigenvektors identifziert ein Element des Clusters.

Der folgende Algorithmus zeigt das Verfahren in Pseudo-Code:

```
computeDatabaseClusters(S,m,r):

   # Die Menge der Hubs ist zu Beginn leer
   H={}

   for u in S do:
        query = link:<u>

        # Durch Abfragen einer generellen Suchmaschine
        # werden maximal m URLs von potentiellen
        # Hub-Seiten in die Hub-Menge H eingefügt
        querySearchEngine(query,m,H)

   # Es sollen nur solche Hubs berücksichtigt werden,
   # die auf mindestens r Authorities in S verweisen
   for hub in H do:
        if getNumberOfLinksPointingToS(hub,S) <= r:
```

```
            delete hub from H

# Erzeuge aus den Mengen H und S die zugehörige
# Adjazenezmatrix A sowie die transponierte Matrix AT
A=createAdjacencyMatrix(H,S)
AT=transpose(A)

# Berechne die Eigenwerte und Eigenvektoren von AT*A
return computeEigenvalues(AT*A)
```

5.1.2.2. Experimente

Durch die folgenden Experimente soll aufgezeigt werden, inwieweit das beschriebene Verfahren dazu geeignet ist, um spezialisierte Suchserver automatisiert in Cluster zu gruppieren. Zur Erzeugung der Menge S werden Seiten aus dem Lycos-Katalog herangezogen, die verschiedenen Kategorien angehören. Anschließend wird überprüft, inwieweit die durch die Kategorien vorgegebene Gruppierung in den erzeugten Clustern reproduziert werden kann.

Es wurden circa 30 Experimente durchgeführt, in denen jeweils die Größen der Parameter r und m modifiziert wurden. Die Suchserver wurden aus verschiedenen Lycos-Kategorien herangezogen, wobei pro Experiment jeweils Seiten berücksichtigt wurden, die drei bis vier verschiedenen Lycos-Kategorien angehörten.

Im folgenden werden die Ergebnisse eines typischen Experimentes tabellarisch zusammengestellt. Jede Zeile repräsentiert einen durch einen Eigenwert identifizierten Cluster. Dabei zeigt der Wert in der ersten Spalte den konkreten Eigenwert und der zweite die Gesamtanzahl der Suchserver, die dem Cluster zugeordnet wurden, also jene, für die im Eigenvektor in der zugehörigen Koordinate ein Wert $\neq 0$ berechnet wurde. Die restlichen Spalten geben an, wieviele der Clusterelemente den ursprünglichen Lycos-Kategorien angehören.

Für dieses Experiment wurden 50 Suchserver mit wissenschaftlichen Inhalten aus dem Lycos-Katalog ausgewählt. Davon gehören 20 der Kategorie Geologie (*geology*), 9 der Kategorie Physik (*physics*) und 21 der Kategorie Astronomie (*astronomy*) an.

Die Menge H besteht aus 67 Hubs, wobei jeder Hub auf mindestens 3 ($= r$) Elemente aus S verweist, d. h. die Dimension n der Adjazenzmatrix A beträgt 117. Die Gesamtanzahl der berücksichtigten Links beträgt 307.

Die folgende Tabelle zeigt die Resultate des Clusterings. Eigenwerte, die mehr als einen Eigenvektor besitzen, werden nicht berücksichtigt. Jeder Eigenwert bzw. jede Zeile identifiziert einen Cluster, wobei die Cluster nach absteigendem Eigenwertbetrag

sortiert dargestellt werden.

Eigenwertbetrag	Anzahl Clusterelemente	Science(50)		
		geology(20)	physics (9)	astronomy (21)
60.22	10	10	0	0
60.0	6	6	0	0
40.55	25	0	8	17
24.86	25	0	9	16

Tabelle 5.1.: Experiment: Wissenschaftliche Suchserver

Die Tabelle zeigt, daß zehn bzw. sechs der geologischen Suchserver in den ersten beiden Clustern zusammen gruppiert wurden. Der durch den größten Eigenwert identifizierte Cluster enthält 10 Elemente, von denen alle der Geologie-Kategorie angehören. Die Physik- und Astronomie Suchserver dahingegen werden durch das Verfahren weniger klar getrennt und gemeinsamen Clustern zugeordnet. Eine Erklärung hierfür mag die thematische Nähe zwischen den Gebieten Astronomie und Physik sein. Die berücksichtigten Hub-Seiten enthalten sowohl Links zu Physik Web-Sites als auch zu Astronomie Web-Sites.

5.1.2.3. Bewertung der Anwendung von HITS

Die Experimentreihe wurde im Zeitraum vom Mai - Juli ’99 durchgeführt. Zusammenfassend zeigte sich, daß die Ergebnisse ähnliche Schlüsse zulassen wie sie auch von den Autoren des ARC-Artikels [28] gezogen wurden (siehe Abschnitt 2.4.1.3): Die Anwendung von HITS liefert gute Ergebnisse bei Themengebieten, die klar von anderen Wissensbereichen abgegrenzt sind. Die Qualität der Ergebnisse verschlechtert sich zunehmend, je stärker die Themen mit anderen Wissensbereichen verknüpft sind, was sich dann auch in der Hyperlinkstruktur des WWW widerspiegelt, z. B. durch Berührungen mit politischen oder kommerziellen Bereichen.

So liefert der ebenfalls auf HITS basierende ARC-Algorithmus beispielsweise gute Ergebnisse für die Themengebiete *cheese*, *alcoholism* oder *zen buddhism*, da diese eine relative geringe Präsenz im WWW aufweisen und jedes für sich eine kompakte Community bildet. Dahingegen wurden schlechte Ergebnisse erzielt für die Gebiete *affirmative action* (auf deutsch in etwa: *Förderungsmaßnahmen zugunsten von Minderheiten*) oder *mutual funds* (auf deutsch: *Wertpapierdepots*).

Vergleichbare Ergebnisse wurden durch die Cluster-Experimente erzielt, die im Rahmen dieser Arbeit vorgenommen wurden. Gute Ergebnisse, d. h. Suchserver, die aufgrund der vorgegebenen Kategorisierung von Lycos diesem Themengebiet angehören, wurden relativ kompakt in gleichen Clustern gruppiert, wurden erzielt für die Gebiete:

geology, *brewery* oder *golf sports*, wohingegen thematisch breiter ausgelegte Communities wie *physics* oder *arts* nur sehr unzuverlässig zusammen gruppiert wurden.

Die im Rahmen dieser Arbeit durchgeführten Experimente haben gezeigt, daß die Qualität eines HITS-basierten Clusterings nicht ausreicht, um eine effektive Erschließung von Suchservern zu unterstützen. In einer neueren Arbeit [16] wird ein vergleichbares Fazit gezogen. Des weiteren erwies sich die Berechnung von Eigenwerten mittels der JSci-Bibliothek[2] als wenig performant. Alternativ wurde eine zweite Version implementiert, in der die Matritzen-Berechnungen durch das Mathematik-Programm *maple* vorgenommen wurde, was jedoch auch keine Verbesserungen für die Performanz erbrachte [52]. Aus diesen Gründen wurde der Ansatz im Rahmen dieser Arbeit nicht weiterverfolgt.

Stattdessen wurde ein neuartiges Verfahren zur automatischen hyperlinkbasierten Kategorisierung von Web-Ressourcen entwickelt und eingesetzt, um eine thematische Einordnung von Suchservern vorzunehmen. Das Verfahren sowie die Ergebnisse von einigen hiermit durchgeführten Experimente werden im folgenden Abschnitt dargestellt.

5.2. Automatisierte Kategorisierung von Web-Ressourcen

Mit dem Feld *category_terms* bietet der Frankfurt Core Platz, um existierende Kategorisierungen von Web-Katalogen zur Beschreibung von Suchservern aufzunehmen. In Abschnitt 4.4.1 wird eine entsprechende Strategie vorgestellt, nach der derartige Kategorisierungen unter Verwendung einer generellen Suchmaschine aus den Web-Katalogen automatisiert ausgelesen werden können. Die Menge der Kategorien, die einem Suchserver zugeordnet werden können, ergibt sich dabei aus der vereinigten Menge aller verwendeten Kategorien der berücksichtigten Web-Kataloge.

Die meisten Web-Kataloge für Ressourcen des Sichtbaren und des Unsichtbaren Webs (siehe Abschnitt 2.3.1 und 3.1.3) verzeichnen jedoch meist nur einige tausend Suchserver, wovon die meisten gleichzeitig in mehreren Katalogen kategorisiert vorliegen. In Anbetracht der geschätzten Anzahl von Suchservern im WWW von 100.000 (siehe Abschnitt 3.1.2) ist anzunehmen, daß ein Großteil der existierenden Suchserver noch nicht von einem der Web-Kataloge erfaßt wurde. Dementsprechend müssen in einem System für Verteiltes Information Retrieval auch Möglichkeiten zur automatisierten Zuordnung von Kategorien zu Suchservern bereit gestellt werden.

In diesem Abschnitt wird ein hyperlinkbasiertes Kategorisierungsverfahren vorge-

[2] Java Science API: http://fourier.dur.ac.uk:8000/~dma3mjh/jsci/ [14. Nov. 2001]

stellt, das ohne die Erstellung von Lernstichproben auskommt. Vielmehr wird das WWW als globale Wissensbasis verwendet: die Zuordnung von Kategorien zu Web-Ressourcen basiert ausschließlich auf der Auswertung von globalen Term- und Linkhäufigkeitsinformationen wie sie unter Verwendung einer generellen Suchmaschine ermittelt werden können.

5.2.1. Das WWW als globale Wissensbasis

Die Gesamtheit der im WWW gespeicherten Datenmengen bildet eine globale Wissensbasis von enormer Größe. Allerdings ist das hinterlegte Wissen primär nur von Menschen interpretierbar und anwendbar – der Aufbau des semantischen Webs steht noch weit am Anfang. Dennoch lassen sich bereits heute eine Vielzahl von Systemen und Algorithmen identifizieren, die durch das Anwenden verschiedenster Heuristiken ansatzweise einen automatisierten Zugriff auf gespeichertes Wissen erlauben:

- Das System STRAND [98] beispielsweise nutzt die Tatsache aus, daß für viele WWW-Seiten alternativ Übersetzungen in verschiedene Fremdsprachen angeboten werden. Die übersetzten WWW-Seiten können oft durch einen entsprechend benannten Link identifiziert werden (meist werden diese durch ein entsprechendes Flaggensymbol angezeigt). Eine systematische Analyse dieser Seiten inklusive ihrer fremdsprachigen Äquivalente erlaubt die qualitative Verbesserung von automatischen Übersetzungen.

- Das System Mulder [72] nutzt das WWW, um automatisiert Fragen zum Allgemeinwissen zu beantworten (z. B. Wie heißt der zweithöchste Berg der Erde?). Dabei wird die Suchmaschine Google verwendet, um potentielle Antwortseiten aus dem WWW zu identifizieren. Zur Analyse der Anfragesätze und der Ergebnisseiten werden NLP (Natural Language Processing)-Techniken eingesetzt.

- Das System InCommonSense [5] verwendet ebenfalls NLP-Techniken sowie verschiedenste Heuristiken, um aus den Ankertexten der Backlink-Seiten jene Beschreibungen zu identifzieren, die eine referenzierte Ressource am treffendsten charakterisieren.

- Im System QPilot [110] (siehe auch Abschnitt 3.3.1.4) werden durch Abfragen einer generellen Suchmaschine thematisch verwandte Terme zu einer Suchanfrage aus den Snippets der gefundenen Dokumente extrahiert, um eine Anfrageerweiterung durchzuführen.

5.2.2. Web-basierte Dokumenthäufigkeiten

Bei den vorgestellten Systemen bleibt allerdings eine Information ungenutzt, die, systematisch ausgewertet, geeignet erscheint, um weiterführende Aussagen abzuleiten: die Größe der durch eine Anfrage erzeugten Treffermenge.

Über das Abfragen von generellen Suchmaschinen wie AltaVista oder Google läßt sich die Dokumenthäufigkeit von Termen gemessen über das gesamte WWW ermitteln. So findet sich auf der für eine Suchanfrage zurückgelieferten Resultatseite an einer eindeutig identifizierbaren Position auch die Information über die Größe der Treffermenge (z. B. AltaVista found 512.318 Documents) und kann somit leicht durch automatische Verfahren extrahiert werden.

Die Verwendung dieser Größenangaben läßt sich an einem einfachen Beispiel demonstrieren: Ist man sich beispielsweise über die korrekte Schreibweise des Termes Libyen nicht im Klaren (oft werden y und i vertauscht), so könnte über eine generelle Suchmaschine eine quantitative Evaluierung der möglichen syntaktischen Varianten durchgeführt werden. In AltaVista ergab eine Abfrage *Lybien* über alle deutschen Web-Seiten 781 Treffer, wohingegen Libyen 10.307 Treffer erzielte, was den Rückschluß auf die korrekte Schreibweise zumindest erleichtert.

Im Gegensatz zu Häufigkeiten, die aus abgeschlossenen Testkollektionen (siehe Abschnitt 2.1.4) abgeleitet wurden, wird der Index einer Web-Suchmaschine in regelmäßigen Abständen aktualisiert und enthält weitaus mehr unterschiedliche Terme als ein einfacher Thesaurus.

Darüber hinaus lassen sich aus der Beobachtung, wie oft zwei verschiedene Terme gemeinsam in Dokumenten auftauchen (Ko-Zitierung), Rückschlüsse ziehen, inwieweit diese thematisch miteinander korrelieren. Im WWW lassen sich selbst Ko-Zitierungen für Terme finden, zwischen denen offensichtlich keine thematische Verbindung besteht. Beispielsweise ergab eine Anfrage an AltaVista über alle Dokumente, in denen die beiden zusammengesetzten Suchterme *“gobi desert”* und *“wolfgang amadeus mozart”* auftauchen – diese hat die Form: *+”gobi desert” +”wolfgang amadeus mozart”* – immerhin noch fünf Dokumente. Berücksichtig man nun die absolute Häufigkeit der einzelnen Terme (*“gobi desert”*:5356 und *”wolfgang amadeus mozart”*:30615) läßt sich hieraus schließen, daß beide Terme wesentlich geringer miteinander korrelieren als beispielsweise *“wolfgang amadeus mozart”* und *“salzburg”*, die zusammen immerhin in 2079 verschiedenen Dokumenten auftauchen.

Diese einfachen Beispiele vernachlässigen natürlich die Tatsache, daß ein einzelner Begriff unterschiedlichen Bedeutungskontexten zugeordnet werden kann (Homonyme). Außerdem sind die Häufigkeitsangaben der Suchmaschinen relativ unzuverlässig und können von Anfrage zu Anfrage mal mehr mal weniger stark voneinander abwei-

chen. In Anbetracht der enormen Menge von Dokumenten, die von generellen Suchmaschinen erfaßt sind fallen diese Nachteile jedoch nicht mehr so stark ins Gewicht. Unterschiedliche Bedeutungskontexte können z. B. durch das Clustering von Snippets mit Verfahren linearer Komplexität wie Suffix Tree Clustering (STC) (siehe Abschnitt 2.4.2.1 identifiziert werden.

Über die Häufigkeit eines Begriffs t_i im WWW läßt sich somit eine globale Dokumenthäufigkeit (df_i) für t_i ableiten und erlaubt daher Einschätzungen über die generelle Eignung von t_i als Deskriptor.

5.2.3. Eine hyperlinkbasierte Kategorisierung von Web-Ressourcen

Thematische Kategorien, die von Web-Katalog-Betreibern wie Yahoo zur Klassifizierung der Web-Ressourcen eingesetzt werden, orientieren sich mehr oder weniger an generellen Klassifikationsschemata aus der bibliothekarischen Welt, wie beispielsweise der Dewey Decimal Classification. Berücksichtigt werden auch spezialisierte Klassifikationsschemata, die innerhalb verschiedener Fachdisziplinen einen breiten Grad an Akzeptanz erreicht haben, wie z. B. die Medical Subject Headings (MeSH) in der Medizin oder die CACM-Klassifikation in der Informatik.

Die Kategorienbezeichner finden sich nicht nur innerhalb von Web-Katalogen (also Seiten mit hohem Hub-Gewicht) wieder, sondern auch auf Seiten, die eine hohe Autorität für ein bestimmtes Themengebiet aufweisen. Die systematische Auswertung der Vorkommenshäufigkeiten von Katgorienbezeichnern im WWW bietet somit einen Hinweis auf den thematischen Bezug von Web-Ressourcen und damit die Möglichkeit effizient eine Zuordnung von Web-Ressourcen zu Kategoriebezeichnern vorzunehmen. Im Gegensatz zu traditionellen Kategorisierungsverfahren ist die vorherige Erzeugung und Auswertung von Lernstichproben nicht vonnöten. Die Zeichenkette des Kategorienbezeichners sowie eine generelle Suchmaschine zur Determinierung von globalen Dokumenthäufigkeiten reichen aus, um Web-Ressourcen, die nur durch ihre URL identifiziert sind, zu klassifizieren. Dies schließt auch die Kategorisierung von Web-Resourcen mit ein, die noch nicht von einem der großen Web-Kataloge erfaßt wurden. Voraussetzung ist lediglich, daß bereits eine ausreichend hohe Anzahl an Hyperlinks auf die zu klassifizierende Ressource verweisen.

Im folgenden wird eine solche Methode vorgestellt: Unter Verwendung einer generellen Suchmaschine wird die Anzahl der Web-Seiten ermittelt, die gleichzeitig einen Link auf eine zu klassifizierende Ressource und einen bestimmten Kategorienbezeichner enthalten. Hieraus wird ein Gewicht zwischen Ressource und Kategorie errechnet. Ist dies für jede Kategorie geschehen, können anschließend für eine Ressource die

Kategorien nach absteigendem Gewicht angeordnet werden. Um eine möglichst objektive Einschätzung über die Güte einer automatisierten Kategorisierung von WWW-Dokumenten zu erhalten, werden oft Vergleiche mit den manuell erzeugten Katalogen großer Internet-Portale (z. B. Yahoo oder Lycos) durchgeführt (siehe Abschnitt 2.4.3.1). Für die Experimente werden deshalb Kategorien verwendet, die sich aus den Teilbäumen des Yahoo-Katalogs ergeben[3].

In den folgenden Abschnitten wird zunächst die Extraktion der Kategorienbezeichner aus dem Yahoo-Katalog und die Zusammenstellung der Testmenge beschrieben. Die Testmenge besteht aus Dokumenten, die bereits kategorisiert im Yahoo-Katalog vorliegen, und dient der abschließenden Evaluierung des Verfahrens. Anschließend erfolgt die Darstellung der eigentlichen Kategorisierungsmethode. Zu deren Evaluierung wurde eine Reihe von Experimenten durchgeführt. Dabei wurden die Dokumente der Testmenge kategorisiert und für diese überprüft, zu welchem Grad die Kategorienzuweisung von Yahoo durch das beschriebene Verfahren reproduziert werden konnte. Abschließend erfolgt eine Interpretation der Resultate.

5.2.3.1. Zusammenstellung der Kategorien und Testmenge aus dem Katalog von Yahoo

Der Katalog von Yahoo besitzt eine baumartige Struktur:

- Die Wurzel des Baumes besteht aus der Startseite der Yahoo Web-Site.
- Die inneren Knoten entsprechen den Unterverzeichnissen, wobei jeder Knoten einen Bezeichner t besitzt.
- Die Blätter entsprechen den Web-Dokumenten externer Server. Auf diese wird von Yahoo aus über einen externen Link verwiesen.

Die mit den inneren Knoten assoziierten Seiten liegen auf demselben Server auf und sind untereinander durch lokale Links miteinander verbunden. Jeder innere Knoten kann sowohl externe Links auf die Blätter – also die klassifizierten externen Web-Ressourcen – als auch lokale Links auf die Seiten weiterer Unterverzeichnisse enthalten, die das Themengebiet weiter einschränken. Ein lokaler Link ist mit einem Anker-Text t_i unterlegt, der das Unterverzeichnis, und damit das Themengebiet, identifiziert, wie z. B. *science* oder *artificial intelligence* etc.

Unter Berücksichtigung der Besonderheiten des Yahoo-Katalogs wird für das Verfahren eine Kategorie folgendermaßen gebildet:

[3] Es hätten natürlich auch andere Web-Kataloge für diesen Zweck verwendet werden können, zumal die Kategorienbezeichner und die Katalogstruktur verschiedener Web-Kataloge große Ähnlichkeiten aufweisen.

- Eine Kategorie c_i ist eine Liste, bestehend aus den Ankertexten t_j, die mit jenen inneren Knoten assoziiert sind, die auf dem Pfad von der Wurzel bis zu einem bestimmten Unterverzeichnis liegen, also $c_i = (t_1, \ldots, t_n)$, wobei n der Länge eines individuellen Pfades entspricht. Hierdurch wird dem Umstand Rechnung getragen, daß manche Ankertexte t_j innerhalb des Yahoo-Baumes mehrfach in unterschiedlichen thematischen Kontexten auftauchen können und somit nicht eindeutig sind. Beispielsweise wird mit dem Anktertext *architecture* ein Unterverzeichnis des *computer science*-Teilbaumes bezeichnet, in dem Web-Ressourcen über Computerarchitekturen abgelegt sind. Gleichzeitig existiert ein Unterverzeichnis im *arts and humanities*-Teilbaum, das ebenfalls *architecture* bezeichnet wird. Hierin befinden sich Ressourcen, die sich mit Gebäudearchitekturen beschäftigen. Zur Vereinfachung der Lesbarkeit wird im folgenden jedoch eine Kategorie immer mit dem letzten Ankertext der Liste t_n bezeichnet, d. h. die Kategorie $c_i = ($*science*, *computer science*, *algorithms*$)$ wird mit *algorithms* bezeichnet. Außerdem werden die Ankertexte in Kleinschreibweise überführt und Sonderzeichen entfernt.

- Im Yahoo-Katalog können manche Verzeichnisse durch interne Verweise mehreren Knoten gleichzeitig untergeordnet werden. Beispielsweise kann die Kategorie *robotics (science, engineering, mechanical engineering, robotics)* auch über einen internen Verweis aus der *computer science*-Kategorie erreicht werden. Im Yahoo-Katalog sind derartige Verweise durch ein vorangestelltes @-Zeichen in den Ankertexten gekennzeichnet und sind somit leicht zu identifizieren. Für das hier vorgestellte Verfahren werden derartige interne Verweise ignoriert, da sie die Baumstruktur des Katalogs unterlaufen, d. h. jede Kategorie wird nur über ihren direkten Pfad identifiziert.

Für die Experimente muß eine geeignete Testmenge bereitgestellt werden. Diese setzt sich aus den von Yahoo klassifizierten Dokumenten, der in den Experimenten berücksichtigten Kategorien zusammen, also den Blättern des Yahoo-Baumes. Jedes zu kategorisierende Dokument ist dabei ausschließlich über deren URL identifiziert. Damit im Anschluß an ein Experiment eine Aussage über die Qualität der automatisierten Kategorisierung abgeleitet werden kann, werden für jedes Dokument zusätzlich der Ankertext des externen Links und die Original-Kategorie mitabgespeichert. Ein zu kategorisierendes Dokument wird somit durch das folgende Tripel beschrieben:
([URL],[Anker-Text],[Kategorie]),
also z. B. : *(“http://www.speech.cs.cmu.edu/speech/”,”Speech at Carnegie Mellon University”,(science, computer science, artificial intelligence, natural language processing))*.

5.2.3.2. Kategorisierungsmethode

Die Kategorisierungsmethode basiert auf der Berechnung eines Gewichtes zwischen Dokument und Kategorie. Für jedes einzelne Dokument werden die Kategorien nach absteigendem Gewicht angeordnet.

Grundlage für die Berechnung des Gewichtes ist die Messung, auf wieviel Web-Seiten ein Hyperlink auf einen bestimmte URL und die einzelnen Bezeichner einer Kategorie gemeinsam auftauchen (also ko-zitiert werden). Für eine Web-Seite, identifiziert durch seine URL u und eine Kategorie c mit $c = (t_1, \ldots, t_n)$, wird das Gesamtgewicht $weight(u, c)$ durch die folgende Funktion berechnet:

$$weight(u, c) = \sum_{i=1}^{n} f_{u,i} * log \frac{N}{df_i}$$

Dabei ist:

N: Gesamtanzahl der von AltaVista indexierten Web-Seiten (im Juli 2000 etwa 350 Mio).
df_i: Anzahl der Web-Seiten, die den Term t_i enthalten, oder 1, falls keine solche Web-Seite existiert.
$f_{u,i}$: Anzahl der Web-Seiten die einen Hyperlink auf die Web-Seite mit der URL u und den Term t_i enthalten.

Der Teil $(log \frac{N}{df_i})$ bestimmt die inverse Dokumenthäufigkeit (siehe Abschnit 2.1.3.1) eines Terms t_i gemessen über die Anzahl der indexierten Web-Seiten.

Die für die Experimente analysierten Yahoo-Seiten enthalten ebenfalls sowohl die Kategorienamen als auch die entsprechenden Hyperlinks. Dies kann insbesondere bei Web-Seiten, auf die im WWW nur in geringem Maße verwiesen wird, zu einer erheblichen Verfälschung der Ergebnisse führen, da diese Seiten bei der Messung der Ko-Zitierungen von AltaVista natürlich mitberücksichtigt werden. Deshalb wird jede an AltaVista versendete Abfrage mit dem Zusatz *-host:yahoo* versehen. Dieser schließt alle Rechner aus, deren Adresse den Substring *yahoo* enthalten. Um beispielsweise die Häufigkeit $f_{u,i}$ zu bestimmen, wird die Anfrage: *+"[t_i]" +link:[u] -host:yahoo* an AltaVista versendet.

5.2.4. Experimente

Inwieweit das beschriebene Verfahren dazu geeignet ist, die Kategorien von Web-Seiten vorherzusagen, wurde durch eine Reihe von Experimenten untersucht. Diese

sind im folgenden beschrieben.

In einer ersten Experimentreihe werden die URLs und Kategorien von drei thematisch getrennten Teilbereichen des Yahoo-Katalogs extrahiert und für jeden untersucht, inwieweit die enthaltenen URLs in der durch die von Yahoo vorgegebene Ordnung einander zugeordnet werden können. Unterkategorien, in denen URLs nach nicht themenbezogenen Kriterien gesammelt wurden, werden in diesen Experimenten nicht berücksichtigt. Typische Beispiele für nicht berücksichtigte Kategorien sind *(science, computer science, artificial intelligence, institutes)* oder *(science, computer science, journals)*, also Kategorien, in denen die Adressen der Startseiten von Instituten oder Online-Journalen hinterlegt sind – unabhängig von einem thematischen Bezug.

Bei der Auswahl der Kategorien wurde darauf geachtet ein möglichst umfassendes Spektrum des übergeordneten Wissensgebietes zu erfassen. Insgesamt wurden ausgewählt:

- 382 URLs aus dem Bereich *biology* mit insgesamt 23 Kategorien,
- 202 URLs aus dem Bereich *computer science* mit insgesamt 18 Kategorien, sowie
- 225 URLs aus dem Bereich *arts and humanities* mit insgesamt 14 Kategorien.

In einem abschließenden Experiment (*alltogether*) wurden alle Dokumente zusammen (809) mit allen Kategorien (55) untersucht. Dies bedeutet, daß z. B. ein Dokument aus dem Bereich *computer science* nun auch unter Berücksichtigung der Kategorien aus den Bereichen *arts and humanities* und *biology* hin eingeordnet werden kann. Hierdurch soll untersucht werden, ob die Ergebnisse stabil bleiben, wenn zusätzlich auch themenfremde Kategorien in die Betrachtung miteingeschlossen werden.

Für jedes Dokument wurde eine Kategorien-Rangliste gemäß den berechneten Gewichten für jede einzelne Kategorie erstellt. Tabelle 5.2 stellt die Ergebnisse der 4 Experimente dar. Die Zahl in den Zellen gibt jeweils an, für wieviele der am Experiment beteiligten Dokumente die jeweils richtige Kategorie an die entsprechende – in den Spalten dargestellte – Position der Rangliste gesetzt wurden. Also z. B. im Experiment *biology* wurde für 293 der 382 Dokumente die von Yahoo vergebene Kategorie durch das Verfahren an die erste Position gesetzt, also 76,7%. Für 35 Dokumente war hingegen die richtige Kategorie an der zweiten Position zu finden etc.

Für einige der Dokumente konnte überhaupt keine Gewichtung für mindestens eine der Kategorien ermittelt werden. Hierbei handelt es sich um Dokumente, für die keine Ko-Zitierung von Termen der Kategorien und Links auf diese Seite festgestellt werden konnten. Diese sind in der vorletzten Spalte von Tabelle 5.2 dargestellt. Hierdurch

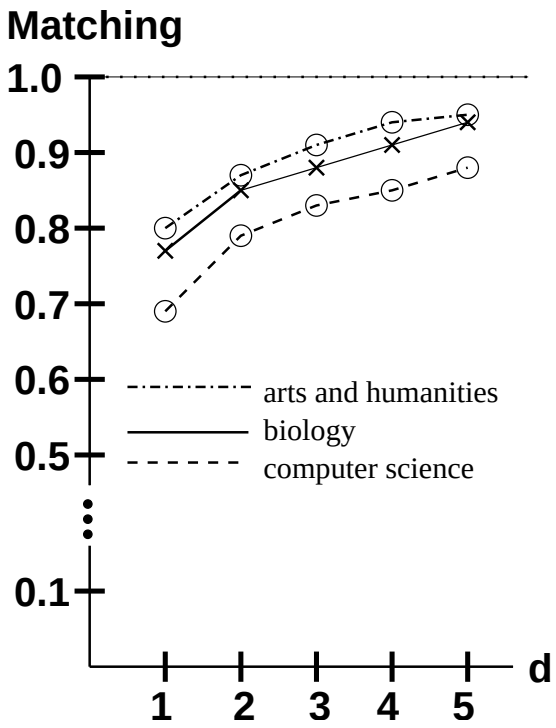

Abbildung 5.2.: Übereinstimmung zwischen der automatischen Kategorisierung und der Yahoo-Kategorisierung in den drei Einzelexperimenten

wird auch klar: Je mehr Links auf eine Seite verweisen, desto höher ist die Wahrscheinlichkeit, daß diese zusammen mit den Kategorienamen zitiert werden, und desto zuverlässiger kann eine Kategorisierung vorgenommen werden.

Experiment/Position	1	2	3	4	5	≥ 6	nicht zugeordnet	gesamt
biology (23)	293 (76.7%)	35	10	10	7	15	12 (3.1%)	382
computer science (18)	140 (69%)	21	8	4	6	16	7 (3.5%)	202
arts and humanities (14)	180 (80%)	19	4	6	1	4	11 (4.9%)	225
alltogether (55)	600 (74.2%)	81	19	25	12	42	30 (3.7%)	809

Tabelle 5.2.: Resultate der automatischen Kategorisierung

Die Abbildung 5.2 zeigt getrennt nach den drei einzelnen Experimenten den prozentualen Anteil jener Dokumente, deren tatsächliche Kategorie sich unter den ersten d Elementen der Kategorien-Rangliste befand. Beispielsweise befand sich im *computer science*-Experiment die korrekte Kategorie für 69% aller Dokumente an der ersten Position ($d = 1$). Für 79,4% aller *computer science*-Dokumente befand sich die richtige Kategorie unter den ersten zwei Postionen ($d = 2$), usw.

In Abbildung 5.3 ist das Ergebnis des Experimentes aufgezeigt, in dem alle Kategorien und Dokumente zusammen berücksichtigt wurden.

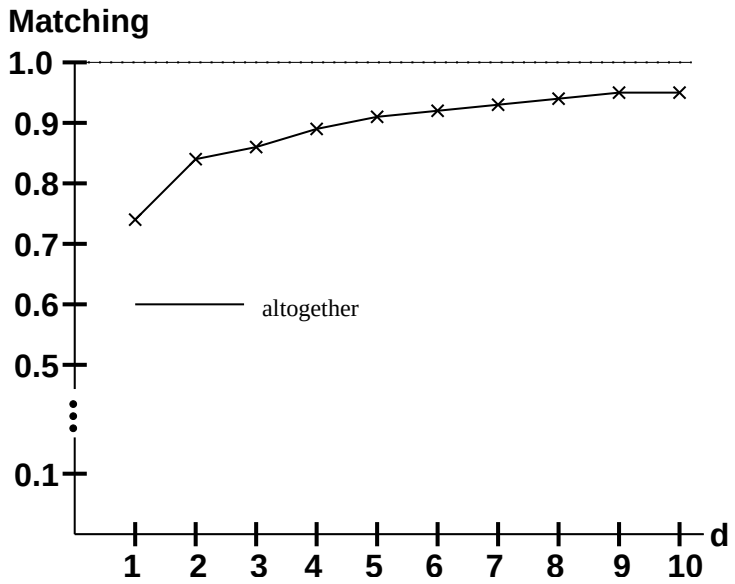

Abbildung 5.3.: Übereinstimmung zwischen der automatischen Kategorisierung und der Yahoo-Kategorisierung über alle Dokumente und Kategorien

5.2.5. Interpretation der Ergebnisse

Zusammengefaßt zeigt sich, daß die Kategorien mit einer Zuverlässigkeit von circa 75% korrekt zugeordnet werden konnten. *computer science*-Dokumente weisen die schlechteste Zuweisung auf – nur für 69% der Dokumente wurde die korrekte Kategorie auf der ersten Position plaziert. *arts and humanities*-Dokumente weisen mit einer korrekten Zuweisung von 80% das beste Ergebnis auf. In über 90% der Fälle befindet sich die korrekte Kategorie unter den ersten 5 Positionen. Etwa 3% bis 5% aller URLs können überhaupt nicht zugeordnet werden, weil keine Backlink-Seiten mit entsprechenden Kategoriebezeichnern gefunden werden konnten.

Betrachtet man die berechneten Ranking-Listen der Dokumente im einzelnen, so fällt auf, daß wenn die korrekte Kategorie nicht an die erste Position gesetzt wurde, oft eine thematisch nah verwandte Kategorie auf die erste Position berechnet wird. Die korrekte Kategorie befindet sich dann meist auf einer der nachfolgenden Top-Positionen. Im Experiment *biology* tauchen z. B. oft die 3 Kategorien *biodiversity, systematics and taxonomy* und *botany* nah beieinander auf. Ähnlich verhalten sich auch die Kategorien *genetics, molecular biology* und *cell biology*. Im Experiment *computer science* lassen sich solche Nähen, z. B. zwischen den Kategorien

- *user interface* und *human computer interaction*,
- *robotics* und *fuzzy logic*,
- *human computer interaction* und *natural language processing*.

beobachten.

Manchmal weist ein falsches Ranking auch auf einen Kategorisierungsfehler hin, der von Yahoo begangen wurde. So wird unter der Kategorie *parasitology* eine Seite mit dem Titel „National Geographic: Parasites“ und der URL *http://www.nationalgeographic.com/index.html*, also zur Start-Seite von National Geographics angeboten. Die wahrscheinlich gemeinte Seite befindet sich unter der URL *http://www.nationalgeographic.com/parasites*. Dementsprechend wurde im Experiment *biology* die Kategorie *parasitology* für diese Seite nur auf die Position 15 (von 23) der Rangliste gesetzt.

Im letzten Experiment wurden sämtliche Kategorien und Dokumente zusammen berücksichtigt. Die Auswertung ergab, daß die Resultate hierdurch nicht wesentlich beeinflußt werden. Dies ist insbesondere deshalb interessant, weil die Kategorien aus dem *computer science*-Bereich auch Terme wie *internet* und *computer* enthalten, also Begriffe, die im WWW sehr weit verbreitet sind – auch außerhalb der Computerwissenschaften.

Eine detaillierte Auswertung der erzeugten Kategorien-Ranglisten einzelner Dokumente des letzten Experimentes zeigt, daß für einige Dokumente Kategorien aus allen drei Teilbereichen sehr hoch gewichtet wurden. Eine anschließende Analyse der Inhalte der einzelnen Dokumente zeigt jedoch, daß dies durchaus Sinn macht, da diese interdisziplinär angelegt sind. Beispielsweise wurde eine Seite mit dem Titel „BMV: Behavioral Model of Visual Perception and Recognition“[4] von Yahoo der *(science, biology, neuroscience)*- Kategorie zugeordnet, die automatische Kategorisierung jedoch gewichtet die Kategorie *(science, computer science, computer vision)* am höchsten. Tatsächlich beschäftigt sich das Dokument inhaltlich mit der Untersuchung von Bewegungen des menschlichen Auges beim Fixieren von Objekten, um den Prozeß der Objekterkennung computergestützt zu simulieren, d. h. die hierin beschriebenen Forschungsergebnisse haben sowohl Implikationen für Neurobiologie als auch für die Entwicklung computergestützter Anwendungen.

Betrachtet man die Anzahl der Backlinks einzelner Dokumente und die Ranglistenposition, auf der die korrekte Kategorie eingeordnet wurde, zeigt sich eine klare Korrelation. Je mehr Backlinks ein Dokument aufweist, desto höher wird die korrekte Kategorie gewichtet. Es ist offensichtlich, daß je mehr Seiten auf eine bestimmtes Dokument verweisen, desto höher ist die Wahrscheinlichkeit, daß der korrekte Kategoriename mit einem Link auf das Dokument ko-zitiert wird. Dennoch konnte beobachtet werden, daß vereinzelt Dokumente mit niedrigen Backlink-Zahlen (weniger als 10) korrekt kategorisiert wurden.

[4] BMV: http://www.rybak-et-al.net/vnc.html [14. Nov. 2001]

In den beschriebenen Experimenten wurden nur Kategorien ausgewählt, die einen echten thematischen Bezug aufweisen. Darüber hinaus bietet Yahoo auch Kategorien an, in denen Links in der letzten Stufe nach organisatorischen Prinzipien abgelegt wurden, also z. B. On-line Journale, Home Pages von Instituten etc. Diese sind in Kategorien, wie z. B. *(science, biology, microbiology, institutes)* abgelegt. Um derartig klassifizierte URLs ebenfalls zu berücksichtigen, wurden in einem weiteren Experiment 253 URLs von Home-Pages verschiedener Informatik-Institute kategorisiert unter Verwendung von 11 Kategorien aus dem Bereich *computer science*. Dabei zeigte sich, daß meist die Kategorien am höchsten gewichtet wurden, für die das entsprechende Institut die höchste Autorität aufweist. 73% der Institute konnten jenen Bereichen zugeordnet werden, unter denen Yahoo die Institute zusätzlich abgelegt hat, also gemäß dem vorletzten Term einer Kategorie. Die Performanz ist also mit den Ergebnissen der vorherigen Experimente vergleichbar.

Abschließend läßt sich festhalten, daß die Experimente gezeigt haben, daß der Anteil korrekt kategorisierter Dokumente an Verfahren heranreicht, die auf Lerntechniken basieren. Damit erscheint die dargestellte Technik – auch wegen ihrer leichten Implementierbarkeit – dazu geeignet, existierende Kategorisierungs-Algorithmen für Web-Ressourcen, wie beispielsweise das Theseus-System [6], zu ergänzen.

6. Assoziationsbasierte Selektion von Suchservern

In diesem Kapitel wird ein neues Selektionsverfahren vorgestellt, das dazu geeignet ist, zu einer gegebenen Suchanfrage potentiell relevante Suchserver auf der Basis der gesammelten Metainformationen auszuwählen. In Abschnitt 6.3 wird das entwickelten Verfahren vorgestellt, wobei das hierzu benötigte Kontextwissen automatisiert durch Auswertung eines web-weiten Index gewonnen wird. Abschließend wird das Verfahren durch Experimente evaluiert.

6.1. Problembeschreibung

Im Verteilten Information Retrieval besteht bei der Selektion ein Hauptproblem darin, daß die Repräsentanten die Suchserver oft nur durch einige wenige Terme beschreiben. Von Seiten des Benutzers hingegen können allerdings beliebige Anfrageterme formuliert werden, so daß eine Auswahl der geeignetsten Server auf der Basis der vorhandenen Informationen kaum vorgenommen werden kann.

Das im Abschnitt 3.3.1.4 vorgestellte System QPilot versucht die Diskrepanz zwischen der freien Anfrageformulierung auf Benutzerseite und einer nur spärlich vorhandenen Suchserver-Beschreibung durch Anfrageerweiterung zu überwinden. Dabei werden dynamisch unter Verwendung einer generellen Suchmaschine aus dem WWW Terme extrahiert, die eine thematische Nähe zu der Anfrage aufweisen. Diese werden an die ursprüngliche Suchanfrage angehängt, was zwar die Wahrscheinlichkeit für Treffer von Anfragetermen in den Suchserver-Beschreibungen erhöht, aber noch keine ausreichende Sicherheit für Treffer gewährleistet.

Als geeigneter erscheint dahingegen die Analyse von Term-Ko-Zitierungen, die sich aus der Auswertung großer Dokumentensammlungen ergeben. Hieraus lassen sich Assoziationsgewichte zwischen Termen ableiten, die auf eine thematische Nähe hinweisen. Neben Anwendungen im Clustering und der automatischen Kategorisierung wird dies z. B. auch beim *Latent Semantic Indexing* angewendet, um eine Dimensions-

reduzierung im Vektorraummodell zu erreichen (siehe z. B. [61] und [48]).

In Abschnitt 5.2 wurde beschrieben, wie sich unter Verwendung einer generellen Suchmaschine Häufigkeiten aus dem WWW extrahieren lassen, um hieraus ein Assoziationsgewicht zwischen einem Dokument und einer Kategorie zu berechnen. Das Dokument wird dabei nur durch einen URL repräsentiert und die Kategorie durch eine Menge von Termen. Grundlage für die Berechnung bildet die Information über die Anzahl der WWW-Dokumente, in denen ein Link auf den URL und die einzelnen Kategorieterme ko-zitiert werden.

In diesem Abschnitt wird ein Verfahren vorgestellt, das diese Vorgehensweise auf das Problem der Selektion von Suchservern überträgt. Dabei wird eine Selektion auf Inhaltsebene durchgeführt, d. h. es wird für alle Suchserver angenommen, daß sie die gleichen Suchkosten aufweisen. Systemrelevante Faktoren wie Antwortzeit, Bandbreite etc. bleiben also unberücksichtigt.

Das Selektions-Verfahren berechnet ein Gewicht zwischen einem Anfrageterm und einem Suchserver auf der Basis von kurzen Suchserver-Beschreibungen. Dabei reicht es aus, einen Suchserver durch eine geringe Anzahl von drei bis fünf Deskriptor-Termen zu repräsentieren, die in den FC-Metadatenbeschreibung des Suchservers gespeichert werden. Die Berechnung des Gewichtes erfolgt auf der Basis von Dokumenthäufigkeiten, die unter Auswertung einer generellen Suchmaschine determiniert werden.

Zunächst erfolgt die Entwicklung und Motivation einer Funktion zur Bestimmung der Assoziation zwischen zwei Termen. Dieses bildet die Grundlage für das Selektionsverfahren, das anschließend anhand von Experimenten in einer geeigneten Testumgebung evaluiert wird.

6.2. Berechnung von Assoziationsgewichten

6.2.1. Spezifikation der Anforderungen

Zu zwei gegebenen Termen t_1 und t_2 sowie einer Dokumentsammlung D soll ein Assoziationsgewicht für t_1 und t_2 berechnet werden.

Gesucht ist somit eine Funktion $f_D : \mathbb{N}^+ \times \mathbb{N}^+ \times \mathbb{N}^+ \times \mathbb{N}_0^+ \to \mathbb{R}_0^+$ mit

$$f_D(N, df_1, df_2, f_{1,2}) = g_{1,2}$$

wobei gilt:

$N \mid N > 0$ (Gesamtanzahl der Dokumente in D),
$df_1 \mid 0 < df_1 \leq N$ (Dokumenthäufigkeit des Terms t_1 in D),
$df_2 \mid 0 < df_2 \leq N$ (Dokumenthäufigkeit des Terms t_2 in D),
$f_{1,2} \mid 0 \leq f_{1,2} \leq min(df_1, df_2)$ (Häufigkeit, mit der die Terme t_1 und t_2 in Dokumenten aus D ko-zitiert werden),
$g_{1,2} \mid g_{1,2} \geq 0$ (Assoziationsgewicht für t_1 und t_2),

die für beliebige aber fest gewählte N,df_1 und df_2 in $f_{1,2}$ streng monoton steigt, d. h. mit steigendem $f_{1,2}$-Wert – also steigenden Ko-Zitierungen – steigt auch das Assoziationsgewicht $g_{1,2}$, und, die für beliebige aber fest gewählte N,df_1 und $f_{1,2}$ bzw. N,df_2 und $f_{1,2}$ streng monoton fällt, d. h. mit sinkenden Werten für df_1 bzw. df_2 Werten sinkt auch das Assoziationsgewicht $g_{1,2}$.

6.2.2. Das $-2log\lambda$-Maß

Ein häufig verwendetes Maß zur Berechnung der Assoziation von Termen in Texten ist das von Dunning entwickelte $-2log\lambda$-Maß [39]. Diese wird beispielsweise in [92] verwendet, um Terme, die in einer gegebenen Dokumentsammlung vorkommen, auf ein kontrolliertes Vokabular abzubilden. Im folgenden soll geprüft werden, inwieweit das $-2log\lambda$-Maß die in Abschnitt 6.2.1 genannten Anforderungen erfüllt.

Der Berechnung des Assoziationsgewichtes liegt die folgende allgemeine Annahme zugrunde: Wenn sich beobachten läßt, daß zwei Terme mit einer größeren Häufigkeit zusammen auftreten als dies durch eine zufällige Verteilung zu erwarten wäre, sind diese offensichtlich miteinander assoziiert.

Das Assoziationsgewicht des $-2log\lambda$-Maßes wird durch einen log-Likelihood-Quotienten bestimmt, wobei das Auftreten von Termen durch eine Binominalverteilung modelliert wird [39]:

$$-2log\lambda = 2[logL(p_1, k_1, n_1) + logL(p_2, k_2, n_2) - logL(p, k_1, n_1) - logL(p, k_2, n_2)]$$

wobei gilt:

$$log\, L(p, k, n) = k\, log\, p + (n - k) log\, (1 - p)$$

sowie $p_1 = \frac{k_1}{n_1}$, $p_2 = \frac{k_2}{n_2}$ und $p = \frac{k_1 + k_2}{n_1 + n_2}$.
Die Parameter k_1,k_2,n_1 und n_2 können durch die in Abschnitt 6.2.1 spezifizierten Häufigkeiten ausgedrückt werden:

$k_1 = f_{1,2}$,
$k_2 = df_1 - f_{1,2}$,
$n_1 = df_2$,
$n_2 = N - df_2$.

Damit kann das $-2log\lambda$-Maß als Funktion über den Parametern N, df_1, df_2 und $f_{1,2}$ spezifiziert werden.

Man kann nun zeigen, daß die erste Ableitung nach $f_{1,2}$ der zugehörigen reellen Funktion $r_D : I\!R \mapsto I\!R$ mit konstanten Werten für N, df_1 und df_2 an der Stelle $\frac{df_1 df_2}{N}$ eine Nullstelle aufweist und daß an dieser Stelle ein Minimum vorliegt.

Damit erfüllt die zu dem $-2log\lambda$ gehörige Funktion nicht die in Abschnitt 6.2.1 geforderte Monotonie-Bedingung. Die Funktion ist nur im Intervall $[\frac{df_1 df_2}{N}, \infty)$ streng monoton steigend und liefert für das Intervall $[0, \frac{df_1 df_2}{N})$ keine aussagekräftigen Werte.

Dieser Zusammenhang wird durch die Abbildung 6.1 verdeutlicht. Diese zeigt den Funktionsgraphen mit $f_{1,2}$ als Parameter, wobei df_1, df_2 und N jeweils mit konstanten Werten versehen wurden ($df_1 = 100.000$, $df_2 = 1.000.000$ und $N = 100.000.000$).

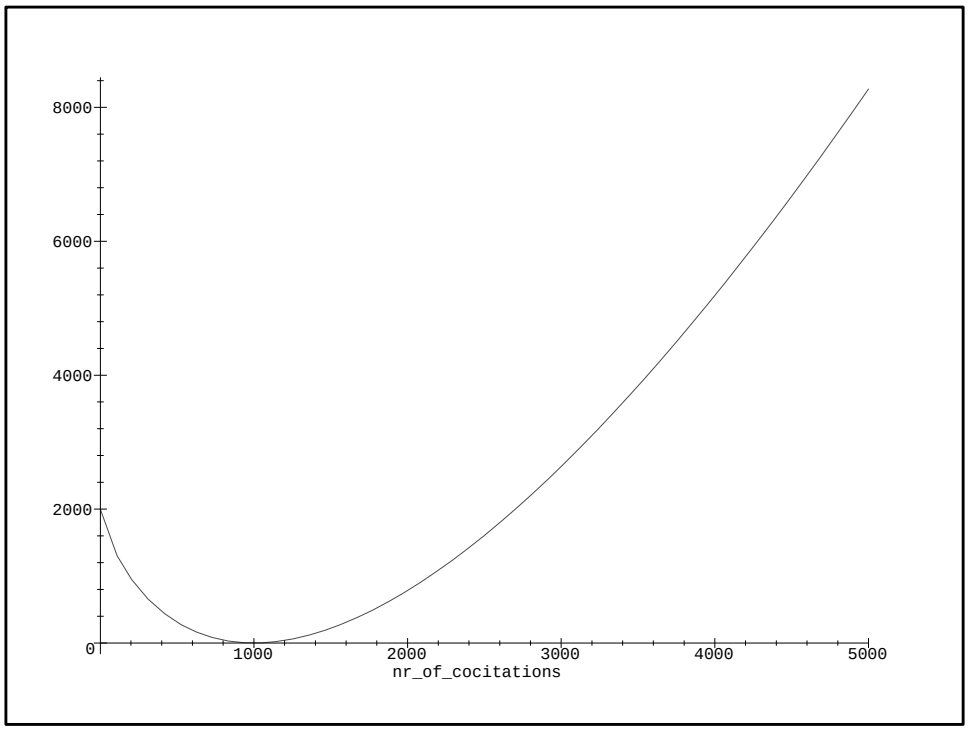

Abbildung 6.1.: Funktionsgraph für $-2log\lambda$-Maß

6.2.3. Ein idf-basiertes Maß zur Berechnung eines Assoziationsgewichtes

Aufgrund der gezeigten Schwächen des $-2log\lambda$-Maßes erfolgt in diesem Abschnitt die Definition einer einfachen Funktion zur Berechnung eines Assoziationsgewichtes zwischen zwei gegebenen Termen t_1 und t_2. Die neue Funktion hat sich in der Praxis bewährt und erfüllt die in Abschnitt 6.2.1 formulierten Anforderungen. Sie bildet die Grundlage für das im folgenden Abschnitt 6.3 Selektionsverfahren.

Es gilt: Je häufiger die einzelnen Terme in der betrachteten Dokumentsammlung auftauchen, desto niedriger muß das berechnete Assoziationsgewicht ausfallen. Diesem Umstand wird durch die Verwendung der inversen Dokumenthäufigkeit Rechnung getragen. Das idf-Gewicht wird für beide Terme bestimmt und jeweils mit der Wahrscheinlichkeit für das Auftreten von Ko-Zitierungen multipliziert. Es ergibt sich die folgende Funktion:

$$assoc(N, df_1, df_2, f_{1,2}) = (log\frac{N}{df_1}) * \frac{f_{1,2}}{df_1} * (log\frac{N}{df_2}) * \frac{f_{1,2}}{df_2}$$

Dabei ist:

N: Gesamtanzahl der Dokumente in der betrachteten Dokumentsammlung D,
df_1: Dokumenthäufigkeit des Terms t_1 in D,
df_2: Dokumenthäufigkeit des Terms t_2 in D,
$f_{1,2}$: Häufigkeit, mit der die Terme t_1 und t_2 in Dokumenten aus D ko-zitiert werden.

$(log\frac{N}{df_i})$ bestimmt das idf-Gewicht eines Terms t_i gemessen über die N Dokumente der betrachteten Dokumentsammlung.

$\frac{f_{1,2}}{df_1}$ (bzw. $\frac{f_{1,2}}{df_2}$) bestimmt die bedingte Wahrscheinlichkeit dafür, daß t_1 mit t_2 zusammen zitiert werden, unter der Voraussetzung, daß t_1 (bzw. t_2) bereits zitiert wurde.

Man kann leicht zeigen, daß die in Abschnitt 6.2.1 genannten Monotonie-Bedingungen durch die $assoc$-Funktion erfüllt sind.

6.3. Ein Verfahren zur Selektion von Suchservern im WWW

Die $assoc$-Funktion bildet die Grundlage für ein Selektionsverfahren für Web-basierte Suchserver. Wie bereits für das in Abschnitt 5.2 vorgestellte Kategorisierungsverfahren erfolgt die Bestimmung der Häufigkeiten df_1,df_2 und $f_{1,2}$ unter Auswertung einer

generellen Suchmaschine.

Durch die Verwendung eines Assoziationsgewichtes reicht es aus, jeden Suchserver für die Selektion nur durch eine geringe Anzahl von Deskriptor-Termen zu repräsentieren. Hierfür können Terme verwendet werden, die zuvor automatisch aus dem sichtbaren Teil des WWW extrahiert und in den einzelnen FC-Beschreibungen der Suchserver hinterlegt wurden. In Abschnitt 4.4.1 wurden bereits verschiedene Strategien der Informationsgewinnung diskutiert.

6.3.1. Bestimmung von Deskriptortermen

Für die in Abschnitt 6.4 beschriebenen Experimente reicht es aus, jeden Suchserver nur durch eine Anzahl von drei bis fünf Deskriptor-Termen zu repräsentieren. Diese können der aus der FC-Beschreibung der Suchserver extrahiert werden.

Hierzu lassen sich beispielsweise die folgenden FC-Informationen verwenden:

- häufige verwendete Terme aus den Anktertexten eingehender Link zur Startseite eines Suchservers (FC-Feld *backlinkpage_terms*),
- Terme aus dem Titel und der Startseite des Suchservers (FC-Feld *title*),
- Bezeichner von Kategorien aus existierenden frei verfügbaren Web-Katalogen unter der der Suchserver bereits eingeordnet wurde, bzw. Bezeichner von Kategorien, die durch automatische Verfahren (siehe Abschnitt 5.2) dem Suchserver zugeordnet werden konnten (FC-Feld: *category_terms*).

Beispielsweise kann der Suchserver s mit dem Titel „Marx/Engels Search“[1] – er enthält Schriften verschiedener bekannter Kommunisten – durch die folgende Menge von Deskriptortermen repräsentiert werden: $T_s = \{marx, engels, history\}$.

6.3.2. Beschreibung des Verfahrens

Seien ein Anfrageterm q und ein Suchserver s gegeben, der durch eine Menge $T_s = \{t_1, t_2, \ldots, t_n\}$ von n Deskriptor-Termen beschrieben ist. Dann wird das Gesamtgewicht zwischen q und T_s durch Aufsummieren der Einzelgewichte zwischen q und jedem einzelnen Deskriptorterm $t_i \in T_s$ unter Verwendung der folgenden Formel definiert:

[1] Marx/Engels Search: http://search.marxists.org [14. Nov. 2001]

$$weight(q, T_s) = \sum_{i=1}^{n} assoc(N, df_i, df_q, f_{q,i})$$

wobei gilt:

N: geschätzte Gesamtanzahl der von AltaVista indexierten Web-Seiten (Im Juli 2000 etwa 350 Mio),
df_q: Anzahl der Web-Seiten, die den Anfrageterm q enthalten, oder 1, falls keine solche Web-Seite existiert,
df_i: Anzahl der Web-Seiten, die den Term t_i enthalten, oder 1, falls keine solche Web-Seite existiert,
$f_{q,i}$: Anzahl der Web-Seiten, die sowohl den Anfrageterm q als auch den Term t_i enthalten.

Tendenziell gilt: Je eindeutiger eine durch eine Folge von Termen gegebene Suchanfrage ein bestimmtes Konzept charakterisiert, desto spezifischer kann sie einem bestimmten Suchserver zugeordnet werden. Beispielsweise kann die Anfrage *q="rosa luxemburg"* leicht dem „Marx/Engels Search" Suchserver zugeordnet werden, denn die meisten Web-Dokumente, die die Zeichenkette der Anfrage enthalten, beziehen sich auf die berühmte deutsche Kommunistin und werden deshalb auch oft mit den Suchserver-Deskriptoren *marx* und *engels* ko-zitiert, also mit Termen, die ebenfalls häufig im Kontext *kommunismus* verwendet werden – ohne daß der Begriff Kommunismus explizit als Deskriptor auftauchen muß. Dementsprechend wird ein hohes Assoziationsgewicht zwischen Anfrage q und Suchserver s berechnet.

Wird z. B. nur der Term *q=luxemburg* als Suchanfrage verwendet, so muß berücksichtigt werden, daß die meisten Zitierungen von *luxemburg* im Web sich auf den europäischen Staat und nicht auf die Kommunistin beziehen, allerdings wird dies über einen entsprechend niedrigeren idf-Wert für *luxemburg* zum Teil ausgeglichen (im Februar 2000: $idf(\text{luxemburg}) = 7.51$ und $idf(\text{"rosa luxemburg"}) = 10.18$).

Durch die Analyse der Ko-Zitierungen von Anfragetermen und Deskriptoren über eine generelle Web-Suchmaschine, werden jeweils die dominantesten Bedeutungskontexte berücksichtigt, in denen die verschiedenen Terme im WWW verwendet werden, d. h. stünde zusätzlich ein Suchserver mit Dokumenten über die Europäische Union zur Auswahl, so würde für diesen wahrscheinlich ein höheres Assoziationsgewicht für den Term *luxemburg* berechnet werden als für den „Marx/Engels Search" Suchserver.

6.4. Experimente

Um die Anwendbarkeit des beschriebenen Verfahrens in einer realistischen Web-Umgebung aufzuzeigen, wurden verschiedene Experimente durchgeführt. Hierzu wurde zunächst eine geeignete Experimentumgebung von spezialisierten Suchservern aus dem WWW zusammengestellt.

6.4.1. Experimentumgebung

Bei der Zusammenstellung der Experimentumgebung standen die folgenden Anforderungen im Vordergrund: Der thematische Bezug der einzelnen Suchserver soll deutlich erkennbar sein und sich von den anderen Suchservern abgrenzen. Jedoch sind partielle thematische Überlappungen der einzelnen Suchserver bis zu einem gewissen Grad erwünscht, um auch Anfragen berücksichtigen zu können, für die mehrere relevante Suchserver existieren. Hinzu kommt die Anforderung, eine Experimentierumgebung zu schaffen, die Rückschlüsse auf das Verhalten der Selektionsmethode in realistische Web-Szenarien zuläßt.

Diese Überlegungen führten zu der Erstellung einer Experimentumgebung, die sich ausschließlich aus realen Suchservern im WWW mit verschiedenen historischen Inhalten zusammensetzt. Dabei besitzt jeder der Suchserver einen individuellen thematischen Schwerpunkt, z. B. enthält er nur Dokumente über eine bestimmte Epoche oder ein historisches Ereignis. Im einzelnen handelt es sich um:

- 5 Suchserver mit Dokumenten aus verschiedenen Epochen:
  - Antikes Ägypten
  - Antikes Griechenland
  - Römisches Reich
  - Mittelalter
  - Viktorianische Zeit
- 5 Suchserver mit Dokumenten über spezielle historische Ereignisse oder Persönlichkeiten:
  - George Washington,
  - Napoleon,
  - Die deutsche Revolution von 1848/1849,
  - Der amerikanische Bürgerkrieg,

  - Der Erste Weltkrieg.

- 2 Suchserver mit Dokumenten, die nicht an eine bestimmte Epoche gebunden sind:

  - Schriften von und über populäre Kommunisten und Philosophen,
  - Geschichte der Medizin und der Naturwissenschaften.

6.4.2. Vergleich durch Probeanfragen

Das in Abschnitt 6.3 beschriebene Verfahren berechnet ein Assoziationsgewicht zwischen einer Suchanfrage und einem Suchserver, ohne auf dessen lokal verborgenen Daten zuzugreifen. Vielmehr werden ausschließlich öffentlich zugängliche Daten wie Linktexte oder existierende Kategorisierungen zur Berechnung des Assoziationsgewichtes ausgenutzt. Um die Güte des Selektions-Verfahrens beurteilen zu können, erscheint es deshalb notwendig, einen Vergleich mit einer Gewichtung anzustreben, die auf der Ausnutzung von lokalen Informationen basiert. Eine solche lokale Information stellt die Dokumenthäufigkeit $df_{i,j}$ dar, also die Anzahl der Treffer, die eine Suchanfrage q_i in einem Suchserver s_j erzeugt.

Deshalb wurde eine Liste Q von Suchbegriffen erstellt und es wurden hiermit automatisierte Probeanfragen an jedem der Suchserver der Experimentumgebung durchgeführt. Bei den Anfragen handelte es sich im wesentlichen um die Namen von historischen Personen, Orten, Ereignissen etc. Da mehrere der Suchserver einem übergeordneten thematischen Kontext zugeordnet werden können (z. B. Geschichte des 19. Jahrhunderts, Amerikanische Geschichte oder Geschichte des Altertums), ist zu erwarten, daß die Mehrzahl der Suchanfragen Treffer in mehreren Suchservern gleichzeitig erzeugt.

Für jeden der 12 Suchserver in der Suchservermenge S wurde ein individuelles Wrapperprogramm implementiert, das die Anfrage in die Abfragesprache des Suchservers übersetzt, ein Anfrage durchführt und von der zurückgelieferten Ergebnisseite die Größe der erzeugten Treffermenge extrahiert. Für jede Anfrage $q_i \in Q$ und jeden Suchserver $s_j \in S$ wurde eine Dokumenthäufigkeit $df_{i,j}$ ermittelt und protokolliert.

6.4.3. Normierung der Dokumenthäufigkeiten

Als Vergleichskriterium sind die ermittelten Dokumenthäufigkeiten noch nicht ausreichend. Eine solche Vorgehensweise vernachlässigt lokale Informationen wie die Gesamtanzahl der über einen Suchserver verfügbaren Dokumente oder die Gesamtan-

zahl der enthaltenen verschiedenen Terme,[2] Diese können von Suchserver zu Suchserver stark variieren und beeinflussen die Größe der einzelnen Treffermengen, was einen Vergleich der Treffermengen untereinander erschwert. Beispielsweise erzeugten die Probeanfragen am „Marx/Engels Search"-Suchserver oft sehr große Treffermengen von mehreren 100 gefundenen Dokumenten, wohingegenen die höchste Dokumenthäufigkeit, die für den 1848 Suchserver festgestellt werden konnte nur 9 betrug.

Dennoch ist es notwendig zu Vergleichszwecken eine Normierung der verschiedenen Dokumenthäufigkeiten vorzunehmen. Allerdings lassen sich private Metadaten wie die Gesamtanzahl der verfügbaren Dokumente über das gezielte Abfragen einer öffentlichen Web-Schnittstelle nur relativ ungenau annähern (siehe Abschnitt 4.4.2). Als Vergleichsmaß wird deshalb eine vereinfachte CORI-Gewichtung herangezogen. In [35] wird die folgende CORI-Variante verwendet, um die Wahrscheinlichkeit $p(q_i \mid s_j)$ zu berechnen, daß ein Suchserver s_j aus einer Menge von Suchservern S (also $s_i \in S$) Dokumente enthält, die für die Anfrage q_i relevant sind.

$$T = \frac{log(df_{i,j} + 0.5)}{log(df_j^{max} + 1.0)}$$

$$I = \frac{log(\frac{|S|+0.5}{sf_i})}{log(\mid S \mid +1.0)}$$

$$p(q_i \mid s_j) = 0.4 + 0.6 \cdot T \cdot I$$

Dabei bezeichnet in [35] df_j^{max} die Dokumenthäufigkeit des häufigsten Terms im Suchserver s_j. Da dieser Wert nicht ermittelt werden kann, ohne den gesamten Volltext eines Suchservers zu kennen, wird im folgenden df_j^{max} als das Maximum aller $df_{i,j}$ über die Menge aller Suchanfragen $q_i \in Q$ bestimmt, wobei Q die Menge der im Experiment berücksichtigten Suchanfragen bezeichnet. Es gilt also $df_j^{max} = max(df_{i,j} \mid \forall q_i \in Q)$. Die Serverhäufigkeit sf_i für eine Suchanfrage q_i ergibt sich aus der Anzahl der Suchserver $s_j \in S$ für die mindestens ein Treffer erzeugt wird, also $sf_i = size\{s_j \in S \mid df_{i,j} > 0\}$. Man beachte, daß für die Experimente gilt $\mid S \mid = 12$.

6.4.4. Bewertung der Selektion

Um die Güte des assoziationsgesteuerten Selektionsverfahrens zu bewerten wird ein Vergleich mit den automatisiert erzeugten und CORI-normierten Dokumenthäufigkeiten vorgenommen. Dabei wird folgendermaßen vorgegangen: Für jede Suchanfrage $q_i \in Q$ werden je zwei Ranglisten erzeugt. In der ersten Rangliste

[2] Also Informationen, die z. B. in die Berechnung der CORI- und GlOSS-Gewichte miteinbezogen werden.

R_i^{assoc} liegen die Suchserver gewichtet nach dem assoziationsbasierten Verfahren vor, wohingegen in der zweiten Rangliste R_i^{cori} die Suchserver gemäß der berechneten CORI-Wahrscheinlichkeit sortiert werden.

Die beiden zu einer Suchanfrage q_i gehörigen Ranglisten R_i^{assoc} und R_i^{cori} werden jeweils miteinander verglichen. Hierdurch soll überprüft werden, inwieweit die assoziationsbasierte Gewichtung von Suchservern geeignet ist, eine Gewichtung anzunähern, die unter Verwendung von lokalen Daten erzeugt wurde. Es sei an dieser Stelle nochmals daran erinnert, daß es für das assoziationsgesteuerte Gewichtungsverfahren nicht erforderlich ist, explizit lokale Informationen über die Web-Schnittstelle zu extrahieren, d. h. man erreicht zum einen eine Verminderung des Programmieraufwandes bei der Integration neuer Suchserver (z. B. in eine Metasuchmaschinen-Architektur) und zum anderen eine Reduzierung der Netzlast durch die Vermeidung von Probe-Anfragen.

In Anlehnung an die beiden im Information Retrieval gebräuchlichen Bewertungsmetriken Abdeckung und Präzision (siehe Abschnitt 2.1.4) erfolgt die Bewertung der Experimentergebnisse gemäß dem Grad der Abdeckung von und der Präzision von R_i^{cori} und R_i^{assoc}. Zur Bestimmung der Abdeckung wird überprüft, wieviele Ranglistenpositionen von R_i^{assoc} berücksichtigt werden müssen, um alle Suchserver zu zu erhalten, die mindestens ein relevantes Dokument enthalten. Für die Präszision dahingegen wird überprüft, wieviele Ranglistenpositionen von R_i^{assoc} berücksichtigt werden müssen, um den relevantesten Suchserver zu erhalten, also jenen, für den normiert nach dem CORI-Gewicht die größte Resultatmenge festgestellt wurde.

6.4.4.1. Grad der Abdeckung von R_i^{cori} und R_i^{assoc}

Für alle Anfragen q_i wird untersucht, wieviele der durch R_i^{cori} am höchsten gewichteten Suchserver sich unter den ersten d Elementen der Rangliste R_i^{assoc} befinden. Die Suchanfragen q_i werden hierzu in drei Gruppen eingeteilt, die getrennt voneinander untersucht werden. Die Einteilung ergibt sich aus der Serverhäufigkeit (server frequency) sf_i ($1 \leq sf_i \leq 12$) einer Suchanfrage q_i:

1. Stark spezifische Anfragen ($\{q_i \mid sf_i = 1\}$): Eine Anfrage erzeugt einen Treffer in jeweils nur einem der 12 Suchserver, d. h. sie besitzt eine hohe Spezifizität für einen bestimmten Suchserver.

2. Mittelspezifische Anfragen ($\{q_i \mid sf_i = 2\}$): Eine Anfrage erzeugt einen Treffer in zwei der 12 Suchserver.

3. Wenig spezifische Anfragen ($\{q_i \mid 3 \leq sf_i \leq 12\}$): Eine Anfrage erzeugt Treffer in mindestens drei der Suchserver.

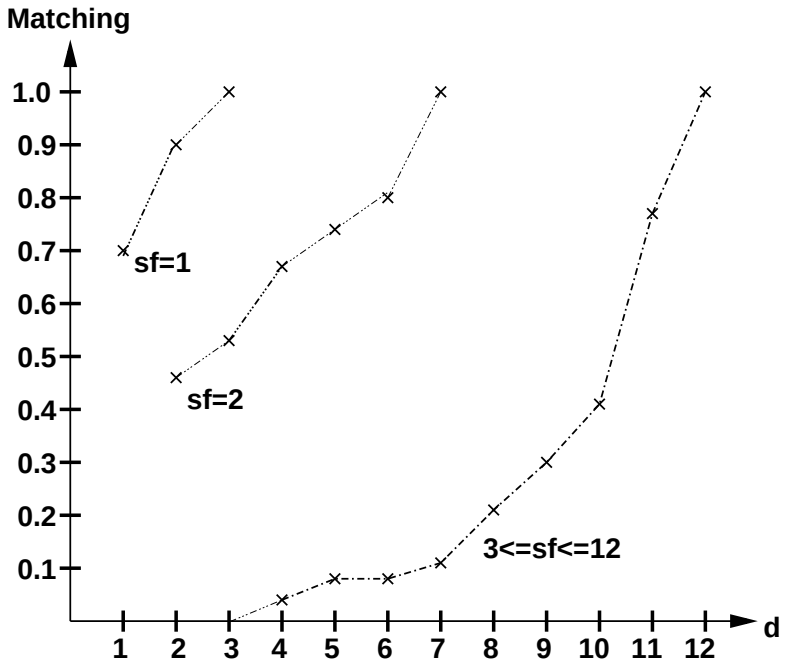

Abbildung 6.2.: Abdeckung von R_i^{cori} und R_i^{assoc}

Die Abbildung 6.2 zeigt – getrennt nach den drei Gruppen – den Anteil der Anfragen q_i, deren treffererzeugende Server sich auch unter den ersten d Positionen von R_i^{assoc} befinden.

Die besten Ergebnisse wurden für stark spezifische Anfragen erzielt ($sf = 1$). 70% dieser Anfragen gewichteten den einzigen Suchserver, in dem Treffer erzeugt wurden, an der ersten Position ($d = 1$) von R_i^{assoc}. Für 20% dieser Anfragen befand sich dieser Suchserver an der zweiten Position, d. h. für 90% aller Anfragen mit $sf = 1$ fand sich der treffererzeugende Suchserver unter den beiden am höchsten gewichteten Positionen ($d = 2$). Alle der Anfragen dieser Gruppe gewichteten den treffererzeugenden Suchserver unter den ersten drei Top-Positionen $d = 3$.

Suchanfragen mit mittlerer Spezifizität, also solche, die Treffer in zwei der Suchserver erzeugen ($sf = 2$), lieferten auch noch gute Resultate: Für 46% dieser Suchanfrage befanden sich die ersten beiden am höchsten gewichteten Suchserver der Rangliste R_i^{cori} auch unter den ersten beiden Positionen von R_i^{assoc} wieder. Für zwei Drittel dieser Suchanfragen befanden sich die beiden treffererzeugenden Suchserver jeweils noch unter den ersten 4 ($d = 4$) Positionen von R_i^{assoc} und für 80% unter den ersten 6 Positionen ($d = 6$).

Je unspezifischer eine Suchanfrage einem bestimmten Suchserver zugeordnet werden kann, desto stärker weichen R_i^{assoc} und R_i^{cori} voneinander ab. Suchanfragen der dritten Gruppe ($\{q_i \mid 2 \leq sf_i \leq 12\}$) liefern nur wenig zufriedenstellende Ergebnisse.

Betrachtet man allerdings nur den relevantesten Suchserver für eine Suchanfrage, fällt das Ergebnis wesentlich besser aus (siehe hierzu den folgenden Abschnitt).

6.4.4.2. Präzision von R_i^{cori} und R_i^{assoc}

Es liegt die Vermutung nahe, daß der Suchserver, der in der Rangliste R_i^{cori} am höchsten gewichtet wurde, für die Suchanfrage q_i am relevantesten ist (Dieser wird im folgenden als s_{rel} bezeichnet). Es wird nun überprüft, an welcher Position in der Rangliste R_i^{assoc} der Suchserver s_{rel} berechnet wurde.

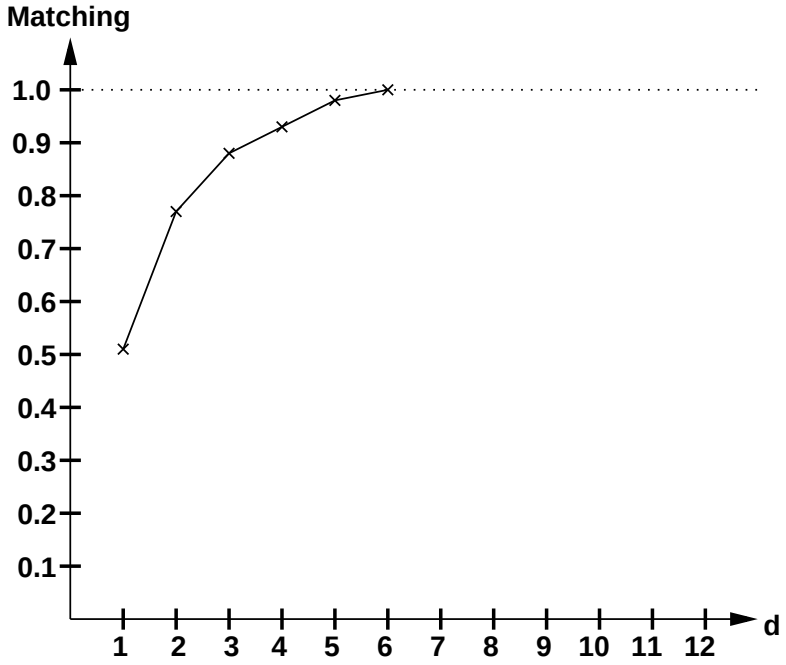

Abbildung 6.3.: Präzision von R_i^{cori} und R_i^{assoc} über alle $q_i \in Q$

Das Diagramm in Abbildung 6.3 zeigt für alle Anfragen $q_i \in Q$, an welcher Stelle sich s_{rel} in den einzelnen R_i^{assoc} befindet unter Berücksichtigung der ersten d Positionen. Für 52,6% aller Suchanfragen befindet sich s_{rel} an der ersten Position. Werden die ersten drei Positionen ($d = 3$) berücksichtigt, befindet sich s_{rel} bereits für 88,7% aller Suchanfragen darunter.

Das Diagramm 6.4 zeigt diesen Sachverhalt zusätzlich aufgeschlüsselt für die Gruppen der Suchanfragen mit mittlerer und niedriger Spezifizität. Für die Suchanfragen mit hoher Spezifizität entspricht dies der Kurve für $sf = 1$ im Diagramm 6.2, da hierfür nur eine einzige relevante Kollektion existiert.

Für 53,3% aller Suchanfragen mittlerer Spezifizität befindet sich s_{rel} an der ersten Position und für 93,9% unter den ersten drei Positionen. Bei den Suchanfragen niedriger

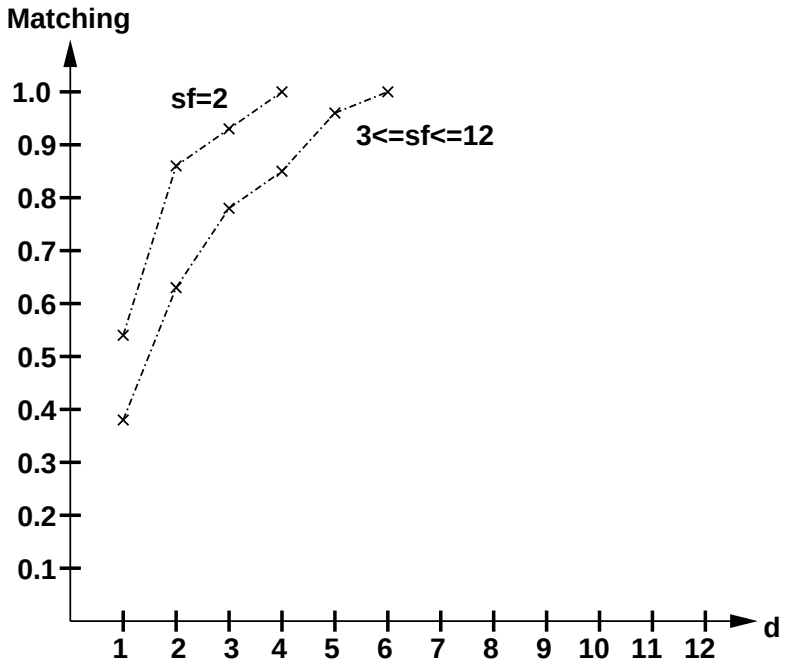

Abbildung 6.4.: Präzision von R_i^{cori} und R_i^{assoc} über alle $q_i \in Q$ mit $sf_i > 1$

Spezifizität befindet sich s_{rel} in 37% aller Fälle an erster Position und für 96,3% unter den ersten 5.

Zusammenfassend zeigt sich also, daß in der Experimentumgebung durch das vorgestellte Verfahren eine erfolgreiche Selektion vorgenommen werden kann, ohne daß man auf das Vorhandensein von privaten Metadaten bzw. auf eine explizite Kooperation von Seiten der Suchserver angewiesen ist. Die für die Selektion notwendigen Suchserver-beschreibenden Daten konnten ausschließlich unter Auswertung des Sichtbaren Webs gewonnen werden.

7. QUEST - Eine Metasuchmaschinen-Architektur für spezialisierte Web-Kollektionen

In diesem Abschnitt erfolgt die Darstellung von QUEST (QUErying Specialized collecTions on the Web) – einer Metasuchmaschine für spezialisierte, Web-basierte, nicht-kooperative Suchserver [58].

QUEST stellt die Zusammenführung der in den vorherigen Abschnitten beschriebenen Konzepten und Komponenten in eine integrierte Systemarchitektur dar. In Abschnitt 7.1 wird zunächst ein Überblick über die Gesamtarchitektur von QUEST gegeben. Dabei werden kurz die Einzelkomponenten vorgestellt sowie deren Zusammenwirken beschrieben. In den Abschnitten 7.2 bis 7.5 werden die Implementierungsaspekte der einzelnen Systemkomponenten behandelt. Abschließend wird in Abschnitt 7.6 exemplarisch eine Anfrage an QUEST dargestellt.

7.1. QUEST Architektur

7.1.1. Überblick

QUEST ist eine Metasuchmaschine für Ressourcen des Unsichtbaren Webs, die es ermöglicht eine gezielte Selektion potentiell relevanter Suchserver in Abhängigkeit von einer gestellten Benutzeranfrage vorzunehmen. Die Selektionsentscheidung erfolgt auf der Basis von Metadaten, die zuvor für jeden Suchserver automatisiert generiert wurden. Des weiteren werden die Suchoptionen berücksichtigt, die von den einzelnen Suchservern unterstützt werden. Die selektierten Suchserver werden lokal abgefragt und eine integrierte Ergebnisliste an den Benutzer zurückgeliefert.

Die Architektur von QUEST setzt sich aus den folgenden Einzelkomponenten zusammen:

- *Kategorisierungskomponente:* Kategorisierung von Suchservern,

- *Metadatenkollektor:* Automatisierte Erzeugung von Metadaten durch Auswertung des Sichtbaren Webs,

- *Assoziations-Server:* Berechnung von Assoziationsgewichten zwischen Anfrage und Suchserver,

- *DBMS-Komponente:* Ermitteln und Abspeichern von Häufigkeiten durch Abfragen einer generellen Suchmaschine.

- *Taxonomie-Server:* Verwaltung einer Taxonomie von Suchkonzepten,

- *Mediator:* Ablaufkoordination und Initiierung der Suchanfragen an den Suchservern,

- *Wrapper:* Durchführung der Anfragen an den Suchservern, Auswertung der Resultatlisten und Rückgabe der Ergebnisse.

Eine Übersicht über die Architektur von QUEST findet sich in Abbildung 7.1.

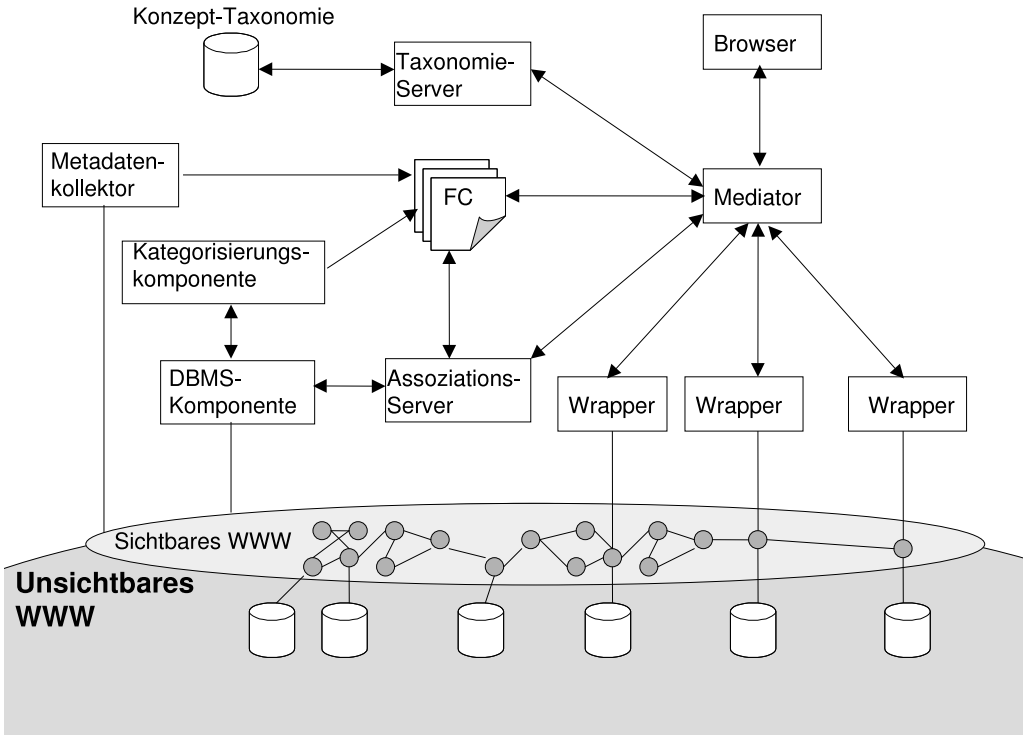

Abbildung 7.1.: Architektur von QUEST

7.1.2. Erzeugung von Suchserver-Repräsentanten

Damit Anfragen im QUEST-System bearbeitet werden können, ist es zuvor notwendig, eine Beschreibung für jeden Suchserver zu erzeugen. In QUEST ist jeder Suchserver durch eine FC-Metadatenbeschreibung repräsentiert. Eine ausführliche Diskussion der einzelnen Felder des FC erfolgte in Abschnitt 4.3.

Die Erzeugung von FC-Suchserver-Repräsentanten wird durch automatisierte Verfahren unterstützt, die systematisch Daten aus dem Sichtbaren Web extrahieren. Dies wird von zweien der in Abbildung 7.1 dargestellten Komponenten erbracht: dem Metadatenkollektor und der Kategorisierungskomponente. Der Metadatenkollektor sammelt Daten unter Anwendung der in Abschnitt 4.4.1 dargestellten Strategien. Die Kategorisierungskomponente ordnet jedem Suchserver eine Menge von Kategorien aus einer vorgegebenen Menge von Yahoo-Kategorien zu. Diese werden im FC-Feld *category_terms* abgespeichert. Hierzu wird das in Abschnitt 5.2 dargestellte Verfahren angewendet. Die durch den Metadatenkollektor und die Kategorisierungskomponente erzeugten Daten können in regelmäßgien Abständen aktualisiert werden.

7.1.3. Anfragebearbeitung in QUEST

Liegen Suchserver-Beschreibungen für alle in QUEST integrierten Suchserver vor, so können Anfragen unter Anwendung der folgenden Arbeitsschritte durchgeführt werden:

1. Anfrageformulierung,
2. Selektion der relevantesten Suchserver,
3. Bestimmung der unterstützten Suchfelder,
4. Durchführen von lokalen Anfragen an den Suchservern und
5. Erzeugung einer integrierten Resultatliste.

Anfrageformulierung Ein Benutzer spezifiziert in einem Web-Browser über ein HTML-Formular eine Suchanfrage. Für jeden spezifizierten Term kann ein Suchkonzept aus einer vorgebenen Taxonomie von Suchkonzepten mitangegeben werden (siehe hierzu das Abschnitt 4.5.4 dargestellte Beispiel) z. B. *title:"artificial intelligence"*, *character:"luke skywalker"* oder *author:"marvin minsky"*. Dabei können mehrere Terme über die boolesche Operation *AND* miteinander verknüpft werden. Wird vom Benutzer kein Suchkonzept mitangegeben, wird vom System das generellste Suchkonzept *thing* vorgegeben.

Selektion der relevantesten Suchserver Anschließend wird die Suchanfrage dem Mediator übergeben. Dessen primäre Aufgabe besteht in der Vorbereitung und Initiierung von Abfragen an den zugrundeliegenden Suchservern. Zuerst kontaktiert der Mediator hierzu den Assoziations-Server. Dieser liefert eine gewichtete Liste der in QUEST integrierten Suchserver zurück, wobei ein einzelnes Gewicht die geschätzte Relevanz des jeweiligen Suchservers zur vorgegebenen Suchanfrage bestimmt. Die Einzel-Gewichte zwischen Anfrage und Suchserver werden durch die Anzahl der im WWW meßbaren Ko-Zitierungen zwischen einer Suchanfrage q und Suchserver-spezifischen Termen T_s bestimmt, die in der FC-Metadatenbeschreibung hinterlegt sind. Zur Anwendung kommt hier die in Abschnitt 6.3.2 beschriebene Funktion $weight(q, T_s)$. Die für die Berechnung notwendigen Dokumenthäufigkeiten der Terme und die Anzahl der Ko-Zitierungen werden vom Assoziations-Server aus der DBMS-Komponente ausgelesen. Eine Anfrage soll nur an jene Suchserver weitergeleitet werden, für die die höchsten Assoziationsgewichte berechnet wurden.

Bestimmung der unterstützten Suchfelder Im nächsten Schritt bestimmt der Mediator, welche Suchfelder der selektierten Suchserver bei der Anfrage verwendet werden sollen. Dabei gilt, daß nur solche Suchfelder angefragt werden sollen, die spezifischer oder gleich spezifisch dem spezifizierten Suchkonzept sind. Die einzelnen Metadatenbeschreibungen enthalten hierzu die Namen der Suchfelder, die von den Suchservern unterstützt werden und ihre Abbildung auf die äquivalenten Suchkonzepte der Konzepttaxonomie (siehe Abschnitt 4.5.2). Die Konzepttaxonomie wird von dem sogenannten Taxonomie-Server verwaltet. Dieser kann entscheiden, ob ein gegebenes Suchkonzept ein anderes subsumiert.

Durchführen von lokalen Anfragen an den Suchservern Jeder Suchserver wird in die Metasuchmaschine durch einen spezifischen Wrapper integriert. Der Mediator initiiert eine Anfrage an einem selektierten Suchserver, indem er dessen Wrapper aufruft. Dieser kapselt die individuellen Eigenschaften der selektierten Suchserver. Hierzu übersetzen Wrapper eine gegebene Suchanfrage in die Abfragesprache des Suchservers. Dies erfordert die Erzeugung eines entsprechenden HTTP-Requests unter Berücksichtigung der Suchserver-spezifischen Gegebenheiten wie dem URL des Suchskripts oder dem verwendeten Request-Typ (GET oder POST). Weiterhin muß der Wrapper aus der im HTTP-Response zurückgelieferten Ergebnisseite eine geordnete Liste der gefundenen Dokumente erzeugen. Diese wird an den Mediator zurückgegeben, wobei sich jeder Treffer aus einem Link auf das gefundene Dokument sowie – falls auf der Ergebnisseite vorhanden – aus einem kurzen Snippet zusammensetzt. Die Verwendung von Mediatoren und Wrappern erleichtert die Integration

neuer Suchserver in QUEST. So muß für jeden neuen Suchserver ein neuer Wrapper implementiert werden, wobei dieser die Programmier-Schnittstelle des Mediators verwenden muß.

Erzeugung einer integrierten Resultatliste Die Rangliste, die der Assoziations-Server erzeugt, indem er die Relevanz eines Suchservers zu einer Suchanfrage einschätzt, kann ebenfalls dazu verwendet werden, um eine integrierte Resultatliste zu erzeugen. Je höher ein individueller Suchserver gewichtet wurde, desto höher sind die einzelnen Dokumente zu gewichten, die über diesen Suchserver recherchiert werden konnten.

7.2. Kategorisierungskomponente

Die Aufgabe der Kategorisierungskomponente besteht in der Zuordnung von Yahoo-Kategorien zu Suchservern, wobei diese ausschließlich über den URL ihrer Start-Seite repräsentiert sind. Die Kategorisierung erfolgt dabei durch die systematische Auswertung der Häufigkeiten von Kategorie-Termen auf den Backlink-Seiten der Suchserver. Die Implementierung der Kategorisierungskomponente und der zugrundeliegenden DBMS-Komponente erfolgte unter Verwendung der Skriptsprache Python [83].

Die Kategorisierungskomponente (siehe Abbildung 7.2) erhält als Eingabe den URL u der Startseite des zu kategorisierenden Suchservers, sowie eine Menge von Kategorien $C = \{c_1, \ldots, c_n\}$. Für jede Kategorie c_i berechnet die Kategorisierungskomponente ein Assoziationsgewicht $weight(u, c_i)$ für den durch u spezifizierten Suchserver. Als Ergebnis wird eine Rangliste $R_u(C, o_u)$ zurückgeliefert. In dieser liegen die $c_i \in C$ sortiert gemäß einer Ordnung o_u vor, wobei sich die Ordnung aus der absteigenden Höhe der einzelnen Gewichte $weight(u, c_i)$ (siehe Abschnitt 5.2.3.2) ergibt.

Jede Kategorie besteht wiederum aus einer Menge von Kategorietermen $c_i = \{t_1, \ldots, t_m\}$. Zur Berechnung des Gewichtes müssen die Häufigkeiten $f_{u,j}$ und df_j ermittelt werden. Diese werden unter Verwendung einer generellen Suchmaschine durch die in der Abbildung 7.2 dargestellten DBMS-Komponente ermittelt und an die Kategorisierungskomponente zurückgeliefert.

Die DBMS-Komponente verfügt über einen lokalen Speicher für die Häufigkeiten $f_{u,j}$ und df_j. Dieser wurde in Python unter Verwendung des *anydbms*-Moduls realisiert [99]. Zuerst erfolgt die Überprüfung, ob die angeforderten Daten im lokalen Speicher vorhanden sind, wenn ja, werden die Daten hieraus ausgelesen und an die Kategorisierungskomponente zurückgeliefert.

Sind die Daten jedoch nicht lokal vorhanden, so müssen diese unter Verwendung ei-

ner generellen Suchmaschine ermittelt werden. Hierzu werden u und t_j an einen Anfragegenerator übergeben. Dieser erzeugt hieraus eine Anfrage gemäß den syntaktischen Vorgaben der Abfragesprache der verwendeten generellen Suchmaschine. Unter Benutzung des Phyton-*urllib*-Moduls [99] baut der Anfragegenerator eine HTTP-Verbindung auf und übergibt einen entsprechend erzeugten HTTP-Request an den Suchserver der generellen Suchmaschine. Aus der im HTTP-Response zurückgelieferten Resultatseite wird die Information extrahiert, wie viele Dokumente gefunden wurden, die die gestellte Anfrage erfüllen. Diese Werte entsprechen den gesuchten Häufigkeiten und werden in den lokalen Speicher eingelesen und anschließend an die Kategorisierungskomponente zurückgeliefert.

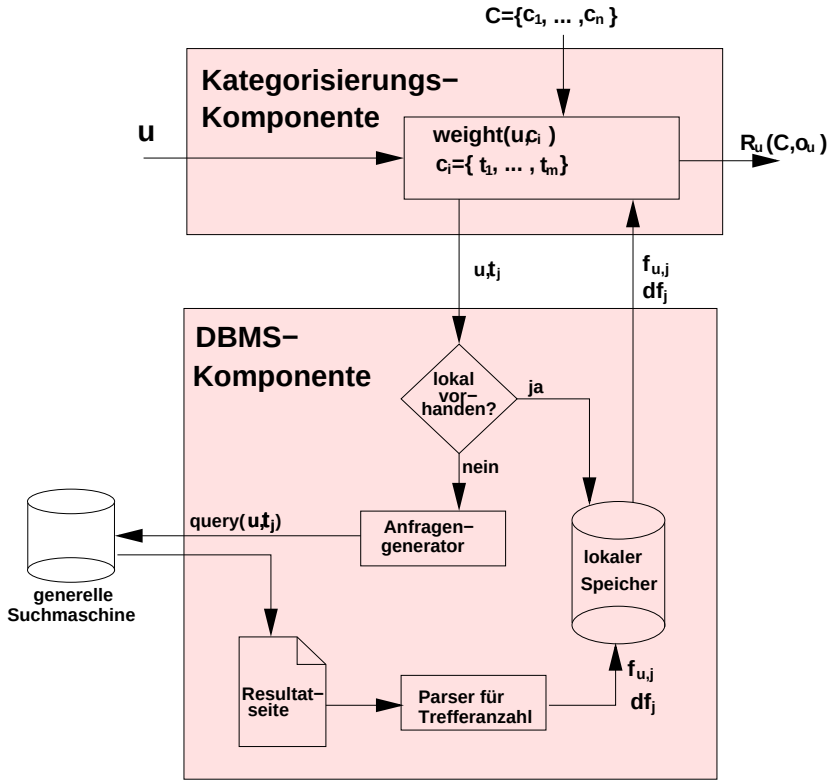

Abbildung 7.2.: Architektur der Kategorisierungskomponente

7.3. Assoziations-Server

Der Assoziations-Server erlaubt die Gewichtung einer Menge von Suchservern $S = \{s_1, \ldots, s_n\}$ bzgl. einer Benutzeranfrage q.

Für jeden Suchserver s_i berechnet der Assoziations-Server ein Gewicht, gemäß der in Abschnitt 6.3.2 beschriebenen Funktion $weight(q, T_{s_i})$. Dabei wird jeder Suchserver

s_i durch eine Menge von charakteristischen Termen repräsentiert $T_{s_i} = \{t_1, \ldots, t_m\}$. Diese werden aus der FC-Metadatenbeschreibung eines Suchservers s_i ausgelesen (siehe Abschnitt 6.3.1). Als Ergebnis wird eine Rangliste $R_q(S, o_q)$ zurückgeliefert. In dieser liegen die $s_i \in S$ sortiert gemäß einer Ordnung o_q vor, wobei sich die Ordnung aus der absteigenden Höhe der berechneten Assoziationsgewichte $weight(q, T_{s_i})$ ergibt. Mittels der Rangliste erfolgt die Selektion der anzufragenden Suchserver durch den Mediator. Die Architektur des Assoziations-Servers ist in Abbildung 7.3 dargestellt.

Zur Berechnung der Gewichte ist es erforderlich, die Häufigkeiten für die Ko-Zitierung von q und allen t_j sowie deren jeweilige Einzelvorkommen im WWW festzustellen, also df_q sowie $f_{q,j}$ und df_j für alle $t_j \in T_{s_i}$. Analog zu der in Abschnitt 5.2.3.2 beschriebenen Vorgehensweise für die Kategorisierungskomponente werden diese Häufigkeiten unter Verwendung einer generellen Suchmaschine ermittelt. Diese werden in der DBMS-Komponente des Assoziations-Servers abgelegt.

Bei der Berechnung eines Assoziationsgewichtes wird zuerst überprüft, ob die benötigten Häufigkeitswerte lokal vorhanden sind. Ist dies nicht der Fall, so werden entsprechende Anfragen für die generelle Suchmaschine generiert und versendet. Anschließend werden die angeforderten Häufigkeiten aus der Resultatseite ausgelesen, im lokalen Speicher abgelegt und an den Assoziations-Server zur Berechnung der Gewichte übergeben.

7.4. Der Taxonomie-Server

Der Taxonomie-Server verwaltet die definierten Suchkonzepte aller in der Metasuchmaschine integrierten Suchserver und erlaubt es festzustellen, ob ein Suchkonzept ein anderes subsumiert. Die Integration und die Verwendung einer derartigen Konzept-Taxonomie in einer Metasuchmaschinen-Architektur wird ausführlich in Abschnitt 4.5 besprochen. In QUEST erfolgt die Spezifikation der Konzept-Taxonomie unter Verwendung von Description Logic (siehe Abschnitt 4.5.5).

Aktuell stehen eine Vielzahl von Description Logic Implementierungen zur Verfügung[1]; im OIL-Projekt [63] wird beispielsweise das FaCT-System [62] eingesetzt. QUEST verwendet zur Beschreibung der Konzept-Taxonomie das LISP-basierte CLASSIC-System.

[1] Übersicht: http://www.ida.liu.se/labs/iislab/people/patla/DL/systems.html [14. Nov. 2001]

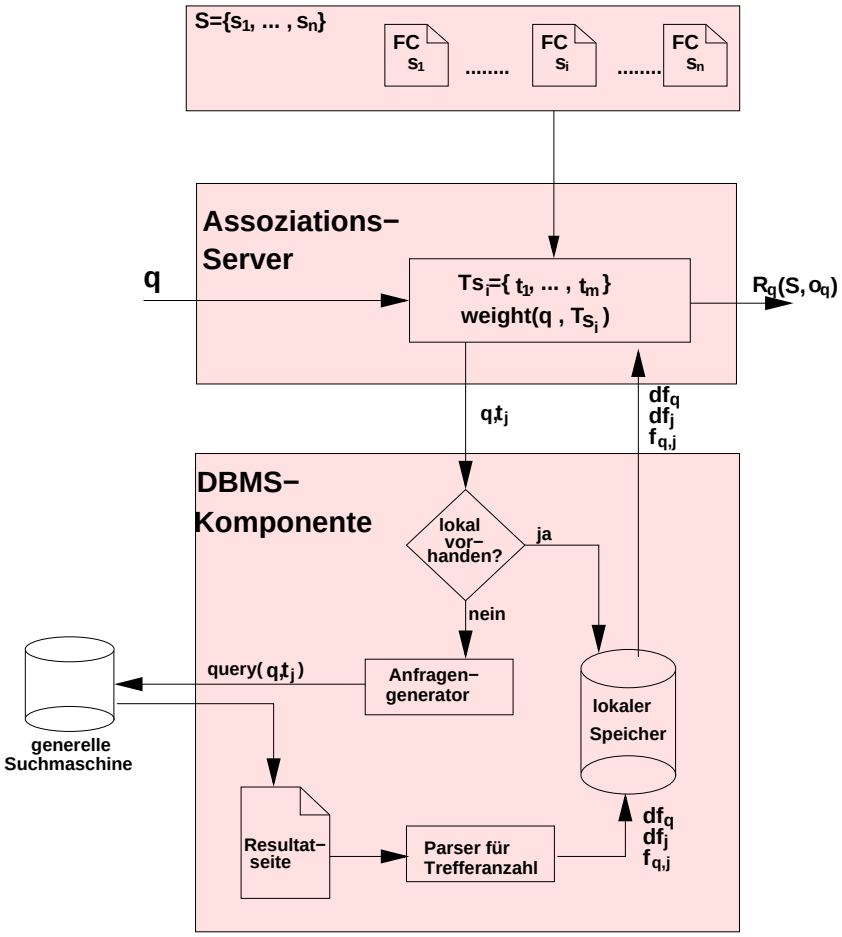

Abbildung 7.3.: Architektur des Assoziations-Servers

7.4.1. Das CLASSIC-System

CLASSIC [17] stellt eines der am weitesten verbreiteten Systeme zur Wissensrepräsentation (Knowledge Representation, KR) dar. Nach Ansicht der Autoren ist CLASSIC dabei eher als ein deduktives objektorientiertes Datenbanksystem zu sehen und weniger als ein generelles KR-System zum Aufbau komplexer und vollständiger Expertensysteme. Vielmehr stand bei dem Entwurf von CLASSIC der Gedanke im Vordergrund, einen einfachen und leicht bedienbaren Zugang zu KR-Techniken zu schaffen, um auch Anwendern, die keine KI-Experten sind, eine Möglichkeit an die Hand zu geben, anwendungsspezifische Wissensbereiche aufzubauen und zu pflegen. Die einzelnen Sprachelemente des CLASSIC-Systems werden ausführlich in [97] beschrieben. Im folgenden werden die wichtigsten Eigenschaften von CLASSIC vorgestellt:

Konzepte in CLASSIC sind Instanzen des vorderfinierten Konzeptes `classic-thing`. In CLASSIC existieren drei Funktionen, um neue Konzepte zu definieren:

1. `cl-define-concept <Symbol> <CLASSIC-Beschreibung>`
2. `cl-define-primitve-concept <Symbol> <CLASSIC-Beschreibung>`
3. `cl-define-disjoint-primitive-concept <Symbol> <CLASSIC-Beschreibung> <Gruppen-Symbol>`

Beispielsweise wird durch den Befehl `(cl-define-primitive-concept 'person 'classic-thing)` ein Konzept *person* spezifiert und als `classic-thing` deklariert. Im Gegensatz zu den durch `cl-define-primitive-concept` definierten Konzepten sind die durch `cl-define-concept` definierten Konzepte vollständig spezifiziert, d. h. die beiden auf den Befehl folgenden Konzepte sind äquivalent. Im Befehl `cl-define-disjoint-primitive-concept` kann zusätzlich die Angabe einer Gruppe (bzw. einer Liste von Gruppen) erfolgen, in der sich die spezifizierten Konzepte disjunkt zueinander verhalten. Beispielsweise kann man durch die Befehle `(cl-define-disjoint-primitive-concept 'man 'person 'gender)` bzw. `(cl-define-disjoint-primitive-concept 'woman 'person 'gender)` festlegen, daß sich die Konzepte *man* und *woman* disjunkt bzgl. einer Gruppe *gender* (=Geschlecht) verhalten. Entsprechende Befehle existieren in CLASSIC für die Definition von Rollen und Instanzen.

CLASSIC stellt darüberhinaus verschiedene Operatoren bereit, mit deren Hilfe komplexe Konzepte konstruiert werden können. Die wichtigsten Befehle sind:

- `(and` $C_1 ... C_n$ `)`: Konjunktion von Konzepten $C_1, \ldots, C_n$,
- `(all R C)`: Einschränkung eines Konzeptes C mittels einer Rolle R,
- `(at-least n R)`: Festlegung einer minimalen Kardinalität n für die Rolle R,
- `(at-most n R)`: Festlegung einer maximalen Kardinalität n für die Rolle R.

7.4.2. Spezifikation einer Konzepttaxonomie mittels CLASSIC

Mit dem Start des CLASSIC-Servers erfolgt die Initialisierung der Wissensbasis. Dabei wird dem Server eine Datei übergeben, in der die CLASSIC-Befehle hinterlegt sind. Die Spezifikation entspricht im Wesentlichen der in Abschnitt 4.5.3 beschriebenen Konzept-Taxonomie. Beispielsweise erfolgt mittels der Befehle: `(cl-define-primitive-concept 'abstract 'concept)`, `(cl-define-primitive-concept 'content 'abstract)` und

`(cl-define-primitive-concept 'title 'content)` die Definition der Konzepte *title*, *content* und *abstract* durch jeweilige Spezialisierung aus einem übergeordneten Konzept. Das allgemeinste Konzept (*concept*) entspricht dem in CLASSIC vorgegebenen Konzept *classic-thing*.

Neben Spezialisierungen in der Wissensbasis gemäß der Konzept-Taxonomie werden die Eigenschaften der Description Logic und CLASSIC allerdings auch dazu genutzt, um weiterführende Eigenschaften von Konzepten zu definieren. Mit den Befehlen:

```
(cl-define-primitive-role 'creatorOf :inver-
se 'creationOf)
(cl-define-primitive-role 'producerOf :inver-
se 'productOf)
(cl-define-concept 'document
                   '(and tangible
                         (at-least 1 creatorOf)
                         (at-least 1 producerOf)))
```

werden zwei Rollen `creatorOf` und `producerOf` sowie deren Umkehrrelationen definiert. Das Konzept *document* wird hier spezifiziert als ein *tangible*-Konzept (also ein stoffliches Konzept) das mindestens einen Erzeuger (z. B. ein Autor) und mindestens einen Produzenten (z. B. ein Verlag) besitzt. Man sieht, daß sich mittels CLASSIC also auch Eigenschaften definieren lassen, die über eine reine Subsumierung hinausgehen.

In der aktuellen Version von QUEST wird die CLASSIC-Wissensbasis bislang dazu genutzt, um zu einem gegebenen Konzept jene Konzepte zu ermitteln, die gleich spezifisch oder spezifischer sind.

7.4.3. Integration in QUEST

Das Abfragen der CLASSIC-Wissensbasis erfolgt in QUEST über die Instantiierung eines Client-Objekts. Die zugehörige Klasse wurde in Python implementiert. Mit der Erzeugung des Objektes wird eine Verbindung zum CLASSIC-Server hergestellt und das Mediatorprogramm kann über die in der Klasse definierten Methoden Abfragen an die Wissensbasis stellen:

- Mittels der Methode `isConcept(concept)` kann der Mediator erfragen, ob ein Konzept in der Taxonomie spezifiziert wurde.

- Mittels der Methode `subsumes(concept1,concept2)` kann erfragt werden, ob ein Konzept ein anderes subsumiert.

Die Konzepte werden dabei als Zeichenketten interpretiert. Der CLASSIC-Client setzt in den Methoden die Abfrage in ein entsprechenden LISP-Kommandos um und versendet diese an den Server, der die Anfrage mit einem entsprechenden Wahrheitswert beantwortet (TRUE oder FALSE).

Soll eine Anfrage an einen Suchserver versendet werden, so wird für ein Suchkonzept, das in der Anfrage spezifiziert wurde, überprüft, ob die in der Metadatenbeschreibung für diesen Suchserver hinterlegten Suchkonzepte spezifischer oder gleich dem Anfragekonzept sind (siehe Abschnitt 4.5.4). Dies ermittelt der Mediator unter Verwendung der `subsumes()`-Methode.

Die Architektur des Taxonomie-Servers ist in Abbildung 7.4 dargestellt.

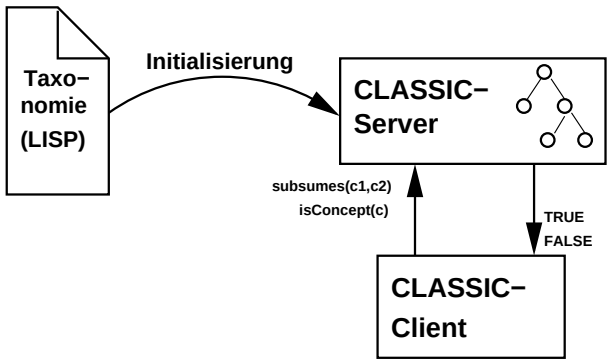

Abbildung 7.4.: Architektur des Taxonomie-Servers

7.5. Mediator und Wrapper

Dem Mediator von QUEST wird beim Aufruf eine Suchanfrage übergeben. Diese besteht aus zwei Teilen: einem Suchkonzept und einem Suchwert, z. B. *person:"steven spielberg"* oder *title="information retrieval"*. Über die Durchführung der Anfrage an den lokalen Suchservern hinaus besteht die Hauptaufgabe des Mediators in der Vorbereitung der einzelnen Suchanfragen. Dies erfordert die Koordinierung der in Abschnitt 7.1.3 beschriebenen Arbeitsschritte.

Der folgende Code ist an die Python-Implementierung des Mediators angelehnt und zeigt dessen Arbeitsweise unter Verwendung externer Komponenten, wie dem Taxonomie-Server und dem Assoziations-Server.

```
# Übergabe einer spezifizierten Suchanfrage an die
# Such-Methode des Mediators. Suchanfrage bestehen aus einem
# Suchkonzept (search_concept) und einem
# (Volltext-)Wert (value). Rückgabe ist eine integrierte
# Trefferliste (integrated_hit_list)
def perform_search(search_concept,value)

   # Liste, in der die gefundenen Treffer der einzelnen
   # Suchserver gesammelt werden. Jeder Treffer besteht aus
   # einer URL und einem kurzen Snippet.
   # Initial ist die Liste leer.
   global_hit_list = []

   # Anfrage an den Assoziations-Server welche Suchserver die
   # höchste Relevanz mit dem spezifizierten Wert value
   # aufweisen. Rückgabewert ist eine Liste der selektierten
   # Suchservern bzw. deren eindeutigen Server-IDs.
   server_list = assoc_server(value)

   # Für jeden Suchserver wird eine Anfrage vorbereitet
   for server_id in server_list:

        # Über die Metadatenbeschreibung des Suchservers
        # wird ermittelt, welche Suchkonzepte in der
        # Suchschnittstelle des Suchservers unterstützt
        # werden. Rückgabe ist eine Liste der unterstützten
        # Suchkonzepte.
        supported_concept_list = lookup_metadata(server_id)

        # Für alle unterstützten Suchkonzepte eines
        # Suchservers wird überprüft, ob diese von dem
        # spezifizierten Suchkonzept subsumiert werden.
        for concept in supported_concept_list:

                # Anfrage an den Taxonomie-Server, ob concept
                # spezieller oder oder gleich dem spezifizierten
                # search_concept ist.
                if (tax_server.subsumes(search_concept,concept) == TRUE):

                       # Anfrage an dem durch server_id
                       # spezifizierten Such-Server. Das Modul
                       # db_access kapselt die suchserver-
                       # spezifischen Wrapper und aktiviert
                       # diese zur Durchführung einer Anfrage.
                       local_hit_list=db_access(server_id,concept,value)
```

```
                    # Die durch die Anfrage erzeugte Ergebnisliste
                    # wird der globalen Ergebnisliste anghängt.
                    global_hit_list.append(local_hit_list)

# Erzeugung einer integrierten Trefferliste.
integrated_hit_list = sort(global_hit_list)
return integrated_hit_list
```

Zuerst kontaktiert der Mediator den Assoziations-Server. Dieser liefert eine Liste eindeutiger Server-IDs der selektierten Suchserver zurück (`server_list`). Für jeden Server dieser Liste werden Anfragen vorbereitet.

Die Methode `lookup_metadata` greift auf die FC-Metadatenbeschreibung des durch `server_id` spezifizierten Suchservers zu und liefert eine Liste der Suchkonzepte zurück, die durch dessen Suchschnittstelle unterstützt werden (`supported_concept_list`).

Unter Verwendung des Taxonomie-Servers wird für jedes der Konzepte in `supported_concept_list` überprüft, ob es vom spezifizierten Suchkonzept der Suchanfrage subsumiert wird. Wenn ja, wird eine entsprechende Suchanfrage an den Suchserver versendet, wobei für jede Anfrage ein eigener Thread erzeugt wird.

Anfragen werden mittels des Moduls `db_access` durchgeführt. Dieser erzeugt das Wrapperobjekt des durch `server_id` spezifizierten Suchservers und übersetzt die Suchanfrage in die Abfragesprache des Suchservers. Der serverspezifische Wrapper führt eine lokale Anfrage beim Suchserver durch und generiert aus der Resultatseite eine Trefferliste. Dabei ist jeder Treffer durch ein Tupel (URL, Snippet) repräsentiert, wobei ein einzelner Snippet maximal 100 Terme umfaßt und sich aus dem Ankertext eines Treffers sowie evtl. nachfolgendem Text bis zum nächsten Treffer zusammensetzt. Als Rückgabe einer Anfrage an `db_access` erhält der Mediator die vom Wrapper erzeugte lokale Trefferliste und fügt diese in eine globale Trefferliste `global_hit_list` ein.

Nachdem alle Anfragen erfolgreich durchgeführt wurden bzw. alle erzeugten Anfrage-Threads terminiert sind, wird durch den Aufruf `integrated_hit_list = sort(global_hit_list)` eine integrierte Trefferliste erzeugt, in der die einzelnen Treffer gemäß der durch den Assoziations-Server berechneten Relevanz eines Suchservers zu einer Suchanfrage angeordnet werden. Aus dieser Liste generiert der Mediator eine Resultat- HTML-Seite und versendet diese an den HTTP-Client.

7.6. Beispiel für eine Anfragen in QUEST

In Abbildung 7.5 ist exemplarisch eine Anfrage an QUEST und deren Resultatseite aufgezeigt.

Zur Anfrageformulierung können die in Abbildung 4.1 dargestellten Suchkonzepte verwendet werden. Im Beispiel soll eine Personensuche durchgeführt werden (Suchkonzept: *person*, Suchbegriff: *gagern*). Zur Durchführung der Anfrage wurden drei Suchserver selektiert:

- Der 1848-Flugschriften-Server (`db_1848`),
- die Internet-Movie Database (`db_imdb`) und
- der Web-OPAC der Stadt und Universitätsbibliothek Frankfurt(Main) (`db_opac`)

Der Suchserver `db_1848` unterstützt ein personenbezogenes Suchfeld und die Suchserver `db_imdb` und `db_opac` jeweils zwei. Dementsprechend werden 5 Anfrage-Threads an die Suchserver generiert und die Anfragen lokal an den Suchservern durchgeführt. Das in Abbildung 7.5 dargestellte Suchprotokoll zeigt die von jedem Anfrage-Thread ermittelte Trefferanzahl und die Antwortzeiten.

In der Resultatliste sind für jeden Suchserver exemplarisch jeweils drei Trefferdokumente angezeigt. Einzelne Resultate werden durch einen Hyperlink auf die Originaldokumente in den verschiedenen Suchservern und einen Snippet repräsentiert.

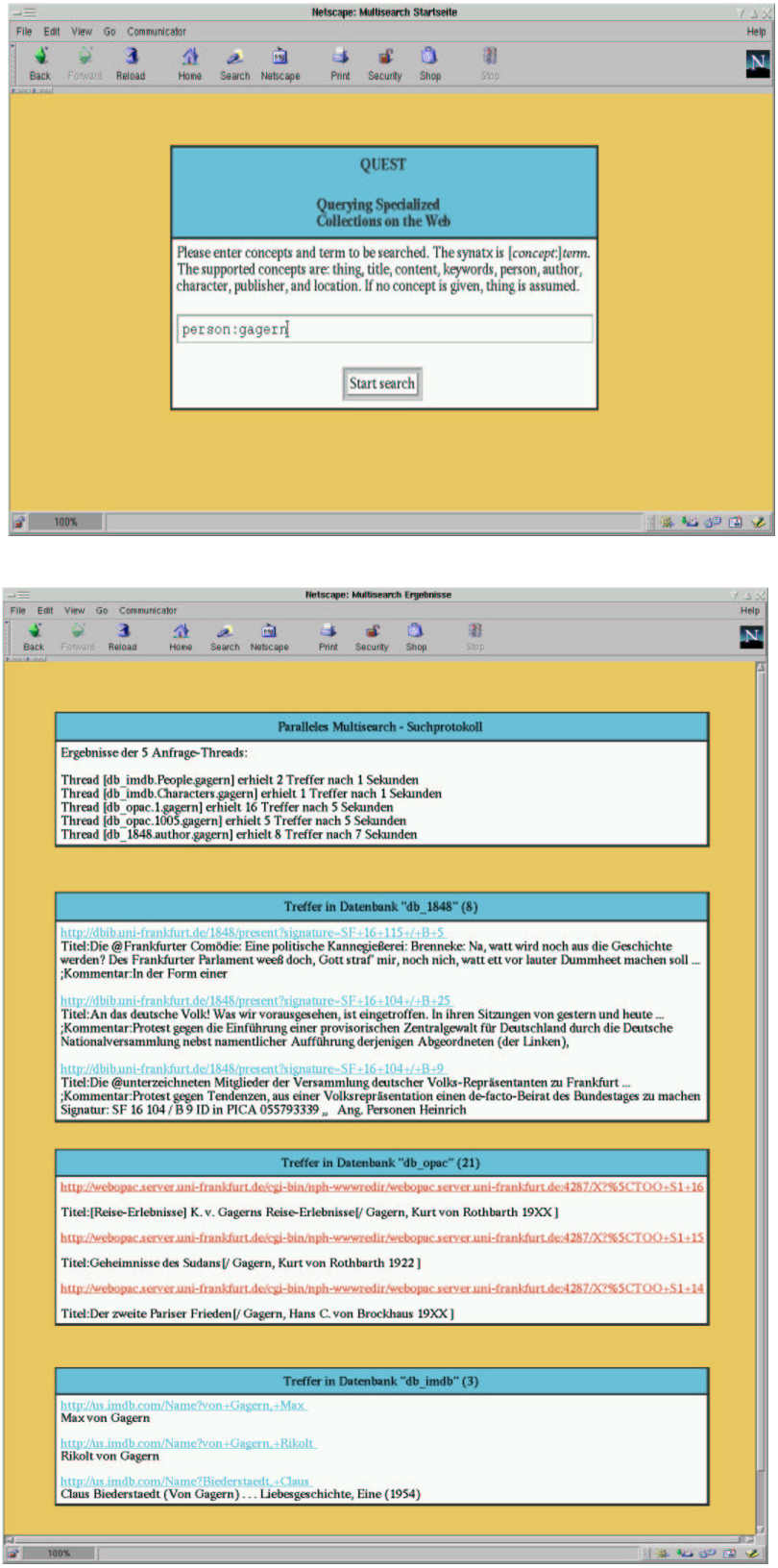

Abbildung 7.5.: Beispielanfrage an QUEST

8. Zusammenfassung und Ausblick

8.1. Zusammenfassung und Beiträge zur Forschung

Ziel der Arbeit war es, neue Techniken zur Erschließung und Selektion von Web-basierten Suchservern zu entwickeln und zu evaluieren, um hieraus eine integrierte Architektur für nicht-kooperative Suchserver im WWW abzuleiten. Dabei konnte gezeigt werden, daß die im Sichtbaren Web vorhandene Informationsmenge dazu geeignet ist, um eine effektive Erschließung des Unsichtbaren Webs zu unterstützen.

Existierende Strategien für verteiltes Information Retrieval setzen eine explizite Kooperation von Seiten der Suchserver voraus. Insbesondere Verfahren zur Selektion von Suchservern basieren auf der Auswertung von umfangreichen Termlisten bzw. Termhäufigkeiten, um eine Auswahl der potentiell relevantesten Suchserver zu einer gegebenen Suchanfrage vornehmen zu können (z. B. CORI [26] und GlOSS [54]). Allerdings werden derartige Informationen von realen Suchservern des WWW in der Regel nicht zu Verfügung gestellt. Die meisten Web-basierten Suchserver verhalten sich nicht kooperativ gegenüber hierauf aufsetzenden Metasuchsystemen, was die Übertragbarkeit der Selektionsverfahren auf das WWW erheblich erschwert.

Außerdem erfolgt die Evaluierung der Selektionsstrategien in der Regel in Experimentumgebungen, die sich aus mehr oder weniger homogenen, künstlich partitionierten Dokumentkollektionen zusammensetzen und somit das Unsichtbare Web und dessen inhärente Heterogenität nur unzureichend simulieren.

Dabei bleiben Daten unberücksichtigt, die sich aus der Einbettung von Suchservern in die Hyperlinkstruktur des WWW ergeben. So bietet z. B. die systematische Auswertung von Backlink-Seiten – also jener Seiten die einen Hyperlink auf die Start- oder Suchseite eines Suchservers enthalten – die Möglichkeit, die im WWW kollektiv geleistete Indexierungsarbeit zu nutzen, um die Erschließung von Suchservern effektiv zu unterstützen.

Eine einheitliche Systematik zur Beschreibung von Suchservern Zunächst ist es notwendig alle Informationen, die über einen Suchserver erreich-

bar sind, in ein allgemeingültiges Beschreibungsmodell zu integrieren. Dies stellt eine Grundvorraussetzung dar, um die einheitliche Intepretierbarkeit der Daten zu gewährleisten, und somit die Vergleichbarkeit von heterogenen Suchservern und den Aufbau komplexer Metasuchsysteme zu erlauben. Ein solche Beschreibung soll auch qualitative Merkmale enthalten, aus denen sich Aussagen über die Reputation einer Ressource ableiten lassen. Existierende Beschreibungen von Suchservern bzw. Dokumentkollektionen wie STARTS-CS [53] oder RSLP-CD [93] realisieren – wenn überhaupt – nur Teilaspekte hiervon.

Ein wichtiger Beitrag dieser Arbeit besteht somit in der Identifizierung und Klassifizierung von suchserverbeschreibenden Metadaten und hierauf aufbauend der Spezifikation eines als Frankfurt Core bezeichneten Metadatensatzes für web-basierte Suchserver, der die genannten Forderungen erfüllt. Der Frankfure Core berücksichtigt Metadaten, deren Erzeugung eine explizite Kooperation von Seiten der Suchserver voraussetzt, als auch Metadaten, die sich automatisiert – z. B. durch linkbasierte Analyseverfahren – aus dem sichtbaren Teil des WWW generieren lassen.

Integration von Wissensdarstellungen in Suchserver-Beschreibungen
Ein wichtige Forderung an Suchserver-Beschreibungen besteht in der zusätzlichen Integration von wissens- bzw. ontologiebasierten Darstellungen. Anhand einer in Description Logic spezifizierten Taxonomie von Suchkonzepten wurde in der Arbeit exemplarisch eine Vorgehensweise aufgezeigt, wie die Integration von Wissensdarstellungen in eine Frankfurt Core Beschreibung praktisch umgesetzt werden kann. Dabei wurde eine Methode entwickelt, um unter Auswertung einer Suchkonzept-Taxonomie Anfragen an heterogene Suchschnittstellen verschiedener Suchserver zu generieren, ohne die Aussagekraft von kollektionsspezifischen Suchfeldern einzuschränken. Durch die Taxonomie wird die einheitliche Verwendung von syntaktisch und semantisch divergierenden Suchfeldern verschiedener Suchserver sowie deren einheitliche Verwendung auf der integrierten Suchschnittstelle eines Metasuchsystems sichergestellt.

Damit kann diese Arbeit auch in Zusammenhang mit den Aktivitäten des Semantischen Webs betrachtet werden. Die Abstützung auf Description Logic zur Wissensrepräsentation sowie die Verwendung von RDF zur Spezifikation des Frankfurt Core verhält sich konform zu aktuellen Aktivitäten im Bereich Semantisches Web, wie beispielsweise der Ontology Inference Layer (OIL) [24]. Darüber hinaus konnte durch die Integration der Suchkonzept-Taxonomie in den Arbeitsablauf einer Metasuchmaschine, bereits eine konkrete Anwendung demonstriert werden.

Entwicklung neuartiger Verfahren zur Erschließung von Suchservern Für einzelne Felder des Frankfurt Core wurden im Rahmen dieser Arbeit Strategien entwickelt, die aufzeigen, wie sich durch die systematische Auswertung von Backlink-Seiten Suchserver-beschreibende Metadaten automatisiert generieren lassen. Dabei konnte gezeigt werden, daß der Prozeß der automatisierten Erschließung von Suchservern durch die strukturelle und inhaltliche Analyse von Hyperlinks sinnvoll unterstützt werden kann. Zwar hat sich ein HITS-basiertes Clustering-Verfahren als wenig praktikabel erwiesen, um eine effiziente Erschließung von Suchservern zu unterstützen, dafür aber ein hyperlinkbasiertes Kategorisierungsverfahren. Das Verfahren erlaubt eine Zuordnung von Kategorien zu Suchservern und kommt ohne zusätzliche Volltextinformationen aus. Dabei wird das WWW als globale Wissenbasis verwendet: die Zuordnung von Kategorienbezeichnern zu Web-Ressourcen basiert ausschließlich auf der Auswertung von globalen Term- und Linkhäufigkeiten wie sie unter Verwendung einer generellen Suchmaschine ermittelt werden können. Der Grad der Ähnlichkeit zwischen einer Kategorie und einer Ressource wird durch die Häufigkeit bestimmt, mit der ein Kategoriebezeichner und ein Backlink auf die Ressource im WWW kozitiert werden.

Durch eine Reihe von Experimenten konnte gezeigt werden, daß der Anteil korrekt kategorisierter Dokumente an Verfahren heranreicht, die auf Lerntechniken basieren. Das dargestellte Verfahren läßt sich leicht implementieren und ist nicht auf eine aufwendige Lernphase angewiesen, da die zu kategorisierenden Ressourcen nur durch ihren URL repräsentiert werden. Somit erscheint das Verfahren geeignet, um existierende Kategorisierungsverfahren für Web-Ressourcen zu ergänzen.

Ein Verfahren zur Selektion von Suchservern Ein gewichtiges Problem, durch welches sich die Selektion von Suchservern im WWW erheblich erschwert, besteht in der Diskrepanz zwischen der freien Anfrageformulierung auf Benutzerseite und nur spärlich ausgezeichneten Suchserver-Beschreibungen auf Seiten des Metasuchsystems. Da auf der Basis der geringen Datenmenge eine Zuordnung der potentiell relevantesten Suchserver zu einer Suchanfrage kaum vorgenommen werden kann, wird oft auf zusätzliches Kontextwissen zurückgegriffen, um z. B. ein Anfragerweiterung durch verwandte Begriffe vornehmen zu können (siehe z. B. QPilot [110]). Eine solche Vorgehensweise erhöht allerdings nur die Wahrscheinlichkeit für Treffer von Anfragetermen in den Suchserver-Beschreibungen und liefert noch keine ausreichende Sicherheit.

Deshalb wurde in der Arbeit ein Selektionsverfahren entwickelt, das sich auf die Auswertung von Ko-Zitierungs- und Dokumenthäufigkeiten von Termen in großen Dokumentsammlungen abstützt. Das Verfahren berechnet ein Gewicht zwischen einem

Anfrageterm und einem Suchserver auf der Basis von einigen wenigen Deskriptortermen, wie sie z. B. aus der FC-Beschreibung eines Suchservers extrahiert werden können. Dies hat den Vorteil, daß die Suchbegriffe nicht explizit in den einzelnen Suchserver-Beschreibungen vorkommen müssen, um eine geeignete Selektion vornehmen zu können.

Um die Anwendbarkeit des Verfahrens in einer realistischen Web-Umgebung zu demonstrieren, wurde eine geeignete Experimentumgebung von spezialisierten Suchservern aus dem WWW zusammengestellt. Durch anschließende Experimente konnte die Tauglichkeit des entwickelten Verfahrens aufgezeigt werden, indem es mit einem Verfahren verglichen wurde, das auf Probe-Anfragen basiert. Das heißt, daß eine erfolgreiche Selektion durchgeführt werden kann, ohne daß man explizit auf das Vorhandensein von lokalen Informationen angewiesen ist, die erst aufwendig durch das Versenden von Probe-Anfragen über die Web-Schnittstelle des Suchservers extrahiert werden müssten.

Herleitung einer integrierten Architektur Um das Zusammenspiel der erarbeiteten Strategien und Techniken zur Erschließung, Beschreibung und Selektion in einer integrierten Architektur umzusetzen, wurde die Metasuchmaschine QUEST entwickelt und prototypisch implementiert.

QUEST erweitert die Architektur einer traditionellen Metasuchmaschinenarchitektur, um Komponenten, die eine praktische Umsetzung der Konzepte und Techniken darstellen, die im Rahmen dieser Arbeit entwickelt wurden. QUEST bildet einen tragfähigen Ansatz zur Kombination von wissensbasierten Darstellungen auf der einen und eher heuristisch orientierten Methoden zur automatischen Metadatengenerierung auf der anderen Seite. Dabei stellt der Frankfurt Core das zentrale Bindeglied dar, um die einheitliche Behandlung der verfügbaren Daten zu gewährleisten.

8.2. Zukünftige Arbeiten

In Anbetracht der Größe und des Wachstums des Unsichtbaren Webs stellt ein Metadatensatz für Suchserver eine wichtige Voraussetzung dar, um die zu erwartende Informationsvielfalt zu beherrschen. Die Felder des Frankfurt Core sind dabei keinesfalls unveränderlich festgeschrieben, sondern sollen vielmehr als initialer Vorschlag zur Auszeichnung von Suchservern betrachtet werden. Der Frankfurt Core kann somit als Diskussionsgrundlage für zukünftige Betrachtungen und Erweiterungen dienen. Insbesondere ist abzusehen, daß Informationen, die eine qualitative Einschätzung von Suchservern zulassen, im Zuge der anwachsenden Datenmenge immer stärker an

Bedeutung gewinnen werden.

Anhand des Problems heterogene Suchfelder verschiedener Suchserver einheitlich zu repräsentieren, wurde in der Arbeit demonstriert, inwieweit sich die Integration von wissensbasierten Darstellungen mit Hilfe des Frankfurt Cores bewerkstelligen und praktisch umsetzen läßt. Eine solche Vorgehensweise kann prinzipiell auch auf weitere Suchserver-Eigenschaften übertragen werden – insbesondere auf solche, die eine inhaltliche Einordnung von Suchservern erlauben. Denkbar wäre beispielsweise der Aufbau und die Integration einer gemeinsamen Ontologie für geographische Zusammenhänge für das FC-Feld places. Der dargestellte Ansatz ist somit geeignet, um umfangreiche Web-Portale für bestimmte wissenschaftliche Disziplinen wie Geschichte oder Biologie aufzubauen, indem die jeweils relevanten Suchserver eines bestimmten Fachgebietes zusammengefaßt werden. Die Qualität der integrierten Recherche kann durch die Berücksichtigung von fachspezifischen Wissensrepräsentationen unterstützt werden.

Wünschenwert wäre auch eine Erweiterung der QUEST-Suchschnittstelle durch eine Verbesserung der Visualisierung von Informationen. Siehe z. B. [7], [31], [40] für eine Übersicht von Visualisierungs-Techniken für die Bereiche Information Retrieval bzw. Digitale Bibliotheken. Denkbar ist beispielsweise die Verwendung einer visualisierten Ontologie – ähnlich dem *Hyperbolic Ontology View* des Ontobroker-Projektes [109] – zur Unterstüztung des Benutzers bei der Auswahl der anzufragenden Suchkonzepte während der Anfrageformulierung. Auf der anderen Seite existieren verschiedene Möglichkeiten zur graphischen Aufbereitung von großen Ergebnismengen bzw. Clustern, die aus den gefundenen Dokumenten gebildet werden. Erste geeignete Technologien hierfür sind beispielsweise *Self-Organizing Maps*[1] [70] oder *Lighthouse*[2] [81].

Eine weitere Fragestellung, die sich aus dieser Arbeit ergibt, besteht in der Entwicklung von Selektionsverfahren, die verstärkt qualitative Informationen berücksichtigen. Hierzu ist es zunächst notwendig geeignete Testumgebungen und Anfragen zusammenzustellen, die realistische WWW-Suchserver Szenarien widerspiegeln. Einen möglichen Aufsetzpunkt hierfür bilden die *Web Research Collections*, die im Rahmen der TREC Web Track Aktivität entwickelt werden[3]. Derartige Experimente würden es erlauben, existierende und zukünftige Qualitätsmerkmale von Suchservern gegeneinander auszumessen, um hieraus verbesserte Selektionsstrategien abzuleiten. Die in dieser Arbeit vorgestellten Entwicklungen bieten hierfür eine gute Ausgangsbasis.

[1] WEBSOM: http://websom.hut.fi/websom/ [14. Nov. 2001]
[2] Lighthouse: http://toowoomba.cs.umass.edu/~leouski/lighthouse/ [14. Nov. 2001]
[3] http://www.ted.cmis.csiro.au/TRECWeb/ [14. Nov. 2001]

Literaturverzeichnis

[1] L. Adamic. The Small World Web. In *Proceedings of the 3rd European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, volume 1696, pages 443–452, Paris (Frankreich), 1999.

[2] R. Albert, H. Jeong, and A. Barabasi. The Diameter of the World Wide Web. *Nature*, 401, September 1999.

[3] J. Ambite, N. Ashish, G. Barish, C. Knoblock, S. Minton, P. Modi, I. Muslea, A. Philpot, and S. Tejada. ARIADNE: A System for constructing Mediators for Internet Sources. In *Proceedings of ACM SIGMOD Conference on Management of Data*, Seattle (USA), 1998.

[4] B. Amento, L. Terveen, and W. Hill. Does 'Authority' mean Quality? Predicting Expert Quality Ratings on Web Documents. In *Proceedings of the 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 296–303, Athen (Griechenland), 2000.

[5] E. Amitay and C. Paris. Automatically Summarising Web Sites - Is There a Way Around it? In *Proceedings of the 9th ACM International Conference on Information and Knowledge Management (CIKM)*, McLean (USA), 2000.

[6] G. Attardi, A. Gullí, and F. Sebastiani. Automatic Web Page Categorization by Link and Context Analysis. In *Proceedings of THAI-99, European Symposium on Telematics, Hypermedia and Artificial Intelligence*, pages 105–119, Varese (Italien), 1999.

[7] R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval*. ACM Press Books, Addison-Wesley, 1999.

[8] T. Baker. A Grammar of Dublin Core. *D-Lib Magazine*, Oktober 2000.

[9] M. Baldonado, C.-C. K. Chang, L. Gravano, and A. Paepcke. The Stanford Digital Library Metadata Architecture. *International Journal of Digital Libraries*, 1(2), 1997.

[10] M.K. Bergman. The Deep Web: Surfacing Hidden Value. <URL:http://128.121.227.57/download/deepwebwhitepaper.pdf> [03.09.01], 2000.

[11] T. Berners-Lee. Semantic Web Road Map. <URL:http://www.w3.org/DesignIssues/Semantic.html> [03.09.01], 1998.

[12] T. Berners-Lee and M. Fischetti. *Weaving the Web: The Original Design and Ultimate Destiny of the World Wide Web by its Inventor*. Harper, San Francisco, 1999.

[13] T. Berners-Lee, J.Hendler, and O. Lassila. The Semantic Web. *Scientific American*, Mai 2001.

[14] K. Bharat and A. Z. Broder. A Technique for Measuring the Relative Size and Overlap of Public Web Search Engines. In *Proceedings of the 7th International World-Wide Web Conference*, Brisbane (Australien), 1998.

[15] K. Bharat and M. Henzinger. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne (Australien), 1998.

[16] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Finding Authorities and Hubs from Link Structures on the World Wide Web. In *Proceedings of the 10th International World-Wide Web Conference*, Hong Kong (China), 2001.

[17] R. Brachmann, A. Borgida, D. McGuinness, and P. Patel-Schneider. 'Reducing' CLASSIC to practice: Knowledge Representation Theory meets Reality. In *Proceedings of the 3rd International Conference on Principles of Knowledge Representation and Reasoning*, pages 247–258, Cambridge (USA), 1992.

[18] T. Bray, D. Hollander, and A. Layman. Namespaces in XML. <URL:http://www.w3.org/TR/1999/REC-xml-names-19990114/> [03.09.01], 1999.

[19] T. Bray, J. Paoli, C. M. Sperberg-McQueen, and E.Maler. Extensible Markup Language (XML) 1.0. <URL:http://www.w3.org/TR/2000/REC-xml-20001006> [03.09.01], 2000.

[20] E.A. Brewer. When everything is searchable. *Communications of the ACM*, 44(3), März 2001.

[21] D. Brickley and R.V. Guha. Resource Description Framework (RDF) Schema Specification 1.0. <URL:http://www.w3.org/TR/2000/CR-rdf-schema-20000327/> [03.09.01], 2000.

[22] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *Proceedings of the 7th International World-Wide Web Conference*, Brisbane (Australien), 1998.

[23] A. Broder, S. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph Structure in the Web. In *Proceedings 9th International World-Wide Web Conference*, Amsterdam (Holland), 2000.

[24] J. Broekstra, M. Klein, S. Decker, D. Fensel, F. Harmelen, and I. Horrocks. Enabling Knowledge Representation on the Web by extending RDF Schema. In *Proceedings of the 10th International World-Wide Web Conference*, Hong Kong (China), 2001.

[25] C. M. Brown, B. B. Danzig, D. Hardy, U. Manber, and M. F. Schwartz. The Harvest Information Discovery and Access System. In *In Proceedings of the 2nd International World-Wide Web Conference*, Chicago (USA), 1994.

[26] J. P. Callan, Z. Lu, and B. Croft. Searching Distributed Collections with Inference Networks. In *Proceedings of the 18th International ACM SIGIR Conference on Research and Development in Information Retrieval*, Seattle (USA), 1995.

[27] S. Chakrabarti. Data Mining for Hypertext: A Tutorial Survey. *SIGKDD Explorations: Newsletter of the Special Interest Group (SIG) on Knowledge Discovery and Data Mining, ACM*, 1, 2000.

[28] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic Resource Compilation by Analyzing Hyperlink Structure and Associated text. In *Proceedings of the 7th International World-Wide Web Conference*, Brisbane (Australien), 1998.

[29] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced Hypertext Categorization Using Hyperlinks. *SIGMOD Record (ACM Special Interest Group on Management of Data)*, 27(2), 1998.

[30] C. Chang and C. Hsu. Customizable multi-engine Search Tool with Clustering. In *Proceedings of 6th International World-Wide Web Conference*, Santa Clara (USA), 1997.

[31] C. Chen. *Information Visualization and Virtual Environments*. Springer, 1999.

[32] R. Cooley, J. Srivastava, and B. Mobasher. Web Mining: Information and Pattern Discovery on the World Wide Web. In *Proceedings of the 9th IEEE International Conference on Tools with Artificial Intelligence (ICTAI)*, Newport Beach (USA), November 1997.

[33] A. Craswell. Methods for Distributed Information Retrieval. Dissertation, Australian National University, 2000.

[34] N. Craswell, D. Hawking, and S. Robertson. Effective Site Finding using Link Anchor Information. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New Orleans (USA), 2001.

[35] N. Craswell, P.Bailey, and D.Hawking. Server Selection on the World Wide Web. In *Proceedings of the 5th ACM Conference on Digital Libraries*, pages 37–46, San Antonio (USA), 2000.

[36] B. D. Davison. Topical Locality in the Web. In *Proceedings of the 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 272–279, Athen (Griechenland), 2000.

[37] P. Diepold, J.Gasteiger, E.R. Hilf, E. Mittler, E. Niggemann, P.Schirmbacher, and G. Törner. DFG-Projekt Dissertation Online – Abschlußbericht. `<URL:http://www.educat.hu-berlin.de/diss_online/texte/abschluss.pdf>` [03.09.01], März 2001.

[38] R. Dolin, D. Agrawal, L. Dillon, and A. El Abbadi. Pharos: A Scalable Distributed Architecture for Locating Heterogeneous Information Sources. In *Proceedings of the 6th ACM International Conference on Information and Knowledge Management (CIKM)*, Las Vegas (USA), 1997.

[39] T. E. Dunning. Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics*, 19(1), 1993.

[40] A. Endres and D.W. Fellner. *Digitale Bibliotheken*. dpunkt.verlag, 2000.

[41] M. Ester and J. Sander. *Knowledge Discovery in Databases*. Springer Verlag, 2000.

[42] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy. *Advances in Knowledge Discovery and Data Mining*. MIT Press, 1996.

[43] D. Fensel. *Ontologies: Silver Bullet for Knowledge Management and Electronic Commerce*. Springer-Verlag, 2001.

[44] J. C. French, A. L. Powell, and J. Callan. Effective and Efficient Automatic Database Selection. Technical Report CS-99-08, Department of Computer Science, University of Virginia, 1999.

[45] J. Fürnkranz. Exploiting Structural Information for Text Classification on the WWW. In *Proceedings of IDA'99*, Amsterdam (Holland), 1999.

[46] N. Fuhr. Probabilistic Models in Information Retrieval. *The Computer Journal*, 35(3), 1992.

[47] N. Fuhr. Resource Discovery in Distributed Digital Libraries. In *Digital Libraries '99: Advanced Methods and Technologies, Digital Collections*, pages 35–45, St. Petersburg (Russia), 1999.

[48] G.W. Furnas, S. Deerwester, S. T. Dumais, T.K. Landauer, R.A. Harshman, L.A. Streeter, and K.E.Lochbaum. Information Retrieval using a Singular Value Decomposition Model of Latent Semantic Structure. In *Proceedings of the 11th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 465–480, 1988.

[49] B. Furrie. Understanding MARC Bibliographic: Machine-Readable Cataloging. `<URL:http://www.loc.gov/marc/umb/>` [03.09.01], 1990.

[50] C.L. Giles, K.D. Bollacker, and S.Lawrence. CiteSeer: An Automatic Citation Indexing System. In *Proceedings of the 3rd ACM Conference on Digital Libraries*, New York (USA), 1998.

[51] A. J. Gilliland-Swetland. *Defining Metadata*. Getty Information Institute, Los Angeles, 2000.

[52] A. Goldbach. Realisierung und Bewertung eines Verfahrens zu hyperlinkbasiertem Information Retrieval. Diplomarbeit, J. W. Goethe–Universität, Frankfurt, 2000.

[53] L. Gravano, K. Chang, H. Garcia-Molina, and A. Paepcke. STARTS: Stanford Proposal for Internet Meta-Searching. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, pages 207–218, Tucson (USA), 1997.

[54] L. Gravano, H. Garcia-Molina, and A. Tomasic. GlOSS: Text-Source Discovery over the Internet. *ACM Transactions on Database Systems*, 24(2), Juni 1999.

[55] N. Guarino. Formal Ontology and Information Systems. In *Proceedings of the 1st International Conference on Formal Ontologies in Information Systems (FOIS)*, pages 3 – 15, Trento (Italien), 1998.

[56] M. Heß and O. Drobnik. Clustering Specialized Web-Databases by Exploiting Hyperlinks. In *Proceedings of The 2nd Asian Digital Library Conference,1999*, pages 19–29, Taipei (Taiwan), November 1999.

[57] M. Heß, G. Krause, A. Nube, and M. Zimmermann. Ein generisches Managementsystem für verteilte Anwendungen - Architektur und Implementierung. In *Entwicklung und Management verteilter Anwendungssysteme*, Dortmund (Deutschland), Oktober 1995.

[58] M. Heß, C. Mönch, and O. Drobnik. QUEST - Querying Specialized Collections on the Web. In *Proceedings of the 4th European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*, pages 117–126, Lissabon (Portugal), September 2000.

[59] M. Heß, C. Mönch, and O. Drobnik. Ergebnisbericht zum Projekt: Entwicklung eines Systems zur Strukturierung, Speicherung und Bereitstellung von Dokumenten als Teil einer Infrastruktur für digitale Bibliotheken. `<URL: http://dbib.uni-frankfurt.de/ergebnisbericht.ps.gz>` [30.10.2001], Oktober 2001.

[60] J. Heflin, J. Hendler, and S. Luke. SHOE: A Knowledge Representation Language for Internet Applications. Technical Report CS-TR-4078, University of Maryland, 1999.

[61] T. Hofmann. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley (USA), 1999.

[62] I. Horrocks. The FaCT System. *Lecture Notes in Artificial Intelligence*, 1397, 1998.

[63] I. Horrocks, D. Fensel, C. Goble, F. Harmelen, J. Broekstra, M. Klein, and S. Staab. The Ontology Inference Layer OIL. Technical Report, Free University of Amsterdam, 2000.

[64] A. E. Howe and D. Dreilinger. SAVVYSEARCH: A Metasearch Engine that Learns which Search Engines to Query. *AI Magazine*, 18(2), 1997.

[65] L. Huang, M. Hemmje, and E. J. Neuhold. ADMIRE: An Adaptive Data Model for Meta Search Engines. In *Proceedings of the 9th International World-Wide Web Conference*, Amsterdam (Holland), 2000.

[66] C. H. Hwang. Incompletely and Imprecisely Speaking: Using Dynamic Ontologies for Representing and Retrieving Information. In *Knowledge Representation Meets Databases*, pages 14–20, Linköping (Schweden), 1999.

[67] B. Jansen, A. Spink, J. Bateman, and T. Saracevic. Real Life Information Retrieval: A Study of User Queries on the Web. *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, 32(1), 1998.

[68] C. P. Klas and N. Fuhr. A New Effective Approach for Categorizing Web Documents. In *Proceedings of the 22th BCS-IRSG Colloquium on IR Research*, Cambridge (England), 2000.

[69] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. In *Proceedings of the ACM-SIAM Smyposium on Discrete Algorithms*, New York (USA), 1998.

[70] T. Kohonen, S. Kaski, K. Lagus, J. Salojärvi, V. Paatero, and A. Saarela. Self Organization of a Massive Document Collection. *IEEE Transactions on Neural Networks, Special Issue on Neural Networks for Data Mining and Knowledge Discovery*, 11(3), Mai 2000.

[71] J. Kunze. Encoding Dublin Core Metadata in HTML. <URL:http://www.ietf.org/rfc/rfc2731.txt> [03.09.01], 1999. RFC 2731.

[72] C. Kwok, O. Etzioni, and D. S. Weld. Scaling Question Answering to the Web. In *Proceedings of the 10th International World-Wide Web Conference*, Hong Kong (China), 2001.

[73] H. Garcia-Molina L. Gravano and A. Tomasic. The Effectiveness of GlOSS for the Text Database Discovery Problem. In *Proceedings of the ACM SIGMOD Conference on Management of Data*, Minneapolis (USA), 1994.

[74] Y. Labrou and T. Finin. Yahoo! as an Ontology - Using Yahoo! Categories to Describe Documents. In *Proceedings of the 8th ACM International Conference on Information and Knowledge Managament (CIKM)*, pages 180–187, Kansas City (USA), 1999.

[75] R.R. Larson. Bibliometrics of the World Wide Web: An Exploratory Analysis of the Intellectual Structure of Cyberspace. In *Proceedings of the 1996 Annual ASIS Meeting*, Baltimore (USA), 1996.

[76] O. Lassila and R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. <URL:http://www.w3.org/TR/REC-rdf-syntax/> [03.09.01], 1999.

[77] S. Lawrence and C.L.Giles. Accessibility of Information on the Web. *Nature*, 400, 1999.

[78] S. Lawrence and C.L. Giles. Searching the World Wide Web. *Science*, 280(5360), 1998.

[79] R. Lempel and S. Moran. The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect. In *Proceedings of the 9th International World-Wide Web Conference*, pages 387–401, Amsterdam (Holland), Mai 2000.

[80] D.B. Lenat and R.V. Guha. *Building Large KnowledgeBased Systems: Representation and Inference in the CYC Project*. Addison-Wesley, 1990.

[81] A. Leuski and J. Allan. Lighthouse: Showing the Way to Relevant Information. In *In the Proceedings of IEEE Symposium on Information Visualization 2000 (InfoVis 2000)*, Salt Lake City (USA), 2000.

[82] K. Liu, W. Meng, and C. Yu. Discovery of Similarity Computations of Search Engines. In *Proceedings of the 9th ACM International Conference on Information and Knowledge Management (CIKM)*, McLean (USA), 2000.

[83] M. Lutz. *Programming Python*. O'Reilly & Associates, Sebastopol, CA, 1996.

[84] M. Marchiori. The Limits of Web Metadata, and Beyond. In *Proceedings of the 7th International World-Wide Web Conference*, Brisbane (Australien), 1998.

[85] D. Mladenic. Turning Yahoo into an automatic Webpage Classifier. In *Proceedings of the 13th European Conference on Artificial Intelligence ECAI98*, Brighton (England), 1998.

[86] D.S. Modha and W.S. Spangler. Clustering Hypertext with Applications to Web Searching. In *Proceedings of the 11th ACM Conference on Hypertext and Hypermedia*, San Antonio (USA), 2000.

[87] J. Myllymaki. Effective Web Data Extraction with Standard XML Technologies. In *Proceedings of the 10th International World-Wide Web Conference*, Hong Kong (China), 2001.

[88] A. Y. Ng, A. X. Zheng, and M. Jordan. Stable Algorithms for Link Analysis. In *Proceedings of the 24th International ACM SIGIR Conference on Research and Development in Information Retrieval*, New Orleans (USA), 2001.

[89] N. Noy and C. Hafner. The State of the Art in Ontology Design. *AI Magazine*, 18(3), 1997.

[90] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical Report, 1998.

[91] J.M. Pierre. Practical Issues for Automated Categorization of Web Sites. In *Proceedings of ECDL 2000 Workshop on the Semantic Web*, Lissabon (Portugal), 2000.

[92] C. Plaunt and B.Norgard. An Association-based Method for Automatic Indexing with a Controlled Vocabulary. *Journal of the American Society for Information Science*, 90(10), 1998.

[93] A. Powell. RSLP Collection Description. *D-Lib Magazine*, September 2000.

[94] D. Rafiei and A. O. Mendelzon. What is this Page known for? Computing Web Page Reputations. In *Proceedings of the 9th International World Wide Web Conference*, Amsterdam (Holland), 2000.

[95] S. Raghavan and H. Garcia-Molina. Crawling the Hidden Web. Technical Report, Stanford University, Dezember 2000.

[96] U. Reh. Entwicklung eines Systems zur Präsentation historischer Dokumente im WWW unter Berücksichtigung bibliothekarischer Anforderungen. Diplomarbeit, J. W. Goethe–Universität, Frankfurt, 1999.

[97] L. Resnick, A. Borgida, R. Brachman, D. McGuinness, and P. Patel-Schneider. Classic Description and Reference Manual for Common LISP Implementation. Technical Report, 1990.

[98] P. Resnik. Parallel Strands: A Preliminary Investigation into Mining the Web for Bilingual Text. In *Third Conference of the Association for Machine Translation in the Americas (AMTA-98)*, pages 72–82, Langhorne (USA), 1998.

[99] G.v. Rossum. Python Library Reference. `<URL:http://www.python.org/doc/current/lib/lib.html>` [03.09.2001], 2001.

[100] M. Sahami, S. Yusufali, and M.Q.W. Baldonado. SONIA: A Service for Organized Networked Information Autonomously. In *Proceedings of the 3rd ACM Conference on Digital Libraries*, New York (USA), 1998.

[101] A. Sahuguet and F. Azavant. Building Light-Weight Wrappers for Legacy Web Data-Sources Using W4F. In *Proceedings of the International Conference on Very Large Data Bases (VLDB)*, Edinburgh (Schottland), 1999.

[102] G. Salton. Associative Document Retrieval Techniques using Bibliographic Information. *Journal of the ACM*, 10(4), Oct 1963.

[103] G. Salton. *The SMART Retrieval System – Experiments in Automatic Document Processing*. Prentice Hall Inc., 1971.

[104] G. Salton and C. Buckley. Term Weighting Approaches in Automatic Text Retrieval. *Information Processing and Management*, 24(5), 1988.

[105] G. Salton and M.E.Lesk. Computer Evaluation of Indexing and Text Processing. *Journal of the ACM*, 15(1), 1968.

[106] F. Sebastiani. Machine Learning in Automated Text Categorisation. Technical Report IEI-B4-31-1999, Istituto di Elaborazione dell'Informazione, Consiglio Nazionale delle Ricerche, Pisa (Italien), 1999.

[107] E. Selberg and O. Etzioni. Multi-Service Search and Comparison Using the MetaCrawler. In *Proceedings of the 4th International World-Wide Web Conference*, Darmstadt (Deutschland), 1995.

[108] E. Selberg and O. Etzioni. The MetaCrawler Architecture for Resource Aggregation on the Web. *IEEE Expert*, pages 11–14, 1997.

[109] S. Staab, J. Angele, S. Decker, M. Erdmann, A. Hotho, A. Maedche, H.-P. Schnurr, R. Studer, and Y. Sure. Semantic Community Web Portals. In *Proceedings of the 9th International World-Wide Web Conference*, Amsterdam (Holland), 2000.

[110] A. Sugiura and O. Etzioni. Query Routing for Web Search Engines: Architecture and Experiments. In *Proceedings of the 9th International WWW Conference*, Amsterdam (Holland), 2000.

[111] D. Sullivan. Search Engine Sizes. *searchenginewatch.com*, November 2000.

[112] H. R. Turtle and W. B. Croft. Efficient Probabilistic Inference for Text Retrieval. In *RIAO 3 Conference Proceedings*, Barcelona (Spain), 1991.

[113] C. J. van Rijsbergen. *Information Retrieval*. 2nd ed., Butterworths, 1979.

[114] E. M. Voorhees, N. K. Gupta, and B. Johnson-Laird. The Collection Fusion Problem. In *Proceedings of the Third Text Retrieval Conference (TREC-3)*, Gaithersburg (USA), 1995.

[115] S. Weibel, J. Kunze, C. Lagoze, and M. Wolf. Dublin Core Metadata for Resource Discovery. `<URL:http://www.ietf.org/rfc/rfc2413.txt>` [03.09.01], 1998. RFC 2413.

[116] G. Weikum. The Web in 2010: Challenges and Opportunities for Database Research. In *Informatics. 10 Years Back, 10 Years Ahead, LNCS 2000*, pages 1–23, 2001.

[117] R. Weiss, B. Velez, M. A. Sheldon, C. Nemprempre, P. Szilagyi, A. Duda, and D. K. Gifford. Hypursuit: A Hierarchical Network Search Engine that exploits content-link Hypertext Clustering. In *Proceedings of the Seventh ACM Conference on Hypertext*, Washington (USA), 1996.

[118] A. Wendling. Automatische Erzeugung von Metadaten zur Recherche über WWW-basierte Dokumentsammlungen. Diplomarbeit, J. W. Goethe–Universität, Frankfurt, 2000.

[119] G. Wiederhold. Mediators in the Architecture of Future Information Systems. *Computer Magazine of the Computer Group News of the IEEE Computer Group Society*, 1992.

[120] H. Winkler. Suchmaschinen. *Telepolis*, 1997.

[121] C. Woods and W. Schmolze. The KL-ONE Family. In *Semantic Networks in Artificial Intelligence*, 1992.

[122] R. R. Yager and V. Kreinovich. On how to Merge Sorted Lists Coming from Different Web Search Tools. *Soft Computing*, 3(2), 1999.

[123] Y. Yang. An Evaluation of statistical Approaches to Text Categorization. *Journal of Information Retrieval, 1/2(1)*, 1999.

[124] Y. Yang and X. Liu. A Re-Examination of Text Cateogrization Methods. In *Proceedings of the 22nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, Berkeley (USA), 1999.

[125] B. Yuwono and D.L. Lee. Server Ranking for Distributed Text Retrieval Systems on the Internet. In *Proceedings of the 5th International Conference on Database Systems for Advanced Applications (DASFAA)*, pages 41–49, Melbourne (Australia), 1997.

[126] O. Zamir and O. Etzioni. Web Document Clustering: A Feasibility Demonstration. In *Proceedings of the 21st International ACM SIGIR Conference on Research and Development in Information Retrieval*, Melbourne (Australia), 1998.

[127] O. Zamir and O. Etzioni. Grouper: A Dynamic Clustering Interface to Web Search Results. In *Proceedings of the 8th International World-Wide Web Conference*, Toronto (Kanada), 1999.

[128] X. Zhu and S. Gauch. Incorporating Quality Metrics in Centralized/Distributed Information Retrieval on the World Wide Web. In *Proceedings of the 23rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 288–295, Athen (Griechenland), 2000.

[129] X. Zhu, S. Gauch, L. Gerhard, N. Kral, and A. Pretschner. Ontology-Based Web Site Mapping for Information Exploration. In *Proceedings of the 8th ACM International Conference on Information and Knowledge Management (CIKM)*, Kansas City (USA), 1999.

[130] G. K. Zipf. *Human Behaviour and the Principle of Least Effort*. Addison-Wesley, 1949.

Anhang

A. Spezifikation des Frankfurt Core

```
<rdf:RDF
    xmlns:rdf="http://www.w3.org/TR/WD-rdf-syntax#"
    xmlns:rdfs="http://www.w3.org/TR/WD-rdf-schema#">

<rdf:Description ID="title">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Title"/>
  <rdfs:label>title</rdfs:label>
  <rdfs:comment>
        The name by which the searchserver is formally
        known. The title of the searchserver can be taken from the
        HTML-TITLE element of the front-page of the searchserver.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="abstract">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Description"/>
  <rdfs:label>abstract</rdfs:label>
  <rdfs:comment>
        A textual description summarizing the content of the searchserver.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="times">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource=
                    "http://purl.org/metadata/dublin_core#Coverage"/>
  <rdfs:label>times</rdfs:label>
  <rdfs:comment>
        Temporal period to which the documents of the
        searchserver can be assigned according to its content,
        for instance an era or a time-interval. Identified
        time-specifications are stored in a RDF-Bag container.
  </rdfs:comment>
</rdf:Description>
```

A. Spezifikation des Frankfurt Core

```
<rdf:Description ID="places">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource=
                    "http://purl.org/metadata/dublin_core#Coverage"/>
  <rdfs:label>places</rdfs:label>
  <rdfs:comment>
      Regional coverage to which the documents of the
      searchserver can be assigned according to its content,
      for instance names of cities or countries etc.
      The names are stored in a RDF-Bag container.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="startpage_terms">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Subject"/>
  <rdfs:label>Startpage_terms</rdfs:label>
  <rdfs:comment>
      Terms extracted from the BODY of the front-page of the
      searchserver. Stopwords and Tags are removed. The terms are
      ordered according to their decreasing frequency and will
      be stored in a RDF-Seq container.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="category_terms">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Subject"/>
  <rdfs:label>category_terms</rdfs:label>
  <rdfs:comment>
      Categories from existing Web-Catalogs under which the
      searchserver has already been categorized or categories
      which have been assigned by automatic categorization.
      The categories will be stored in a RDF-Bag container.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="catalog_abstracts">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource=
                    "http://purl.org/metadata/dublin_core#Description"/>
  <rdfs:label>catalog_abstracts</rdfs:label>
  <rdfs:comment>
      Abstracts extracted from existing Web-Catalogs describing
      the searchserver. The available abstracts
      are stored in a RDF-Bag container.
  </rdfs:comment>
```

```
</rdf:Description>

<rdf:Description ID="backlinkpage_terms">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Subject"/>
  <rdfs:label>backlinkpage_terms</rdfs:label>
  <rdfs:comment>
        Terms extracted from the anchortexts of backlinks to the
        front-page of the searchserver. Stopwords and Tags are
        removed. The terms are ordered according to their decreasing
        frequency and will be stored in a RDF-Seq container.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="backlinkpage_titles">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Title"/>
  <rdfs:label>backlinkpage_titles</rdfs:label>
  <rdfs:comment>
        Titles extracted from the HTML-Title elements of
        backlink-pages of the searchserver. The titles are
        stored in a RDF Bag container.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="query_terms">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Subject"/>
  <rdfs:label>query_terms</rdfs:label>
  <rdfs:comment>
        Terms, that are most often queried via the searchinterface
        of the searchserver and which produce at least on hit.
        Stopwords are removed. The terms are ordered according
        to their increasing request frequency and will be
        stored in a RDF-Seq container.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="title_terms">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource=
                      "http://purl.org/metadata/dublin_core#Subject"/>
  <rdfs:label>title_terms</rdfs:label>
  <rdfs:comment>
        Terms from the titles of documents, that can be retrieved
        via the searchinterface of the searchserver.
        Stopwords are removed. The terms ordered according to
```

```
        their decreasing df-weight (df=number of documents in the
        searchserver that contain the term in the document-title)
        and will be stored in a RDF-Seq container.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="content_terms">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Subject"/>
  <rdfs:label> content_terms </rdfs:label>
  <rdfs:comment>
        Content-descriptive terms of documents, that can be
        retrieved via the searchinterface of the searchserver.
        The Terms can be extracted from content-descriptive
        information of the documents in the searchserver
        (i.e. abstracts, available fulltext). Stopwords are
        removed. The terms are ordered according to their
        decreasing df-weight (df=number of documents in the
        searchserver that contain the term) and will be stored
        in a RDF-Seq container.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="language">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Language"/>
  <rdfs:label> language </rdfs:label>
  <rdfs:comment>
         The language in which most of the documents of a searchserver
         can be retrieved.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="publisher">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource=
                    "http://purl.org/metadata/dublin_core#Publisher"/>
  <rdfs:label> publisher </rdfs:label>
  <rdfs:comment>
        The institution or organization that is responsible for
        making the searchserver available
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="format">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Format"/>
```

```
  <rdfs:label> format </rdfs:label>
  <rdfs:comment>
        The data-format of most of the documents that can be
        retrieved via the searchinterface of the searchserver.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="date_created">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Date"/>
  <rdfs:label> date </rdfs:label>
  <rdfs:comment>
        Date at which the searchserver was available in the WWW the first time.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="date_modified">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Date"/>
  <rdfs:label> date </rdfs:label>
  <rdfs:comment>
        Date at which the searchserver's content has been modified
        the last time.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="number_of_backlinks">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:label> number_of_backlinks </rdfs:label>
  <rdfs:comment> Total number of backlinks to the startpage of the
                 searchserver determined via AltaVista's link-keyword-search.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="authority_weight">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:label> authority_weight </rdfs:label>
  <rdfs:comment>
        An authority-weight of the searchserver considering the link-structure
        of the whole Visible Web, respectively a subset that is sufficiently
        large. This can determined for instance by using HITS or Pagerank.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="number_of_documents">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:label> number_of_documents </rdfs:label>
```

```
  <rdfs:comment> Total number of different documents that can be accessed
                 via the search interface of the searchserver.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="number_of_access_per_day">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:label> number_of_access_per_day </rdfs:label>
  <rdfs:comment> Number of unique accesses from different
                 addresses on a single day, averaged over a period of
                 at least 30 days.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="search_interface">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:subPropertyOf resource="http://purl.org/metadata/dublin_core#Relation"/>
  <rdfs:label> search_interface </rdfs:label>
  <rdfs:comment>
        Address of the search-interface of the searchserver.
  </rdfs:comment>
</rdf:Description>


<rdf:Description ID="search_fields">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:label> search_fields </rdfs:label>
  <rdfs:comment>
        Names of searchfields by which the scope of a term can be limited
        (for instance author, title, etc.). The names are stored in a
        RDF-Bag container.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="search_operators">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:label> search_operators </rdfs:label>
  <rdfs:comment>
        Names Operators by which two terms can be combined logically to
        perform a search (for instance AND, OR, AND NOT, NEAR, etc.).
        The names are stored in a RDF-Bag container.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="search_qualifiers">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:label> search_qualifiers </rdfs:label>
```

```
  <rdfs:comment>
        Names of qualifiers by which the quality of a term can be modified
        (for instance Truncation, Concatenation of terms,
        case sensitivity, etc.). The names are stored in a RDF-Bag container.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="search_field_mapping">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:label> search_field_mapping </rdfs:label>
  <rdfs:comment>
        Names of the searchfields that are supported by the
        searchinterface of the searchserver, followed by a colon and
        the names of the semantically equivalent search concepts
        from the concept-taxonomy.
  </rdfs:comment>
</rdf:Description>

<rdf:Description ID="concept_descriptions">
  <rdf:type rdf:resource="http://www.w3.org/TR/WD-rdf-syntax#Property"/>
  <rdfs:label> concept_descriptions </rdfs:label>
  <rdfs:comment>
        Specifications of concept-descriptions for integrating knowledge-
        and ontology based representations. The concept-descriptions are
        stored in a RDF-Bag container.
  </rdfs:comment>
</rdf:Description>
</rdf:RDF>
```

B. Lebenslauf

Persönliche Daten:

Martin Heß
geboren am 19.7.1968 in Hanau
wohnhaft in Frankfurt am Main

Schulausbildung:

1974 – 1978	Grundschule in Rodenbach bei Hanau
1978 – 1987	Gesamtschule Freigericht in Freigericht/Somborn
Mai 1987	Abschluß: Abitur

Wehrdienst:

1987 – 1988	Hindenburgkaserne in Kassel

Hochschulausbildung:

1988 – 1996	Studium der Informatik an der Johann Wolfgang Goethe-Universität in Frankfurt am Main Nebenfach: Medizin, Vordiplom: 1991
September 1996	Abschluß: Diplom

Berufstätigkeit:

Januar 1995 – Juni 1996	Freier Mitarbeiter am IBM European Network Center (ENC) in Heidelberg
Oktober 1996 – September 2001	Wissenschaftlicher Mitarbeiter am Fachbereich Informatik der Johann Wolfgang Goethe-Universität in Frankfurt am Main, Professur für Architektur und Betrieb Verteilter Systeme bei Prof. Dr. O. Drobnik.